

**ACADEMIC REGULATIONS,  
COURSE STRUCTURE & SYLLABUS**

FOR

**MASTER OF COMPUTER APPLICATIONS**  
Three Year PG Day-Time Program  
(With effect from 2019 – 20)



**SCHOOL OF INFORMATION TECHNOLOGY**  
**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**  
**KUKATPALLY, HYDERABAD – 500 085, TS**



SCHOOL OF INFORMATION TECHNOLOGY  
 JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD  
 HYDERABAD – 500 085 - TELANGANA

**ACADEMIC REGULATIONS FOR THE AWARD OF M.C.A DEGREE BASED ON  
 CHOICE BASED CREDIT SYSTEM (CBCS)  
 (WITH EFFECT FROM THE ACADEMIC YEAR 2019 – 2020)**

The Master of Computer Applications Postgraduate Degree of the Jawaharlal Nehru Technological University Hyderabad (JNTUH) shall be conferred on candidates who are admitted to the program and who fulfill all the requirements for the award of the Degree.

JNTUH offers 3 Years (6 Semesters) Master of Computer Applications (M.C.A) degree program, under Choice Based Credit System (CBCS) at its constituent Autonomous College – *School of Information Technology (SIT), JNTUH*, Hyderabad.

**1. ELIGIBILITY FOR ADMISSIONS**

Admission to the above program shall be made subject to eligibility, qualification and specialization as prescribed by the University, from time to time.

- 1.1.** Admissions shall be made on the basis of merit rank obtained by the qualifying candidate in **ICET, a state wide MCA Entrance exam by Telangana State Government**, subject to reservations prescribed by the University time to time.

**2. AWARD OF M.C.A. DEGREE**

- 2.1** A candidate shall be declared eligible for the award of the M.C.A. Degree, if he pursues a course of study in not less than three and not more than six academic years.
- 2.2** A candidate who fails to fulfill all the academic requirements for the award of the degree within six academic years from the year of his admission shall forfeit his seat in M.C.A. course.

**3. M.C.A PROGRAM STRUCTURE**

**3.1** The M.C.A Program of SIT-JNTUH is of Semester Pattern, with 6 Semesters constituting 3 Academic Years, each Academic Year having TWO Semesters (First/Odd and Second/Even Semesters). Each Semester shall be of 22 Weeks duration (inclusive of Examinations).

**3.2** UGC/ AICTE specified Definitions/ Descriptions are adopted appropriately for various terms and abbreviations used in these Academic Regulations/ Norms, which are as listed below.

**3.3 Semester Scheme:** The program is of 3 Academic Years (6 Semesters), with the year being divided into two Semesters of 22 weeks ( $\geq 90$  working days) each, each Semester having - 'Continuous Internal Evaluation (CIE)' and 'Semester End Examination (SEE)'. Choice Based Credit System (CBCS) and Credit Based Semester System (CBSS) as denoted by UGC, and Curriculum/ Course Structure as suggested by AICTE are followed.

**3.4 Credit Courses:** All Subjects/ Courses are to be registered by candidates in a Semester to earn Credits. Credits shall be assigned to each Subject/ Course in a L: P: C (Lecture Periods: Practical Periods: Credits) structure, based on the following general pattern.

- One Credit - for One hour/ Week for Theory/ Lecture (L) Courses
- One Credit - for Two hours/ Week for Laboratory/ Practical (P) Courses

#### **4. COURSE WORK**

**4.1** A candidate after securing admission shall pursue the M.C.A. in a minimum period of 3 Academic Years, and a maximum period of 6 Academic Years (starting from the Date of Commencement of I Year).

**4.2** Each candidate shall register for and secure the specified number of Credits required for the completion and award of the M.C.A. Degree in respective specialization.

**4.3** Each Semester is structured to provide 21 Credits, except final semester which is of 20 credits, totaling to 125 Credits (125 C) for the entire M.C.A. Program.

**4.4 The candidates shall register for all 125 credits and secure all the 125 credits.**

#### **5. COURSE REGISTRATION**

**5.1** A 'Course Coordinator or Faculty Advisor' shall be assigned to each candidates, who will advise him about them M.C.A. program, its Course Structure and Curriculum,

Choice/Option for Subjects/ Courses, based on his competence, progress, pre-requisites and interest.

- 5.2** Academic Section of the College invites 'Registration Forms' from candidates' priory (before beginning of the IV and V Semester). The Registration Requests should be completed BEFORE the commencement of SEEs (Semester End Examinations) of the 'PRECEDING SEMESTER'.
- 5.3** A candidate can register, ONLY AFTER obtaining the 'WRITTEN APPROVAL' from his Course Coordinator, which should be submitted to the College Academic Section.
- 5.4** A candidate may be permitted to register for his Subjects/ Course with a typical total of 21 Credits per Semester (I - V Semesters): **Minimum being 18 Credits and Maximum being 24 Credits for IV and V Semesters**, based on his PROGRESS and SGPA/ CGPA, and completion of the 'PRE-REQUISITES' as indicated for various Subjects/Courses, in the Department Course Structure and Syllabus contents. A candidate must register all the CORE subjects/courses.
- 5.5** The candidate has to register for the audit courses and pass the audit courses for successful completion of the degree. However the credits earned in the audit courses are not included in the computation of SGPA/ CGPA.
- 5.6** Choice for 'additional Subjects/ Courses' to reach the Maximum Permissible Limit of 24 Credits (above the typical 21 Credit norm) must be clearly indicated, which needs the specific approval and signature of the Course Coordinator.
- 5.7** If the candidate submits ambiguous choices or multiple options or erroneous entries - during Registration for the Subject(s) / Course(s) under a given/ specified Course Group/ Category as listed in the Course Structure, only the first mentioned Subject/ Course in that Category will be taken into consideration.
- 5.8** Subject/ Course Options exercised are final and CAN NOT be changed. However, if the Subject/ Course that has already been listed for Registration (by the Course Coordinator) in a Semester could not be offered due to any unforeseen or unexpected reasons, then the Candidates shall be allowed to have alternate choice - either for a new Subject (subject to offering of such a Subject), or for another existing Subject (subject to availability of seats), which may be considered. Such alternate arrangements will be made by the Course Coordinator, with due notification and time framed schedule, within the FIRST WEEK from the commencement of Class-work for that Semester.

**5.9** Dropping of Subjects/ Courses may be permitted, ONLY AFTER obtaining prior approval from the Faculty Advisor (subject to retaining a minimum of 18C in IV and V Semesters), 'within 15 Days of Time' from the beginning of the current Semester.

**5.10** Candidates may register for NPTEL/SWAYAM as per the university rules.

## **6. SUBJECTS/ COURSES TO BE OFFERED**

**6.1** A typical Section (or Class) Strength for each Semester shall be 30.

**6.2** A Subject/ Course (ELECTIVE) may be offered to the Candidates, ONLY If a Minimum of 1/3<sup>rd</sup> of the Section Strength opts for the same.

**6.3** More than ONE TEACHER may offer the SAME SUBJECT (Lab. / Practical's may be included with the corresponding Theory Subject in the same Semester) in any Semester. However, selection choice for candidates will be based on - 'FIRST COME FIRST SERVE Basis and CGPA Criterion'.

**6.4** If more entries for Registration of a Subject come into picture, then the concerned Course Coordinator shall take necessary action, whether to offer such a Subject/ Course for TWO (or multiple) SECTIONS or NOT .

**6.5** In case of options coming from Candidates of other Departments/ Branches/ Disciplines, in case of OPEN ELECTIVES, PRIORITY shall be given to the candidates of the 'Parent Department' first.

## **7. ATTENDANCE REQUIREMENTS**

**7.1** The candidate shall put in a minimum of 75% attendance per semester independently for each of the course/subject registered.

**7.2** Condonation of shortage of attendance up to 10% in each course/subject registered (65% and above and less than 75%) may be given by the College/school Academic Committee.

**7.3.** Shortage of Attendance below 65% shall not be condoned. Condonation of shortage of attendance shall be granted only on genuine and valid reasons on representation by the candidate with supporting evidence and by paying stipulated condonation fee.

**7.4** Candidates whose shortage of attendance is not condoned in any course/subject registered are not eligible to write their end semester examination of that course/subject, they get **DETAINED** in that course/subject. The candidate will have to repeat that course/subject as and when offered; in case if there are any Professional Electives and/or Open Electives,

the same may also be re-registered if offered, however, if those electives are not offered in later semesters, then alternate electives may be chosen from the SAME set of ELECTIVE subjects offered under that category. In such a case candidate has to pay tuition fee for that course/subject.

**7.5** A Candidate shall put in a minimum required attendance at least three (3) theory subjects in each semester for promoting to next Semester. In order to qualify for the award of the MCA Degree, the candidate shall complete all the academic requirements of the subjects, as per the course structure.

## **8. EVALUATION**

The performance of the candidate in each semester shall be evaluated subject-wise, with a maximum of 100 marks for theory and 100 marks for practical's, on the basis of Internal Evaluation and End Semester Examination.

**8.1** Evaluation of candidates in a course/subject involves both external and internal components. External evaluation will be in the form of end semester examination in a course/subject for which is allocated 60% of the marks. The remaining 40% of marks are allocated to internal evaluation.

**8.2** The internal evaluation has two Mid Term-Examinations (each of 40 marks) are conducted- one in the middle of the Semester and the other immediately after the completion of instructions. Each midterm examination shall be conducted for a total duration of 120 minutes. The best one will be considered.

**8.3** The End semester Examination will be conducted in each subject for 60 marks. The question paper consists of 8 questions. Each of these questions may contain sub-questions. Each question carries 12 marks. The candidates will be required to answer 5 questions. The questions are to be prepared to cover the entire range of prescribed syllabi of the subjects and units.

**8.4** For practical subjects, 60 marks shall be awarded based on the performance in the End Semester Examinations and 40 marks for internal evaluation shall be awarded based on day-to-day performance and the internal exam.

**8.5** A candidate shall be deemed to have secured the minimum academic requirement in a subject if he secures a minimum of 40% marks in the End Examination and a minimum aggregate of 50% of the total marks in the End semester examination and internal evaluation taken together.

**8.6** In case the candidate does not secure the minimum academic requirement in any subject

(as specified in 8.5) he has to reappear for the Semester End Examination in that subject.

**8.7** A candidate can re-register for the subjects if the internal marks secured by him/her are less than 50% and failed in that subject for maximum of two subjects and should register within four weeks of commencement of the class work. In such a case, the candidate must re-register for the subjects and secure the required minimum attendance. The candidate's attendance in the re-registered subject(s) shall be calculated to decide upon his eligibility for writing the Semester End Examination in those subjects. In the event of the candidates taking another chance, his Internal Evaluation (internal) marks and Semester End Examination marks obtained in the previous attempt stands cancelled.

## **9. Evaluation of Project / Dissertation Work.**

Every candidate shall be required to submit the thesis or dissertation after taking up a topic approved by the School/College.

**9.1 Registration of Project Work:** A candidate is permitted to register for the project work after satisfying the attendance requirement of all the subjects (theory and practical subjects). A candidate has to choose the topic in the first 2 weeks of the VI Semester in consultation with the Internal/External guides. After 2 weeks candidate has to submit an abstract of work to be carried out to the Project Review Committee (PRC), which in turn allows the candidate to register for thesis work if it is satisfied with the abstract submitted by the candidate.

**9.2** A Project Review Committee (PRC) shall be constituted with Course Coordinator and two members from the school.

**9.3** Only after obtaining the approval of Project Review Committee (PRC), the candidates can initiate the Project work.

**9.4** If a candidate wishes to change his supervisor or topic of the project he can do so with the approval of a committee appointed by the Director, SIT. However, the committee shall examine whether the change of topic/supervisor leads to a major change of his initial plans of project proposal. If so, his date of registration for the project work starts from the date of change of Supervisor or topic as the case may be.

**9.5** The total duration of the project is for 22 weeks which is spread across 22 weeks in VI semester. The candidates can submit the project only after 22 weeks from the date of registration after the approval of PRC.

**9.6** At end of the VI semester,

- a. A candidate has to present a project seminar for which, Internal marks are for 40 marks which will be awarded by an internal committee consists of the concerned guide, course coordinator and a faculty member appointed by the Director, SIT.

External marks are for 60 marks which will be awarded by an external examiner appointed by the Director, SIT. Evaluation of project by Internal Committee (for 40 marks) and by the external examiner (for 60 marks) will be done simultaneously.

b. A candidate shall submit the thesis/dissertation in a hard bound copy form. He will attend the viva-voce conducted by the (External Viva-Voce) an external examiner appointed by the Director SIT. The project is evaluated for 100 marks. There will be no internal marks.

**9.7** The candidate has to submit two hard copies and one soft copy of Thesis/Dissertation, certified in the prescribed format by the supervisor to the school.

**9.8** In case the candidate fails in viva-voce examination, based on the recommendation of the board the candidate has to retake the viva-voce examination after three months. If s/he fails in this viva-voce examination also, he will not be eligible for the award of the degree unless the candidate is asked to revise and resubmit.

## **10. Award of Degree and Class – The Grading System**

**10.1** A candidates shall be declared eligible for the award of M.C.A. degree, if he pursues a course of study and completes it successfully in not less than three academic years and not more than six academic years.

**10.2** A candidates, who fails to fulfill all the academic requirements for the award of the degree within six academic years from the year of his admission, for any reason whatsoever, shall forfeit his seat in M.C.A. Course.

**10.3** A candidates shall register and put up minimum academic requirement in all 125 credits and earn the 125 credits. Marks obtained in all 125 credits shall be considered for the calculation of Cumulative Grade Point Average (CGPA) and percentage of marks.

**10.4** Marks will be awarded to indicate the performance of each candidates in each Theory Subject, or Lab/Practical's, or Seminar, or Project etc., based on the % of marks obtained in CIE + SEE (Continuous Internal Evaluation + Semester End Examination, both taken together) and a corresponding Letter Grade shall be given.

**10.5** As a measure of the candidates' performance, a 10-point Absolute Grading System using the following Letter Grades (UGC Guidelines) and corresponding percentage of marks shall be followed.....

<b>Marks Obtained</b>	<b>Grade</b>	<b>Description of Grade</b>	<b>Grade Points (GP) Value Per Credit</b>
>=90	O	Outstanding	10



>=80 and <90	A+	Excellent	9
>=70 and <80	A	Very Good	8
>=60 and <70	B+	Good	7
>=55 and <60	B	Average	6
>=50 and <55	C	Pass	5
<50	F	Fail	0
Not Appeared the Exam(s)	AB	Absent	0

- 10.6** A candidate obtaining F Grade in any Subject shall be considered ‘failed’ and is be required to reappear as ‘Supplementary Candidate’ in the Semester End Examination (SEE), as and when offered. In such cases, his Internal Marks (IE Marks) in those Subjects will remain the same as those he obtained earlier.
- 10.7** A candidate not appeared for examination then ‘AB’ Grade will be allocated in any Subject shall be considered ‘failed’ and will be required to reappear as ‘Supplementary Candidate’ in the Semester End Examination (SEE), as and when offered.
- 10.8** A Letter Grade does not imply any specific Marks percentage and it will be the range of marks percentage.
- 10.9** In general, a candidate shall not be permitted to repeat any Subject/ Course (s) only for the sake of ‘Grade Improvement’ or ‘CGPA Improvement’.
- 10.10** A candidate earns Grade Point (GP) in each Subject, on the basis of the Letter Grade obtained by him in that Subject. The corresponding ‘Credit Points’ (CP) is computed by multiplying the Grade Point with Credits for that particular Subject.

**Credit Points (CP) = Grade Point (GP) x Credits .... For a Subject**

- 10.11** The Candidates passes the Subject only when he gets **GP ≥ 5 (C Grade or above)**.
- 10.12** The Grade Point Average (GPA) is calculated by dividing the Sum of Credit Points ( $\Sigma CP$ ) secured from ALL Subjects registered in a Semester or for the Exam appeared (like supplementary), by the Total Number of Credits registered during that Semester or for the Exam appeared (like supplementary). GPA is rounded off to FOUR Decimal Places. GPA is thus computed as

$$GPA = \frac{\sum_1^n C_i \times GP_i}{\sum_1^n C_i}$$

where n is the number of subjects Registered in that semester / exam.

$C_i$  is Credits for the subjects.

$GP_i$  is the grade point obtained for the subject

where 'i' is the Subject indicator index (takes into account all Subjects in a Semester or for the Exam appeared), 'N' is the no. of Subjects 'REGISTERED' for the Semester or for the Exam appeared,  $C_i$  is the no. of Credits allotted to the  $i^{\text{th}}$  Subject, and  $G_i$  represents the Grade Points (GP) corresponding to the Letter Grade awarded for that  $i^{\text{th}}$  Subject.

- 10.13** The Cumulative Grade Point Average (CGPA) is a measure of the overall cumulative performance of a candidates over all Subjects in all considered for registration. The CGPA is the ratio of the Total Credit Points secured by a candidate in ALL registered Courses in ALL Semesters, and the Total Number of Credits registered in ALL the Semesters. CGPA is rounded off to FOUR Decimal Places. CGPA is thus computed as per the formula

$$CGPA = \frac{\sum_{j=1}^m GPA_j \times TC_j}{\sum_{j=1}^m TC_j}$$

where m is the number of subjects registered in the course.

$TC_j$  the total number of credits for a  $j^{\text{th}}$  subject.

$GPA_j$  is the Grade point of the  $j^{\text{th}}$  subject.

- 10.14** For Calculations listed in Item 10.6 – 10.10, performance in failed Subjects/ Courses (securing F Grade) will also be taken into account, and the Credits of such Subjects/ Courses will also be included in the multiplications and summations.

## 11. AWARD OF DEGREE AND CLASS

- 11.1** A Candidate who registers for all the specified Subjects/ Courses as listed in the Course Structure, satisfies all the Course Requirements, and passes the examinations prescribed in the entire PG Programme (PGP), and secures the required number of **125** Credits (with  $CGPA \geq 5.0$ ), shall be declared to have 'QUALIFIED' for the award of the M.C.A.

- 11.2 Award of Class:** After a candidate has satisfied the requirements prescribed for the completion of the programme and is eligible for the award of M. C.A. Degree, he shall be placed in one of the following three classes based on the CGPA:

Class Awarded	CGPA
First Class with Distinction	$\geq 8.00$
First Class	$7.00 \leq CGPA < 8.00$
Second Class	$5.00 \leq CGPA < 7.00$

- 11.3** A candidate with final CGPA (at the end of the PGP)  $< 5.00$  will not be eligible for the Award of Degree.

## 12. WITHHOLDING OF RESULTS

**12.1** If the candidate has not paid the dues, if any, to the University or if any case of indiscipline is pending against him, the result of the candidate will be withheld and he will not be allowed into the next semester. His degree will be withheld in such cases.

## 13. TRANSITORY REGULATIONS

**13.1** If any candidate is detained due to shortage of attendance in one or more subjects, they are eligible for re-registration to same or equivalent subjects at a time as and when offered.

**13.2** In case any candidate makes a re-registration then the academic regulations which were applicable for the year of his joining year will be applicable.

## 14. GENERAL

**14.1 Credit:** A unit by which the course work is measured. It determines the number of hours of instructions required per week. One credit is equivalent to one hour of teaching (lecture or tutorial) or two hours of practical work/field work per week.

**14.2 Credit Point:** It is the product of grade point and number of credits for a course.

**14.3** Wherever the words “he”, “him”, “his”, occur in the regulations, they include “she”, “her”.

**14.4** The academic regulation should be read as a whole for the purpose of any interpretation.

**14.5** In the case of any doubt or ambiguity in the interpretation of the above rules, the decision of the Vice-Chancellor is final.

**14.6** The University may change or amend the academic regulations or syllabi at any time and the changes or amendments made shall be applicable to all the candidates with effect from the dates notified by the University.

## 15. MALPRACTICES RULES

### 15.1 DISCIPLINARY ACTION FOR / IMPROPER CONDUCT IN EXAMINATIONS

	Nature of Malpractices/Improper conduct	Punishment
	<i>If the candidate:</i>	
1. (a)	Possesses or keeps accessible in examination hall, any paper, note book, programmable calculators, Cell phones, pager, palm computers or any other form of material concerned with or related to the subject of the examination (theory or practical) in which he is appearing but has not made use of	Expulsion from the examination hall and cancellation of the performance in that subject only.

	(material shall include any marks on the body of the candidate which can be used as an aid in the subject of the examination)	
(b)	Gives assistance or guidance or receives it from any other candidate orally or by any other body language methods or communicates through cell phones with any candidate or persons in or outside the exam hall in respect of any matter.	Expulsion from the examination hall and cancellation of the performance in that subject only of all the candidates involved. In case of an outsider, he will be handed over to the police and a case is registered against him.
2.	Has copied in the examination hall from any paper, book, programmable calculators, palm computers or any other form of material relevant to the subject of the examination (theory or practical) in which the candidate is appearing.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that Semester/year.  The Hall Ticket of the candidate is to be cancelled and sent to the University.
3.	Impersonates any other candidate in connection with the examination.	The candidate who has impersonated shall be expelled from examination hall. The candidate is also debarred and forfeits the seat. The performance of the original candidate who has been impersonated, shall be cancelled in all the subjects of the examination (including practicals and project work) already appeared and shall not be allowed to appear for examinations of the remaining subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat. If the imposter is an outsider, he will be handed over to the police and a case is registered against him.
4.	Smuggles in the Answer book or additional sheet or takes out or arranges to send out the question paper during the examination or answer book or additional sheet, during or after the examination.	Expulsion from the examination hall and cancellation of performance in that subject and all the other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat.
5.	Uses objectionable, abusive or offensive language in the answer paper or in letters to the examiners or writes to the examiner requesting him to award pass marks.	Cancellation of the performance in that subject.

6.	Refuses to obey the orders of the Chief Superintendent/Assistant Superintendent / any officer on duty or misbehaves or creates disturbance of any kind in and around the examination hall or organizes a walk out or instigates others to walk out, or threatens the officer-in charge or any person on duty in or outside the examination hall of any injury to his person or to any of his relations whether by signs or by visible representation, assaults the officer-in- charge, or any person on duty in or outside the examination hall or any of his relations, or indulges in any other act of misconduct or mischief which result in damage to or destruction of property in the examination hall or any part of the College campus or engages in any other act which in the opinion of the officer on duty amounts to use of unfair means or misconduct or has the tendency to disrupt the orderly conduct of the examination.	In case of candidates of the college, they shall be expelled from examination halls and cancellation of their performance in that subject and all other subjects the candidate(s) has (have) already appeared and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The candidates also are debarred and forfeit their seats. In case of outsiders, they will be handed over to the police and a police case is registered against them.
7.	Leaves the exam hall taking away answer script or intentionally tears of the script or any part thereof inside or outside the examination hall.	Expulsion from the examination hall and cancellation of performance in that subject and all the other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat.
8.	Possess any lethal weapon or firearm in the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeits the seat.
9.	If candidates of the college, who is not a candidate for the particular examination or any person not connected with the college indulges in any malpractice or improper conduct mentioned in clause 6 to 8.	Candidates of the colleges expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeits the seat. Person(s) who do not belong to the College will be handed over to police and, a police case will be registered against

		them.
10.	Comes in a drunken condition to the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year.
11.	Copying detected on the basis of internal evidence, such as, during valuation or during special scrutiny.	Cancellation of the performance in that subject and all other subjects the candidate has appeared including practical examinations and project work of that semester/year examinations.
12.	If any malpractice is detected which is not covered in the above clauses 1 to 11 shall be reported to the University for further action to award suitable punishment.	

### 15.2 Malpractices identified by squad or special invigilators

1. Punishments to the candidates as per the above guidelines.
2. Punishment for institutions: (if the squad reports that the college is also involved in encouraging malpractices)
  - (i) A show cause notice shall be issued to the college.
  - (ii) Impose a suitable fine on the college.
  - (iii) Shifting the examination centre from the college to another college for a specific period of not less than one year.

**JNTUH SCHOOL OF INFORMATION TECHNOLOGY**  
(Autonomous)  
**MASTER OF COMPUTER APPLICATIONS**

**I YEAR I SEMESTER**

S. No.	Subject Code	Subject	L	T	P	Max Marks(100)		Credits
						Int	Ext	
1.	MC1C10	Mathematical Foundations of Computer Science	3	-	-	40	60	3
2.	MC1C20	Computer Organization	3	-	-	40	60	3
3.	MC1C30	Object Oriented Programming	3	-	-	40	60	3
4.	MC1C40	Probability and Statistics	3	-	-	40	60	3
5.	MC1C50	Accounting and Financial Management	3	-	-	40	60	3
6.	MC1L10	Object Oriented Programming Through C++ Lab	-	-	4	40	60	2
7.	MC1L20	Computer Organization Lab	-	-	4	40	60	2
8.	MC1L30	IT Workshop	-	-	4	40	60	2
<b>Total</b>			<b>15</b>		<b>12</b>	<b>-</b>	<b>-</b>	<b>21</b>

**I Year –II Semester**

S. No.	Subject Code	Subject	L	T	P	Max Marks(100)		Credits
						Int	Ext	
1.	MC2C10	Data Structures	3	-	-	40	60	3
2.	MC2C20	Linux Programming	3	-	-	40	60	3
3.	MC2C30	Data Communications and computer Networks	3	-	-	40	60	3
4.	MC2C40	Operations Research	3	-	-	40	60	3
5.	MC2C50	Operating Systems	3	-	-	40	60	3
6.	MC2L10	Data Structures through C++ Lab	-	-	4	40	60	2
7.	MC2L20	Computer Networks Lab.	-	-	4	40	60	2
8.	MC2L30	Operating Systems Lab.	-	-	4	40	60	2
<b>Total</b>			<b>15</b>		<b>12</b>	<b>-</b>	<b>-</b>	<b>21</b>

## II Year –I Semester

S. No.	Subject Code	Subject	L	T	P	Max Marks(100)		Credits
						Int	Ext	
1.	MC3C10	Database Systems	3	-	-	40	60	3
2.	MC3C20	Network Security	3	-	-	40	60	3
3.	MC3C30	Design and Analysis of Algorithms	3	-	-	40	60	3
4.	MC3C40	Software Engineering	3	-	-	40	60	3
5	MC3C50	Java Programming	3	-		40	60	3
6.	MC3L10	Java Programming Lab.	-	-	4	40	60	2
7.	MC3L20	Network Security Lab.	-	-	4	40	60	2
8	MC3L30	Database Systems Lab.	-	-	4	40	60	2
9	MC3A10 (AUDIT)	Professional Communication Skills	2	-	-	40	60	0
		<b>Total</b>	<b>17</b>		<b>12</b>	<b>-</b>	<b>-</b>	<b>21</b>



## II-Year –II Semester

S. No.	Subject Code	Subject	L	T	P	Max Marks(100)		Credits
						Int	Ext	
1.	MC4C10	Core-I: Internet Technologies	3	-	-	40	60	3
2.	MC4C20	Core-II:Software Testing Methodologies	3	-	-	40	60	3
3.	MC4C30	Core-III: Data Mining	3	-	-	40	60	3
4.		Open Elective –I	3	-	-	40	60	3
	MC4O11	Soft Computing						
	MC4O12	Distributed Systems						
	MC4O13	Organization Structure and Personnel Management						
	MC4O14	Human Computer Interaction						
	MC4O15	Computer Forensics						
	MC4O16	Internet of Things						
5		Professional Elective – I	3	-		40	60	3
	MC4P11	Mobile Computing						
	MC4P12	Information Retrieval Systems						
	MC4P13	Management Information Systems						
	MC4P14	Programming Languages						
	MC4P15	Computer Graphics						
	MC4P16	Artificial Intelligence						
6.	MC4L10	Internet Technologies Lab.	-	-	4	40	60	2
7.	MC4L20	Data Mining Lab.	-	-	4	40	60	2
8.	MC4L30	Software Testing Lab.	-	-	4	40	60	2
9	MC4A10(AUDIT)	Soft Skills	2	-	-	40	60	0
<b>Total</b>			<b>15</b>		<b>14</b>	<b>-</b>	<b>-</b>	<b>21</b>

## III Year- I Semester

S. No.	Subject Code	Subject	L	T	P	Max Marks(100)		Credits
						Int	Ext	
1.	MC5C10	Core-IV: Android Application Development	3	-	-	40	60	3
2.	MC5C20	Core-V: Data Science	3	-	-	40	60	3
3.	MC5C30	Core-VI: Object Oriented Analysis and Design	3	-	-	40	60	3
4.		Open Elective -II	3	-	-	40	60	3
	MC5O21	Software Project Management						
	MC5O22	Digital Image Processing						
	MC5O23	Social Media Intelligence						
	MC5O24	Machine Learning						
	MC5O25	Advanced Programming						
	MC5O26	Storage Area Networks						
5		Professional Elective - II	3	-		40	60	3
	MC5P21	Cyber Security						
	MC5P22	Distributed Databases						
	MC5P23	Distributed Computing						
	MC5P24	Cloud Computing						
	MC5P25	E-Commerce						
	MC5P26	Semantic Analytics						
6.	MC5L10	Android Application Development Lab	-	-	4	40	60	2
7.	MC5L20	Data Science Lab	-	-	4	40	60	2
8	MC5L30	Object Oriented Analysis and Design Lab	-	-	4	40	60	2
<b>Total</b>			<b>15</b>		<b>12</b>	<b>-</b>	<b>-</b>	<b>21</b>

## III YEAR II SEMESTER

Subject Code	Subject	L	T	P	MAX.MARKS		Credits
					Int.	Ext.	
MC6S10	Project Seminar	-	-	-	40	60	4
MC6P10	Project (Viva – Voce)	-	-	32	-	100	16
	<b>Total</b>						<b>20</b>

**MASTER OF COMPUTER APPLICATIONS****I Year I Semester****MC1C10 MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE****UNIT I**

Mathematical Logic: Statements and notations, Connectives, Well formed formulas, Truth Tables, tautology, equivalence implication, Normal forms. Theory of inference for the statement calculus: Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving. Predicate calculus: Predicative logic, Free and Bound variables, The Universe of Discourse. Inference theory of predicate calculus involving quantifiers.

**UNIT II**

Relations: Properties of binary Relations, Relation matrix and graph of a relation, partition and covering of a set, equivalence relation, compatibility relations, composition of binary relations, partial ordering, Partially ordered set: Lattices, Hasse diagram. Functions: Composition of functions, Inverse Function, Hashing functions.

Algebraic structures: Algebraic systems, Examples and general properties, Semi groups and monoids, groups and sub groups, homomorphism.

**UNIT III**

Elementary Combinatorics: Basics of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial and Multinomial theorems, the principles of Inclusion – Exclusion, Pigeon hole principles and its application.

**UNIT IV**

Recurrence Relations: Generating Functions, Function of Sequences, Calculating Coefficients of generating functions, Recurrence relations, Solving recurrence relation by substitution and Generating functions, the method of Characteristic roots, solution of Inhomogeneous Recurrence Relations.

**UNIT V**

Graphs: Basic Concepts, Isomorphism and Sub graphs, Trees and their properties, Spanning Trees, Directed trees, Binary trees, Planar Graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

## TEXT BOOKS:

1. Discrete Mathematics with Applications to Computer Science, J P Trembley and R Manohar, TMH, rp 2008. (Units I and II)
2. Discrete Mathematics for Computer Scientists and Mathematicians, second edition, J.L.Mott, A. Kandel, T.P. Baker, PHI(Unit III, IV & V).

## REFERENCE BOOKS:

1. Elements of Discrete Mathematics- A Computer Oriented Approach,C.L.Liu, D.P. Mohapatra,3<sup>rd</sup> edition,TMH.
2. Discrete and Combinatorial Mathematics- An Applied Introduction-5th Edition– Ralph. P.Grimaldi, Pearson Education.
3. Discrete Mathematics with applications, Thomas Koshy,Elsevier.
4. Discrete Mathematical Structures, Mallik and Sen, CengageLearning.
5. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby, Sharon Cutler Ross, PHI.
6. Discrete Mathematics and its Applications,6<sup>th</sup> edition,K.H.Rosen,TMH.
7. Discrete Mathematics, Lovasz, Springer.
8. Logic and Discrete Mathematics, Grass Man and Tremblay,Pearson Education.
9. Discrete Mathematics, S K Chakraborty and B K Sarkar, Oxford, 2011.

## MASTER OF COMPUTER APPLICATIONS

### I Year I Semester COMPUTER ORGANIZATION

#### MC1C20

#### UNIT I

NUMBER SYSTEMS AND COMPUTER ARITHMETIC- Signed and unsigned numbers, Addition and subtraction, multiplication, division, Floating point representation, logical operation, Gray code, BCD codes, Error detecting codes, Boolean algebra, Simplification of Boolean expressions, K-Maps, Combinational and Sequential Circuits- decoders, Encoders, Multiplexers, Half and Full adders, Shift registers, Sequential circuits- flip-flops.

#### UNIT II

MEMORY ORGANIZATION-Memory hierarchy, Main memory-RAM, ROM chips, Memory address map, memory contention to CPU, Associative Memory-Hardware logic, match, read and write logic, Cache Memory-Associative mapping, Direct mapping, Set-associative mapping, hit and miss ratio.

#### UNIT III

BASIC CPU ORGANIZATION-Introduction to CPU, Instruction formats-INTEL-8086 CPU architecture-Addressing modes - generation of physical address- code segment registers, Zero, one, two, and three address instructions. INTEL 8086 ASSEMBLY LANGUAGE INSTRUCTIONS-Data transfer instructions-input- output instructions, address transfer, Flag transfer, arithmetic, logical, shift, and rotate instructions.conditional and unconditional transfer, iteration control, interrupts and process control instructions, assembler directives, Programming with assembly language instructions.

#### UNIT IV

INPUT -OUTPUT ORGANIZATION-Peripheral devices, input-output interface-I/O Bus and interface modules, I/O versus Memory bus, isolated versus memory mapped I/O, Modes of transfer-Programmed I/O, Interrupt-initiated I/O, priority interrupts-Daisy chaining, parallel priority, interrupt cycle, DMA-DMA control, DMA transfer, Input output processor-CPU-IOP communication.

#### UNIT V

PIPELINE AND VECTOR PROCESSING: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

MULTI PROCESSORS: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor Arbitration, InterProcessor Communication and Synchronization Cache Coherence, Shared Memory Multiprocessors.

#### TEXT BOOKS:

1. Computer System Architecture, M. Morris Mano , 3rd Edition, Pearson Education, 2008.
2. Microprocessors and Interfacing, Douglas Hall, Tata McGraw-Hill.

#### REFERENCE BOOKS:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Vth Edition, McGraw Hill.

**MASTER OF COMPUTER APPLICATIONS****I Year I Semester****MC1C30****OBJECT ORIENTED PROGRAMMING****UNIT I**

Different paradigms for problem solving, need for OOP paradigm, classes and instances, fundamental characteristics of OOP (Alan key), differences between OOP and Procedure Oriented Programming.

C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statements- if, switch, while, for, do, break, continue, goto statements.

**UNIT II**

C++ Functions-Scope of variables, Parameter passing methods, Default arguments, inline functions, Recursive functions, Pointers to functions.

C++ Classes And Data Abstraction: Class definition, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction, ADT and information hiding.

**UNIT III**

Dynamic memory allocation and deallocation operators-new and delete, Dynamic creation and destruction of objects, Preprocessor directives, command line arguments, name spaces.

Polymorphism: Function overloading, Operator overloading, Generic programming-necessity of templates, Function templates and class templates.

**UNIT IV**

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.

Virtual Functions And Run Time Polymorphism: Overriding, Static and Dynamic bindings, Base and Derived class virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Virtual destructors.

**UNIT V**

C++ I/O: I/O using C functions, C++ Stream classes hierarchy, Stream I/O, File streams and String streams, File Operations, Overloading << and >> operators, Error handling during file operations, Formatted I/O.

Exception Handling: Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

**TEXT BOOKS:**

1. C++, The Complete Reference, 4th Edition, Herbert Schildt, TMH.
2. Object Oriented Programming in C++, 4th Edition, R.Lafore, Pearson Education

**REFERENCE BOOKS:**

1. An Introduction to OOP, 3rd Edition, T. Budd, Pearson Education,2008.
2. Programming Principles and Practice Using C++, B.Stroutstrup, Pearson Education.
3. Problem solving with C++, 6th Edition, Walter Savitch, Pearson Education,2007.
4. Mastering C++, K.R.Venu Gopal, Raj Kumar and T.Ravi Shankar, TMH.
5. OOP in C++, 3rd Edition, T.Gaddis, J.Walters and G.Muganda, Wiley DreamTech Press.
6. An Introduction to OOP in C++ with applications in Computer Graphics, 2nd Edition, G.M.Seed, Springer.
7. Programming with ANSI C++, B.Trivedi,Oxford Press.
8. Programming in C++,M.T.Somasekara,PHI.



**MASTER OF COMPUTER APPLICATIONS**  
**I Year I Semester**  
**PROBABILITY AND STATISTICS**

**MC1C40**  
**UNIT I**

Probability: Sample space and events – Probability – The axioms of probability - Some elementary theorems - Conditional probability – Baye’s theorem.

**UNIT II**

Random variables – Discrete and continuous – Distribution – Distribution function.

Distribution - Binomial, poisson and normal distribution – related properties.

**UNIT III**

Sampling distribution: Populations and samples - Sampling distributions of mean (known and unknown) proportions, sums and differences.

Estimation: Point estimation – interval estimation - Bayesian estimation.

**UNIT IV**

Test of Hypothesis – Means and proportions – Hypothesis concerning one and two means – Type I and Type II errors. One tail, two-tail tests. Tests of significance – Student’s t-test, F-test,  $\chi^2$  test. Estimation of proportions.

**UNIT V**

Curve fitting: The method of least squares – Inferences based on the least squares estimations - Curvilinear regression – multiple regressions – correlation for univariate and bivariate distributions.

**TEXT BOOKS:**

1. Probability and statistics for engineers (Erwin Miller and John E.Freund), R A Johnson and C.B.Gupta.7th edition, PHI.
2. Introduction to Probability and Statistics, 12th edition, W.Mendenhall, R.J.Beaver and B.M.Beaver, Cengage Learning.

**REFERENCE BOOKS:**

1. Text book of Probability and Statistics Dr.Shahnaz Bathul, V.G.S.Publishers 2003.
2. Probability and Statistics in Engineering, 4th Edition, William W.Hines, Douglas C.Montgomery, David M.Goldsmn, Connie M.Borrer, Wiley Student Edition.
3. Probability, Statistics and Queuing Theory, 2nd Edition, Trivedi, John Wiley and Sons
4. Introduction to Probability and Statistics, J.S.Milton, Jesse C.Arnold, 4th edition, TMH.
5. Probability,Statistics and Random Processes,Dr.K.Murugesan,P.Guruswamy,Anuradha Agencies, Deepti Publications.
6. Probability and Statistics for Engineers,G.S.S.Bhisma Rao,Sitech,2<sup>nd</sup> edition,2005.
7. Probability and Statistics for Engineers and Scientists,R.E.Walpole,S.L.Myers,K.Ye, Pearson.

**MASTER OF COMPUTER APPLICATIONS**  
**I Year I Semester**  
**MC1C50 ACCOUNTING AND FINANCIAL MANAGEMENT**

**UNIT I**

**Introduction to Accounting:** Principles, concepts and conventions, double entry system of accounting, introduction to basic books of accounts, Journal, ledger- Trial Balance - Preparation of Final accounts: Trading Account, Profit and Loss Account and Balance Sheet. Use of spread sheet to solve the above problems.

**UNIT II**

**Financial Management** - Meaning and scope, role of Financial Manager, Objectives of time value of money - Goals of Financial Management, Leverages: Operating, Financial Leverage and Combined Leverage Cost of Capital: Cost of Equity, Preference Shares, Bonds- Weighted Average Cost of Capital – Capital Gearing- Overcapitalization and Undercapitalization, Sources of Finance. Use of spread sheet to solve the above problems.

**Unit III**

**Tools and Techniques for Financial Statement Analysis:** Ratio Analysis – Classification of Ratios – Short term solvency and long term solvency – Profitability ratios - Analysis and Interpretation of Financial Statements through ratios of Liquidity, Solvency and Profitability ratios. **Fund Flow Statement** - Meaning, Importance, Statement of changes in working capital and statement of Sources and application of funds. Cash flow Analysis: cash flow Statements: Preparation, Analysis and interpretation. Use of spread sheet to solve the above problems.

**UNIT IV**

**Break-even Analysis:** Concept of Break Even Point, Cost-Volume-Profit Analysis, Determination of Break Even Point, Margin of Safety and PV ratio, Impact of changes in Cost or selling price on BEP Practical applications of Break-even Analysis.

**Budgeting :** Budgeting – cash budget, sales budget – flexible Budgets and master budgets. Use of spread sheet to solve the above problems.

**Unit V**

**Capital Budgeting:** Capital and its significance, Types of Capital, Estimation of Fixed and Working capital requirements, Methods and sources of raising capital.. Capital Budgeting: features of capital budgeting proposals, Methods of Capital Budgeting: Payback Method, Accounting Rate of Return (ARR) and Net Present Value Method (simple problems). Use of spread sheet to solve the above problems.

## TEXT BOOKS:

1. Aryasri: Accounting And Financial Management,, TMH, 2009
2. Van Horne, James, C: Financial Management and Policy, Pearson, 2009

## REFERENCE BOOKS:

1. Dr. G. Vidyanath G. Lakshmi, Accounting and Financial Management.
2. Prasanna Chandra, Financial Management, TMH, 2009
3. S.N.Maheshwari, Financial Accounting, Sultan Chand, 2009.
4. Tulsian, Financial Accounting, S Chand, 2009.
5. Khan and Jain: Financial Management, TMH, 2009
6. Gokul Sinha: Financial Statement Analysis, PHI, 2009
7. Bhat Sundhindra:Financial Management, Excel:2009
8. Jawaharlal:Accounting for Management, Himalaya, 2009
9. Paresh Shah : Basic Financial Accounting for Management, Oxford 2010.
10. K. Scott Proctor, Building Financial Models with Microsoft Excel, 2<sup>nd</sup> edition, Wiley Finance.

**MASTER OF COMPUTER APPLICATIONS****I Year I Semester****MC1L10 OBJECT ORIENTED PROGRAMMING THROUGH C++ LAB**

1. Write a C++ program to find the sum of individual digits of a positive integer.
2. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.
3. Write a C++ program to generate all the prime numbers between 1 and n ,  
where n is a value supplied by the user.
4. Write C++ programs that use both recursive and non-recursive functions
  - a) To find the factorial of a given integer.
  - b) To find the GCD of two given integers.
  - c) To find the nth Fibonacci number.
5. Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.
6. Write a C++ program that uses functions
  - a) To swap two integers.
  - b) To swap two characters.
  - c) To swap two reals. Note: Use overloaded functions.
7. Write a C++ program to find both the largest and smallest number in a list of integers.
8. Write a C++ program to sort a list of numbers in ascending order.
9. Write a C++ program that uses function templates to solve problems-7&8.
10. Write a C++ program to sort a list of names in ascending order.
11. Write a C++ program to implement the matrix ADT using a class. The operations supported by this ADT are:
  - a) Reading a matrix.
  - b) Printing a matrix.
  - c) Addition of two matrices.
  - d) Multiplication of two matrices.
12. Implement the matrix ADT presented in the problem-11 using overloaded operators (<<, >>, +, \*) and templates.
13. Implement the complex number ADT in C++ using a class. The complex ADT is used to represent complex numbers of the form  $c=a+ib$ , where a and b are real numbers. The operations supported by this ADT are:

- a) Reading a complex number.
- b) Writing a complex number.
- c) Addition of two complex numbers.
- d) Multiplication of two complex numbers.

14. Write a C++ program that overloads the + operator and relational operators (suitable) to perform the following operations:

- a) Concatenation of two strings.
- b) Comparison of two strings.

15. Implement the complex number ADT in C++ using a class. The complex ADT is used to represent complex numbers of the form  $c=a+ib$ , where a and b are real numbers. The operations supported by this ADT are:

- a) Reading a complex number.
- b) Writing a complex number.
- c) Addition of two complex numbers.
- d) Multiplication of two complex numbers.

Note: 1. overload << and >> operators in part a) and part b).

2. overload +, \* operators in parts c) and d).

16. Write a template based C++ program that determines if a particular value occurs in an array of values.

17. Write a C++ program that uses functions to perform the following operations:

- a) Insert a sub-string into the given main string from a given position.
- b) Delete n characters from a given position in a given string.

18. Write a C++ program that uses a function to reverse the given character string in place, without any duplication of characters.

19. Write a C++ program to make the frequency count of letters in a given text.

20. Write a C++ program to count the lines, words and characters in a given text.

21. Write a C++ program to determine if the given string is a palindrome or not.

22. Write a C++ program to make frequency count of words in a given text.

23. Write a C++ program that displays the position or index in the string S where the string t begins, or -1 if S doesn't contain t.

24. 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C++ program to find the 2's complement of a binary number.

25. Write a C++ program that counts the number of 1 bits in a given integer.

26. Write a C++ program to generate Pascal's triangle.
27. Write a C++ program to construct of pyramid of numbers.
28. Write a C++ program to compute the Sine series.
29. Write a C++ program that converts Roman numeral into an Arabic integer.
30. Write a C++ program which converts a positive Arabic integer into its corresponding Roman Numeral.
31. Write a C++ program to display the contents of a text file.
32. Write a C++ program which copies one file to another.
33. Write a C++ program to that counts the characters, lines and words in the text file.
34. Write a C++ program to change a specific character in a file.

Note: Filename , number of the byte in the file to be changed and the new character are specified on the command line.

35. Write a C++ program to reverse the first n characters in a file.
36. Write a C++ program that uses a function to delete all duplicate characters in the given string.
37. Write a C++ program that uses a function to convert a number to a character string.
38. Write a C++ program that uses a recursive function to find the binary equivalent of a given non-negative integer n.
39. Write a C++ program to generate prime numbers up to n using Sieve of Eratosthenes method.
40. Write a C++ program
  - a) To write an object to a file.
  - b) To read an object from the file.
41. Write C++ programs that illustrate how the following forms of inheritance are supported:
  - a) Single inheritance      b) Multiple inheritance
  - c) Multi level inheritance      d) Hierarchical inheritance
42. Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class.
43. Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions.
44. Write a C++ program that illustrates the role of virtual base class in building class hierarchy.
45. Write a C++ program that illustrates the role of abstract class in building class hierarchy.

## TEXT BOOKS:

1. Mastering C++, K.R.Venu Gopal, Raj Kumar and T.Ravi Shankar, TMH.
2. C++ Programming, D.S.Malik, Cengage Learning.
3. Practical C++ Programming,S.Qualline,SPD.
4. Object Oriented Programming with C++, E.Balaguruswamy, 4<sup>th</sup> Edition, TMH,2008.
5. OOP with C++,S.Sahay,Oxford Higher Education.
6. C++ and OOP Paradigm,D.Jana,2<sup>nd</sup> Edition, PHI
7. Fundamentals of C++ Programming,S.Subramanian,Jaico Publishing House.

**MASTER OF COMPUTER APPLICATIONS**  
**I Year I Semester**  
**MC1L20 COMPUTER ORGANIZATION LAB**

Write assembly language programs for the following using MASAM.

1. Write assembly language programs to evaluate the expressions:

i)  $a = b + c - d * e$

ii)  $z = x * y + w - v + u / k$

a. Considering 8-bit, 16 bit and 32 bit binary numbers as b, c, d, e.

b. Considering 2 digit, 4digit and 8 digit BCD numbers.

Take the input in consecutive memory locations and results also.

Display the results by using “int xx” of 8086. Validate program for the boundary conditions.

2. Write an ALP of 8086 to add two exponential numbers which are in IEEE 754 notation.

Display the results by using “int xx” of 8086. Validate program for the boundary conditions.

3. Write an ALP of 8086 to take N numbers as input. And do the following operations on them.

a) Arrange in ascending and Descending order.

b) Find max and minimum

c) Find average

Considering 8-bit, 16 bit binary numbers and 2 digit, 4digit and 8 digit BCD numbers.

Display the results by using “int xx” of 8086. Validate program for the boundaryconditions.

4. Write an ALP of 8086 to take a string of as input (in ‘C’ format)and do the following Operations on it.

a) Find the length                      b) Find it is Palindrome or not

c) Find whether given string substring or not.



- d) Reverse a string                      e) Concatenate by taking another string

Display the results by using “int xx” of 8086.

5. Write the ALP to implement the above operations as procedures and call

from the main procedure.

6. Write an ALP of 8086 to find the factorial of a given number as a

Procedure and call from the main program which display the result.

7. Write an assembly language program to encrypt digits as shown below:

Input digit : 0 1 2 3 4 5 6 7 8 9

Encrypted digit : 4 6 9 5 0 3 1 8 7 2

Your program should accept a string consisting of digits. The encrypted string should be displayed using “int xx” of 8086.

8. Write a procedure to locate a character in a given string. The procedure receives a

pointer to a string and character to be located. When the first occurrence of the character is located, its position is returned to main. If no match is found, a negative value is returned. The main procedure requests a character string and a character to be located and displays the result.

9. Write an assembly language program to read a string of characters from the user and that prints the vowel count. Display the results by using “int xx” of 8086.

ex. Input : Advanced Programming in UNIX

Out put:

Vowel	count
a or A	3
e or E	1
i or I	3
o or O	1
u or U	1

10. A computer uses RAM chips of 1024 X 1 capacity.

a) How many chips are needed, and how should their address lines be connected to provide a memory capacity of 1024 bytes?

b) How many chips are needed to provide a memory capacity of 16K bytes?

11. A computer employs RAM chips of 256X8 and ROM chips of 1024 X 8. The computer needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped I/O configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, 10 for interface registers.
- How many RAM and ROM chips are needed?
  - Draw a memory-address map for the system.
  - Give the address range in hexadecimal for RAM, ROM and interface.
12. Obtain the complement function for the match logic of one word in an associative memory. Draw the logic diagram for it and compare with the actual match logic diagram.
13. A two-way set associative cache memory uses blocks of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128K X 32.
- Formulate all pertinent information required to construct the cache memory.
  - What is the size of the cache memory?
14. A digital computer has a memory unit of 64K X 16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.
- How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.
  - How many bits are there in the tag, index, block, and word fields of the address format?
  - How many blocks can the cache accommodate?
15. An address space is specified by 24 bits and the corresponding memory space by 16 bits.
- How many words are there in the address space?
  - How many words are there in the memory space?
  - If a page consists of 2K words, how many pages and blocks are there in the system.
16. A virtual memory has a page size of 1K words. There are eight pages and four blocks. The associative memory page table contains the following entries. Make a list of all virtual addresses(in decimal) that will cause a page fault.

Page	Block
0	3

1	1
4	2
6	0

**TEXT BOOKS:**

1. IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI.
2. Introduction to Assembly Language Programming, Sivarama P.Dandamudi, Springer Int. Edition,2003.
3. The 8088 and 8086 Microprocessors: Programming , Interfacing,Software,Hardware and Application,4<sup>th</sup> edition,W.A.Triebel,A.Singh,N.K.Srinath,Pearson Education.

**MASTER OF COMPUTER APPLICATIONS**  
**I Year I Semester**  
**IT WORKSHOP**

**MC1L30**

**Objectives:**

The IT Lab. for students is a training lab course spread over one semester. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Documentation tool (like Word/OO Writer), Spread sheet tool (like Excel/OO Calc), Presentation tool (like Power point/OO Impress).

**PC Hardware** introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. **The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible.**

**Internet & World Wide Web** module introduces the different ways of hooking the PC on to the internet from home and workplace and effectively usage of the internet. Usage of web browsers, email.

**Productivity tools** module would enable the students in crafting professional documents, spread sheets and presentations.

**PC Hardware**

**Week 1 – Task 1 :** Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

**Week 2 – Task 2 :** Every student should disassemble and **assemble the PC back to working condition.** Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

**Week 3 – Task 3 :** Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

**Week 4 – Task 4 :** Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva

**Week 5 – Task 5 : Hardware Troubleshooting :** Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva

**Week 6 – Task 6 : Software Troubleshooting :** Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

### **Internet & World Wide Web**

**Week 7 - Task 1 : Orientation & Connectivity Boot Camp :** Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

**Week 8 - Task 3 : Search Engines & Netiquette :** Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

### **MS Word(or Similar tool)**

**Week 9&10:** The mentor needs to give an overview of Microsoft (MS) word 2007: Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word. Give a task covering to create project certificate. Features to be covered:-Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Inserting table, using Drawing toolbar in word.

### **MS Excel(or Similar tool)**

**Week 11&12:** The mentor needs to tell the importance of MS office 2007 Excel as a Spreadsheet tool covering Accessing, overview of toolbars, saving excel files, Using help and resources., Also give a task that is covering the features like Gridlines, Format Cells, Summation, auto fill, Formatting Text. Solve Accounts and Financial Management problems.

**MS Power Point(or Similar tool)**

**Week 13&14:** Students will be working on MS power point which help them create basic power point presentation. Topic covered during this week includes :- PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in Power point. Students will be given model power point presentation which needs to be replicated (exactly how it's asked).

In addition to above, every student should learn basic linux commands(file and directory commands), editor commands(vi/gedit).

**REFERENCE BOOKS:**

1. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dreamtech
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt,  
WILEY Dreamtech
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
4. PC Hardware and A+ Handbook – Kate J. Chase PHI (Microsoft)
5. LaTeX Companion – Leslie Lamport, PHI/Pearson.
6. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson  
and Ken Quamme. – CISCO Press, Pearson Education.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan,  
CISCO Press, Pearson Education.
8. Troubleshooting, Maintaining and Repairing PCs, S.J. Bigelow, 5<sup>th</sup> edition, TMH.
9. Office 2007, C. Grover, M. Donald and E.A. Vander Veer, O' Reilly, SPD.
10. Step by Step 2007 Microsoft Office System, J. Cox and others (Microsoft), PHI.

**MASTER OF COMPUTER APPLICATIONS**  
**I Year II Semester**  
**DATA STRUCTURES**

**MC2C10**

**Unit I**

Basic concepts-Data types, Abstract Data Types, Data structures, Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Analysis- Big O, Omega and Theta notations, Basic concepts of STL (Standard Template Library)-Containers, Iterators, algorithms.

Introduction to Linear and Non Linear data structures, Linear data structures- Linear Lists, Sequential and Linked allocation, The list ADT, array and linked Implementations, Singly Linked Lists-Operations- Insertion, Deletion, Doubly Linked Lists- Operations- Insertion, Deletion, Circularly linked lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

**Unit II**

Stack ADT, definition, operations, array and linked implementations, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definitions and operations, array and linked Implementations, Circular queues, Insertion and deletion operations, Deque (Double ended queue) ADT, array and linked implementations.

**Unit III**

Non Linear data structures- Trees – Basic Terminology, Binary tree ADT, array and linked representations, traversals, threaded binary trees, Priority Queue ADT-implementation, Heaps.

Graphs – Introduction, Basic Terminology, Graph Representations- Adjacency matrix, Adjacency lists, Graph traversals- DFS and BFS

**Unit IV**

Searching- Linear Search, Binary Search, Hashing-Introduction, hash tables, hash functions, collision resolution methods, Comparison of Searching methods.

Sorting- Bubble Sort, Insertion Sort, Selection Sort, Radix Sort, Quick sort, Merge sort, Heap Sort Comparison of Sorting methods.

**Unit V**

Search Trees-Binary Search Trees, Definition, ADT, Implementation, Operations- Searching, Insertion and Deletion, AVL Trees (Definition and Example only), B-Trees, Definition, B-Tree of order m, operations- Insertion and Searching, Red-Black trees (Elementary treatment-only Definitions and Examples), Comparison of Search Trees.

Pattern matching algorithm- The Knuth-Morris-Pratt algorithm, Text Compression-Huffman coding and decoding algorithms, Tries (examples only).

**TEXT BOOKS :**

1. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Cengage Learning.
2. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, 3<sup>rd</sup> edition, Pearson Education. Ltd.
3. Data structures , Algorithms and Applications in C++, S.Sahani, Universities Press.

**REFERENCE BOOKS :**

1. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and D.Mount, Wiley student edition, seventh edition, John Wiley and Sons.
2. ADTs, Data structures and Problem Solving with C++, 2<sup>nd</sup> edition, L.Nyhoff, Pearson (Prentice Hall).
3. Data Structures using C++, D.S. Malik, Cengage Learning, India Edition.
4. Data structures with C++ Using STL, 2<sup>nd</sup> edition, W.H.Ford and W.R.Topp, Pearson (Prentice Hall).
5. Data Structures using C++, V.H.Patil, Oxford University Press.
6. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
7. Advanced Data structures & Algorithms in C++, V.V.Muniswamy, Jaico Publishing House
8. C++ Plus Data Structures, 4<sup>th</sup> edition, Nell Dale, Jones and Bartlett India Pvt. Ltd.



**MASTER OF COMPUTER APPLICATIONS**  
**I Year II Semester**  
**LINUX PROGRAMMING**

**MC2C20**

**UNIT I**

Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

Sed-Scripts,Operation,Addresses,Commands,Applications, awk-Execution,Fields and Records,

Scripts,Operation,Patterns,Actions,Associative Arrays,String and Mathematical functions,System commands in awk,Applications.

Shell programming with Bourne again shell(bash)- Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

**UNIT II**

Files and Directories- File Concept, File types, File System Structure,file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, creat, read, write, close, lseek, dup2,file status information-stat family, file and record locking-lockf and fcntl functions,file permissions - chmod, fchmod,file ownership-chown,lchown,fchown, links-soft links and hard links – symlink, link, unlink.

Directories-Creating,removing and changing Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, Directory contents,Scanning Directories-opendir, readdir, closedir,rewinddir, seekdir, telldir functions.

**UNIT III**

Process – Process concept, Layout of a C program image in main memory,Process environment-environment list,environment variables,getenv,setenv,Kernel support for process, process identification, process hierarchy,process states, process control - process creation, replacing a process image,waiting for a process, process termination, zombie process,orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, system,I/O redirection,Process Groups,Sessions and Controlling Terminal.

Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.

**UNIT IV**

Interprocess Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems,pipes-creation,IPC between related processes using unnamed

pipes, FIFOs-creation,IPC between unrelated processes using FIFOs(Named pipes), differences between unnamed and named pipes,popen and pclose library functions.

Message Queues- Kernel support for messages, APIs for message queues, client/server example.

Semaphores-Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

## UNIT V

Shared Memory- Kernel support for shared memory, APIs for shared memory, shared memory example.

Sockets- Introduction to Berkeley Sockets, IPC over a network,Client-Server model,Socket address structures (Unix domain and Internet domain),Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection,Multiple simultaneous clients, Comparison of IPC mechanisms.

### TEXT BOOKS:

1. Unix System Programming using C++, T.Chan, PHI.
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH,2006.
3. Beginning Linux Programming, 4<sup>th</sup> Edition, N.Matthew, R.Stones,Wrox, Wiley India Edition,rp-2008.
4. Unix Network Programming ,W.R.Stevens,PHI.
5. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.

### REFERENCE BOOKS:

1. Linux System Programming, Robert Love, O'Reilly, SPD, rp-2007.
2. Unix for programmers and users, 3<sup>rd</sup> Edition, Graham Glass, King Ables, Pearson Education, 2003.
3. Advanced Programming in the Unix environment, 2<sup>nd</sup> Edition, W.R.Stevens, Pearson Education.
4. System Programming with C and Unix,A.Hoover,Pearson.
5. Unix System Programming,Communication,Concurrency and Threads,K.A.Robbins and S.Robbins,Pearson Education.
6. Unix shell Programming,S.G.Kochan and P.Wood,3<sup>rd</sup> edition,Pearson Education.
7. Shell Scripting,S.Parker,Wiley India Pvt. Ltd.
8. C Programming Language,Kernighan and Ritchie,PHI

**MASTER OF COMPUTER APPLICATIONS**  
**I Year II Semester**

**MC2C30 DATA COMMUNICATIONS AND COMPUTER NETWORKS**

**UNIT I**

Introduction to Networks, internet, protocols and standards, the OSI model, layers in OSI model, TCP/IP suite, Addressing, Analog and digital signals.

Physical Layer: digital transmission, multiplexing, transmission media, circuit switched networks, Datagram networks, virtual circuit networks, switch and Telephone network.

**UNIT II**

Data link layer: Introduction, Block coding, cyclic codes, checksum, framing, flow and error control, Noiseless channels, noisy channels, HDLC, point to point protocols

Medium Access sub layer: Random access, controlled access, channelization, IEEE standards, Ethernet, Fast Ethernet, Giga-Bit Ethernet, wireless LANs.

**UNIT III**

Connecting LANs, backbone networks and virtual LANs, Wireless WANs, SONET, frame relay and ATM.

Network Layer: Logical addressing, internetworking, tunneling, address mapping, ICMP, IGMP, forwarding, uni-cast routing protocols, multicast routing protocols.

**UNIT IV**

Transport Layer: Process to process delivery, UDP and TCP protocols, SCTP, data traffic, congestion, congestion control, QoS, integrated services, differentiated services, QoS in switched networks.

**UNIT V**

Application Layer – Domain name space, DNS in internet, electronic mail, FTP, WWW, HTTP, SNMP, multi-media, network security

**TEXT BOOKS:**

- 1.Data Communications and Networking , Behrouz A. Forouzan, Fourth Edition TMH.
- 2.Computer Networks, Andrew S Tanenbaum, 4th Edition. Pearson Education.

**REFERENCE BOOKS:**

- 1.An Engineering Approach to Computer Networks,S.Keshav,2nd Edition,Pearson Education.

**MASTER OF COMPUTER APPLICATIONS****I Year II Semester  
OPERATIONS RESEARCH****MC2C40****UNIT I**

Introduction to Operations Research: Basics definition, scope, objectives, phases, models and limitations of Operations Research. Linear Programming Problem – Formulation of LPP, Graphical solution of LPP. Simplex Method, Artificial variables, big-M method, two-phase method, degeneracy and unbound solutions.

**UNIT II**

Transportation Problem: Formulation, solution, unbalanced Transportation problem. Finding basic feasible solutions – Northwest corner rule, least cost method and Vogel's approximation method. Optimality test: the stepping stone method and MODI method.

Assignment model: Formulation. Hungarian method for optimal solution. Solving unbalanced problem. Traveling salesman problem as assignment problem.

**UNIT III**

Sequencing models: Solution of Sequencing Problem – Processing n Jobs through 2 Machines – Processing n Jobs through 3 Machines – Processing 2 Jobs through m machines – Processing n Jobs through m Machines.

Replacement Models: Replacement of Items that Deteriorate whose maintenance costs increase with time without change in the money value. Replacement of items that fail suddenly: individual replacement policy, group replacement policy.

**UNIT IV**

Dynamic programming: Characteristics of dynamic programming. Dynamic programming approach for Priority Management employment smoothening, Stage Coach/Shortest Path and Reliability problems.

Games Theory: Competitive games, rectangular game, saddle point, minimax (maximin) method of optimal strategies, value of the game. Solution of games with saddle points, dominance principle. Rectangular games without saddle point – mixed strategy for 2 X 2 games.

**UNIT V**

Inventory models: Inventory costs. Models with deterministic demand – model (a) demand rate uniform and production rate infinite, model (b) demand rate non-uniform and production rate infinite, model (c) demand rate uniform and production rate finite.

Queuing Theory: Essential Features of a queuing system. Performance measures of a queuing system. Model 1:  $\{(M/M/1) : (\infty/FCFS)\}$  Single server, Unlimited Queue model. Model 2:  $\{(M/M/1) : (\infty/SIRO)\}$  Single server, Unlimited Queue model. Model III:  $\{(M/M/1) : (N/FCFS)\}$  Single server, Finite Queue model.

**TEXT BOOKS:**

1. J K Sharma., “Operations Research Theory & Applications 4e”, Macmillan India Ltd.
2. P. K. Gupta and D. S. Hira, “Operations Research”, S. Chand & co., 2007.

**REFERENCE BOOKS:**

1. Pradeep Prabhakar Pai, Operations Research – principles and Practice, Oxford University Press, 2012.
2. A.M. Natarajan, P. Balasubramani, A. Tamilarasi, “Operations Research”, Pearson Education.
3. P Sankara Iyer, ”Operations Research”, Tata McGraw-Hill, 2008.
4. N.V.S. Raju, “Operations Research”, HI-TECH, 2002.
5. Col. D. S. Cheema, “Operations Research”, Laxmi Publications Ltd., 2005.
6. F.S. Hillier, G.J. Lieberman, “Introduction to Operations Research – 8ed”, TMH.
7. H.S. Kasana & K.D. Kumar, “Introductory Operations Research – Theory and applications”, Springer, 2003, rp2005.
8. Billy E. Gillett, “Introduction to Operations Research – A Computer-Oriented Algorithmic Approach”, Tata McGraw-Hill, 1979, rp2004.
9. A.B.Rao, Operations Research, Jaico .
10. Ravindran,Phillips,Solberg, Operations Research, 2<sup>nd</sup> edition,Wiley India.
11. W.L.Winston, Operations Research, 4<sup>th</sup> edition,Cengage Learning.
12. R. Panneerselvam, “Operations Research”, PHI-2e, 2006, rp2008.
13. ANITHA H S, “Operations Research”, EXEL books, 2011.

**MASTER OF COMPUTER APPLICATIONS  
I YEAR II SEMESTER  
OPERATING SYSTEMS**

**MC2C50**

**UNIT I**

Operating System Introduction: Operating Systems objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special -Purpose Systems, Operating System services, User OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure, Virtual Machines.

**UNIT II**

Process and CPU Scheduling - Process concepts-The Process, Process States, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Dispatcher, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling.

Process Coordination – Process Synchronization, The Critical Section Problem, Peterson’s solution, Synchronization Hardware, Semaphores.

**UNIT III**

Memory Management and Virtual Memory - Logical & Physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing.

**UNIT IV**

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free-space Management.

**UNIT V**

Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling.

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

**TEXT BOOKS:**

1. Operating System Principles , Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley Student Edition
2. Operating Systems – Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

**REFERENCE BOOKS:**

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI
2. Operating Systems A concept-based Approach, 2nd Edition, D.M.Dhamdhare, TMH.
3. Principles of Operating Systems , B.L.Stuart, Cengage learning, India Edition.
4. Operating Systems, A.S.Godbole,2nd Edition, TMH
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems,S.Haldar and A.A.Aravind,Pearson Education.

**MASTER OF COMPUTER APPLICATIONS****I Year II Semester****MC2L10****DATA STRUCTURES THROUGH C++ LAB**

1. Write a C++ program to perform the following operations :
  - a) Create a Singly linked list of elements.
  - b) Search for a given element in the above list.
  - c) Delete an element from the above Singly linked list.
  - d) Display the contents of the list.
2. Write a C++ program to perform the following operations:
  - a) Create a doubly linked list of elements.
  - b) Delete an element from the above doubly linked list
  - c) Display the contents of the list.
3. Write C++ programs to implement the following using an array.
  - a) Stack ADT b) Queue ADT
4. Write C++ programs to implement the following using a singly linked list.
  - a) Stack ADT b) Queue ADT
5. Write a C++ program to convert i) a given infix expression into postfix form using stack and ii) evaluate the Postfix expression using stack.
6. Write C++ programs to implement the deque (double ended queue) ADT using
  - a) Singly linked list b) Doubly linked list c) an array.
7. Write C++ programs that use recursive functions to traverse the given binary tree in
  - a) Preorder b) Inorder and c) Postorder
8. Write a C++ program to perform the following operations:
  - a) Construct a binary search tree of elements.
  - b) Search for a key element in the above binary search tree.
  - c) Delete an element from the above binary search tree.

9. Write C++ programs for implementing the following Searching methods:

- a) Linear Search b) Binary Search

10. Write C++ programs for implementing the following sorting methods:

- a) Bubble Sort b) Selection Sort c) Insertion Sort

11. Write C++ programs for implementing the following sorting methods:

- a) Merge sort b) Quick sort c) Heap sort d) Radix sort.

12. Write a C++ program to perform the following operation:

- a) Insertion into a B-tree

13. Write a C++ program to implement all the functions of a dictionary (ADT)

using hashing.

14. Write a C++ program for implementing Knuth-Morris-Pratt pattern matching

algorithm.

15. Write C++ programs that use non-recursive functions to traverse the given

binary tree in

- a) Preorder b) inorder and c) postorder.

16. Write C++ programs for the depth first and breadth first traversals of a graph.

### **TEXT BOOKS :**

1. Data Structures A Pseudocode Approach with C++, India Edition, R.F.Gilberg  
and B.A.Forouzan, Cengage Learning.

2. Data structures with C++, J.R.Hubbard, Schaum's Outlines, TMH.

3. Data Structures and STL, W.J.Collins, Mc Graw Hill, International edition.

4. Data structures and Algorithms with OODesign patterns in C++, B.R.Priess, John Wiley & sons.

5. Advanced Data structures & Algorithms in C++, V.V.Muniswamy, Jaico Publishing House.

6. C++ for Programmers, P.J.Deitel and H.M.Deitel, Pearson/C++: How to Program, Deitel  
and Deitel, PHI.

7. Data structures via C++, A.M.Berman, Oxford University Press.



**MASTER OF COMPUTER APPLICATIONS**  
**I Year II Semester**  
**COMPUTER NETWORKS LAB**

**MC2L20**

**Sample Problems on Computer Networks (Use C/C++ Programming Language):**

1. Implement the data link layer framing methods such as character stuffing and bit stuffing.
2. Implement the Aloha protocols.
3. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP .
4. Implement Dijkstra's algorithm to compute the Shortest path through a graph.
5. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table at each node using distance vector routing algorithm.
6. Take an example subnet of hosts . Obtain broadcast tree for it.
7. Take a 64 bit plain text and encrypt the same using DES algorithm.
8. Write a program to break the above DES coding.
9. Using RSA algorithm, Encrypt text data and Decrypt the same

**MASTER OF COMPUTER APPLICATIONS**  
**I Year II Semester**  
**OPERATING SYSTEMS LAB**

**MC2L30**

**List of Sample Problems:**

1. Simulate FCFS CPU scheduling algorithm.
2. Simulate non preemptive SJF CPU scheduling algorithm.
3. Simulate preemptive SJF CPU scheduling algorithm.
4. Simulate non preemptive priority CPU scheduling algorithm.
5. Simulate preemptive priority CPU scheduling algorithm.
6. Simulate round robin CPU scheduling algorithm.
7. Simulate Bankers Algorithm for Dead Lock Avoidance.

**Additional Problems:**

**Note: Use Bash for Shell scripts.**

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a C program that makes a copy of a file using standard I/O and system calls.
11. Implement in C the following Unix commands using System calls
  - a). cat
  - b) mv
12. Write a C program to list files in a directory.
13. Write a C program to emulate the Unix ls -l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C program that redirects standard output to a file. Ex: ls > f1.
16. Write a C program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen.
17. Write a C program to create a Zombie process.
18. Write a C program that illustrates how an orphan is created.
19. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls -l | sort
20. Write C programs that illustrate communication between two unrelated processes using named pipe.
21. Write a C program in which a parent writes a message to a pipe and the child reads the message.

22. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
23. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (22)) and displays them.
24. Write a C programs to transfer a large amount of data between processes, using
  - a) a pipe
  - b)a FIFO
25. Write a C program to allow cooperating processes to lock a resource for exclusive use, using:
  - a)Semaphores
  - b)flock or lockf system calls.
26. Write a C program that illustrates suspending and resuming processes using signals.
27. Write a C program that implements a producer-consumer system with two processes. (using Semaphores).
28. Write client and server programs in C for interaction between server and client processes using Unix Domain sockets.
29. Write client and server programs in C for interaction between server and client processes using Internet Domain sockets.
30. Write C programs that illustrate two processes communicating using shared memory.

**TEXT BOOKS:**

1. Advanced Unix Programming, N.B.Venkateswarulu, BS Publications.
2. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.
3. Unix and Shell Programming, M.G. Venkatesh Murthy, Pearson Education, 2005.
4. Unix Shells by Example, 4th Edition, Ellie Quigley, Pearson Education.
5. Sed and Awk, O.Dougherty&A.Robbins, 2<sup>nd</sup> edition, SPD.
6. An Introduction to Operating Systems, P.C.P Bhatt, 2<sup>nd</sup> edition, PHI.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**

MC3C10

**DATABASE SYSTEMS****Objectives:**

By the end of the course, you will know:

- History and Structure of databases
- How to design a database
- How to convert the design into the appropriate tables
- Handling Keys appropriately
- Enforcing Integrity Constraints to keep the database consistent
- Querying relational data ,Triggers, Procedures and Cursors
- Normalizing the tables to eliminate redundancies
- Transaction Management
- Storage Optimizing Strategies for easy retrieval of data through index
- Processing the queries

**UNIT I**

Database System Applications, Purpose of Database Systems, View of Data – Data Abstraction, Instances and Schemas, Data Models – the ER Model, Relational Model, Other Models – Database Languages – DDL,DML, Database Access from Applications Programs, Transaction Management, Data Storage and Querying, Database Architecture, Database Users and Administrators, ER diagrams,. Relational Model: Introduction to the Relational Model – Integrity Constraints Over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design, Introduction to Views –Altering Tables and Views, Relational Algebra, Basic SQL Queries, Nested Queries, Complex Integrity Constraints in SQL, Triggers.

**UNIT II**

**Introduction to Schema Refinement** – Problems Caused by redundancy, Decompositions – Problem related to decomposition, Functional Dependencies - Reasoning about FDS, Normal Forms – FIRST, SECOND, THIRD Normal forms – BCNF –Properties of Decompositions- Loss less- join Decomposition, Dependency preserving Decomposition, Schema Refinement in Data base Design – Multi valued Dependencies – FOURTH Normal Form, Join Dependencies, FIFTH Normal form.

**UNIT III**

**Transaction Management:** The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control, Deadlocks – Performance of Locking – Transaction Support in SQL.

**Concurrency Control:** Serializability, and recoverability – Introduction to Lock Management – Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques – Concurrency Control without Locking.

**Crash recovery:** Introduction to Crash recovery, Introduction to ARIES, the Log, and Other Recovery related Structures, the Write-Ahead Log Protocol, Check pointing, recovering from a System Crash, Media recovery

#### UNIT IV

**Overview of Storage and Indexing:** Data on External Storage, File Organization and Indexing – Clustered Indexes, Primary and Secondary Indexes, Index data Structures – Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes and Performance Tuning

**Storing data: Disks and Files:** -The Memory Hierarchy – Redundant Arrays of Independent Disks.

Disk Space Management, Buffer Manager, Files of Records, Page Formats, Record Formats Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM)

B+ Trees: A Dynamic Index Structure, Search, Insert, Delete.

Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendable Vs Linear Hashing.

#### UNIT V

**Overview Of Query Evaluation:** The System Catalog, Introduction to Operator Evaluation, Algorithms for Relational Operations, Introduction to Query Optimization, Alternative Plans: A Motivating Example, What a Typical Optimizer Does?

**Evaluating Relational Operators:** The' Selection Operation, General Selection Conditions, The Projection Operation, The Join Operation, The Set Operations, Aggregate Operations, The Impact of Buffering.

**A Typical Relational Query Optimizer:** Translating SQL Queries into Algebra, Estimating the Cost of a Plan, Relational Algebra Equivalences, Enumeration of Alternative Plans, Nested Subqueries, The System R Optimizer, Other Approaches to Query Optimization.

#### TEXT BOOKS:

1. Data base Management Systems, Raghu Ramakrishnan, Johannes Gehrke, TMH, 3<sup>rd</sup> Edition. Database Systems: The Complete Book, Hector Garcia-Molina, Jeffrey Ullman, Jennifer Widom

#### REFERENCE BOOKS:

1. Database Systems implementation Hector Garcia-Molina, Jeffrey Ullman, Jennifer Widom
2. Introduction to Database Systems, C.J.Date, Pearson Education.
3. Data base System Concepts, A.Silberschatz, H.F. Korth, S.Sudarshan, McGraw hill, VI edition
4. Fundamentals of Database Systems , Ramez Elmasri, Shamkant B.Navathe, Pearson Education,
5. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, PHI.
6. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning, 2008.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**  
**NETWORK SECURITY**

**MC3C20****Objectives:**

- Understand the basic categories of threats to computers and networks
- Understand various cryptographic algorithms.
- Describe public-key cryptosystem.
- Describe the enhancements made to IPv4 by IPSec
- Understand Intrusions and intrusion detection
- Discuss the fundamental ideas of public-key cryptography.
- Generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
- Discuss Web security and Firewalls

**UNIT – I**

**Attacks on Computers and Computer Security:** Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security **Cryptography: Concepts and Techniques:** Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

**UNIT – II**

**Symmetric key Ciphers:** Block Cipher principles & Algorithms(DES, AES,Blowfish), Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers, RC4,Location and placement of encryption function, Key distribution **Asymmetric key Ciphers:** Principles of public key cryptosystems, Algorithms(RSA, Diffie-Hellman,ECC), Key Distribution

**UNIT – III**

**Message Authentication Algorithms and Hash Functions:** Authentication requirements, Functions, Message authentication codes, Hash Functions, Secure hash algorithm, Whirlpool, HMAC, CMAC, Digital signatures, knapsack algorithm.

**UNIT – IV**

**E-Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, combining security associations, key management

**UNIT – V**

**Web Security:** Web security considerations, Secure Socket Layer and Transport Layer Security, Secure electronic transaction **Intruders, Virus and Firewalls:** Intruders, Intrusion detection, password management, Virus and related threats, Countermeasures, Firewall design principles, Types of firewalls

**Case Studies on Cryptography and security:** Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability, Virtual Elections.

**TEXT BOOKS:**

1. Cryptography and Network Security : William Stallings, Pearson Education, 5<sup>th</sup> Edition
2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 2<sup>nd</sup> Edition.
3. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

**REFERENCES:**

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1<sup>st</sup> Edition.
2. Cryptography and Network Security : Forouzan Mukhopadhyay, Mc Graw Hill, 2<sup>nd</sup> Edition
3. Information Security, Principles and Practice : Mark Stamp, Wiley India.
4. Principles of Computer Security: WM.Arthur Conklin, Greg White, TMH
5. Introduction to Network Security: Neal Krawetz, CENGAGE Learning.
6. Principles of Information security by Michael E Whitman and Herbert J.Mattord.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**  
**DESIGN AND ANALYSIS OF ALGORITHMS**

**MC3C30**

**UNIT I**

Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, Probabilistic analysis, Amortized complexity.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's Matrix Multiplication.

**UNIT II**

Greedy method: General method, applications-Job sequencing with dead lines, 0/1 knapsack problem, Disjoint set operations, union and find algorithms, Spanning trees, Minimum cost spanning trees ,Prim's and Kruskal's algorithms, Single source shortest path problem.

**UNIT III**

Dynamic Programming: General method, applications-Multistage graphs, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Traveling sales person problem, Reliability design.

**UNIT IV**

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

**UNIT V**

Branch and Bound: General method, applications - Traveling sales person problem,0/1 knapsack problem-LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, Non-deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

**TEXT BOOKS:**

1. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Satraj Sahni and S.Rajasekharan, Universities Press,2008.
2. Foundations of Algorithms,4<sup>th</sup> edition,R.Neapolitan and K.Naimipour,Jones and Bartlett Learning.
3. Design and Analysis of Algorithms, P.H.Dave, H.B.Dave, Pearson Education,2008.

**REFERENCE BOOKS:**

1. Computer Algorithms, Introduction to Design and Analysis, 3rd Edition, Sara Baase, Allen, Van, Gelder, Pearson Education.
2. Algorithm Design: Foundations, Analysis and Internet examples, M.T.Goodrich and R.Tomassia, John Wiley and sons.



**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**  
**SOFTWARE ENGINEERING**

**MC3C40**

**UNIT I**

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, legacy software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, Specialized process models, The Unified process.

**UNIT II**

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

**UNIT III**

Design Engineering: Design process and Design quality, Design concepts, the design model, pattern based software design.

Creating an architectural design: software architecture, Data design, Architectural styles and patterns, Architectural Design, assessing alternative architectural designs, mapping data flow into a software architecture.

Modeling component-level design : Designing class-based components, conducting component-level design, Object constraint language, designing conventional components.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

**UNIT IV**

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Frame work for Product metrics, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

## UNIT V

Risk management: Reactive vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

### TEXT BOOKS:

1. Software Engineering :A practitioner's Approach, Roger S Pressman, sixth edition.  
McGrawHill International Edition, 2005
2. Software Engineering, Ian Sommerville, seventh edition, Pearson education, 2004.

### REFERENCE BOOKS:

1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
2. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3. Fundamentals of Software Engineering, Rajib Mall, PHI, 2005
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition , 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8. Software Engineering Principles and Practice, Hans Van Vliet, 3<sup>rd</sup> edition, John Wiley & Sons Ltd.
9. Software Engineering 3: Domains, Requirements, and Software Design, D. Bjorner, Springer International Edition.
10. Introduction to Software Engineering, R.J. Leach, CRC Press.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**  
**JAVA PROGRAMMING**

**MC3C50**

**UNIT I**

**Java Basics** - History of Java, Java buzzwords, comments, data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow-block scope, conditional statements, loops, break and continue statements, simple java program, arrays, input and output, formatting output, Review of OOP concepts, encapsulation, inheritance, polymorphism, classes, objects, constructors, methods, parameter passing, static fields and methods, access control, this reference, overloading methods and constructors, recursion, garbage collection, building strings, exploring string class, Enumerations, autoboxing and unboxing, Generics.

**UNIT II**

**Inheritance** – Inheritance concept, benefits of inheritance, Super classes and Sub classes, Member access rules, Inheritance hierarchies, super uses, preventing inheritance: final classes and methods, casting, polymorphism- dynamic binding, method overriding, abstract classes and methods, the Object class and its methods. **Interfaces** – Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interface. **Inner classes** – Uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples.

**Packages**-Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

**UNIT III**

**Data structures creation and manipulation in java** – Introduction to Java Collections, Overview of Java Collection frame work, Commonly used Collection classes– ArrayList, LinkedList, HashSet, HashMap, TreeMap, Collection Interfaces – Collection, Set, List, Map, Iteration over Collections – Iterator interface, ListIterator interface. **Files** – streams- byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class java.io.

**UNIT IV**

**Exception handling** – Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

**Multithreading** - Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, interthread communication.

**UNIT V**

**GUI Programming with Java** - The AWT class hierarchy, Introduction to Swing, Swing vs. AWT, MVC architecture, Hierarchy for Swing components, Containers – Top-level containers – JFrame, JApplet, JWindow, JDialog, Light weight containers – JPanel, A simple swing application, Overview of several swing components- JButton, JToggleButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JList, JComboBox, JMenu, Java's Graphics capabilities – Introduction, Graphics contexts and Graphics objects, color control, Font control, Drawing lines, rectangles and ovals, Drawing arcs, Layout management - Layout manager types – border, grid, flow, box. **Event Handling** - Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Semantic and Low-level events, Examples: handling a button click, handling mouse and keyboard events, Adapter classes.

**Applets** – Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet - Four methods of an applet, Developing applets and testing, passing parameters to applets, applet security issues..

**TEXT BOOKS :**

1. Java: the complete reference, 8<sup>th</sup> edition, Herbert Schildt, TMH.
2. Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education / Java: How to Program P.J.Deitel and H.M.Deitel ,8<sup>th</sup> edition, PHI.

**REFERENCE BOOKS :**

1. Java Programming, D.S.Malik, Cengage Learning.
2. Core Java, Volume 1-Fundamentals, eighth edition, Cay S.Horstmann and Gary Cornell, Pearson education.
3. An introduction to Java programming and object oriented application development, R.A. Johnson- Cengage Learning.
4. Advanced Programming in Java2, K.Somasundaram, Jaico Publishing House.
5. Programming in Java, S.Malhotra and S.Choudhary, Oxford Univ. Press.
6. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
7. Object Oriented Programming through Java, P.Radha Krishna, Universities Press.
8. An introduction to programming and OO design using Java, J.Nino, F.A.Hosch, John Wiley&Sons.
9. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
10. Maurach's Beginning Java2, D.Lowe, J.Murach, A.Stelman, SPD.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**  
**JAVA PROGRAMMING LAB.**

**MC3L10**

**List of Sample Problems:**

1. a) Write a Java program that prints all real solutions to the quadratic equation  $ax^2 + bx + c = 0$ . Read in a, b, c and use the quadratic formula. If the discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.
  - b) The Fibonacci sequence is defined by the following rule: The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.
2. a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
  - b) Write a Java program to multiply two given matrices.
  - c) Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)
3. Write a Java program to find both the largest and smallest number in a list of integers.
4. Write a Java program to illustrate method overloading.
5. Write a Java program that implements the Sieve of Eratosthenes to find prime numbers.
6. Write a Java program to sort a list of names in ascending order.
7. Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:
  - a) Reading a matrix.
  - b) Addition of matrices.
  - c) Printing a matrix.
  - d) Multiplication of matrices.
8. Write a Java Program to solve Towers of Hanoi problem .
9. Write a Java Program that uses a recursive function to compute ncr.  
 (Note: n and r values are given.)
10. Write a Java program to perform the following operations:
  - a) Concatenation of two strings.
  - b) Comparison of two strings.
11. Implement the complex number ADT in Java using a class. The complex ADT is used to represent complex numbers of the form  $c=a+ib$ , where a and b are real numbers. The operations supported by this ADT are:
  - a) Reading a complex number.

- b) Writing a complex number.
  - c) Addition of Complex numbers.
  - d) Multiplication of complex numbers.
12. Write a Java program that makes frequency count of letters in a given text.
13. Write a Java program that uses functions to perform the following operations :
- a) Inserting a sub-string in to the given main string from a given position.
  - b) Deleting n characters from a given position in a given string.
14. a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.  
b) Write a Java program to make frequency count of words in a given text.
15. a) Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.  
b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.  
c) Write a Java program that displays the number of characters, lines and words in a text file.  
d) Write a Java program to change a specific character in a file.
- Note: Filename , number of the byte in the file to be changed and the new character are specified on the command line.
16. Write a Java program that illustrates the following
- a) Creation of simple package.
  - b) Accessing a package.
  - c) Implementing interfaces.
17. a) Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.  
b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
18. Write Java programs that illustrates the following
- a) Handling predefined exceptions
  - b) Handling user defined exceptions
19. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, \*, % operations. Add a text field to display the result.
20. a) Develop an applet in Java that displays a simple message.  
b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

21. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.
22. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)
23. a) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.  
b) Write a Java program that allows the user to draw lines, rectangles and ovals.
24. a) Write a Java program to create an abstract class named Shape that contains an empty method named numberOfSides ( ). Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides ( ) that shows the number of sides in the given geometrical figures.  
b) Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a Java program to display the table using Jtable component.
25. Write a Java program for handling Key events.
26. Write a Java program for handling mouse events. (Use Adapter classes).
27. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:  
a) Linear search                      b) Binary search
28. Write Java programs to implement the List ADT using arrays and linked lists.
29. Write Java programs to implement the Stack ADT and Queue ADT using arrays.
30. Write Java programs to implement the following using a singly linked list.  
a) Stack ADT                      b) Queue ADT .
31. Write Java programs for implementing the following sorting methods:  
a) Bubble sort                      c) Quick sort  
b) Selection sort
32. Write a Java Program to perform the following:  
a) Create a binary search tree of elements.

- b) Search the above binary search tree for a key value.
- c) Traverse the above binary search tree in inorder.

**Sample problems on Algorithms:**

**Note: You may develop programs using Java or C++**

1. Write a program that implements Prim's algorithm to generate minimum cost spanning tree.
2. Write a program that implements Kruskal's algorithm to generate minimum cost spanning tree.
3. Write a program to implement Dijkstra's algorithm for Single source shortest path problem.

**TEXT BOOKS**

1. Data structures and Algorithms in java, 3<sup>rd</sup> edition, A. Drozdek, Cengage Learning.
2. Data structures with Java, J.R. Hubbard, 2<sup>nd</sup> edition, Schaum's Outlines, TMH.
3. Data structures and algorithms in Java, 2<sup>nd</sup> Edition, R. Lafore, Pearson Education.
4. Data Structures using Java, D.S. Malik and P.S. Nair, Cengage Learning.
5. Data structures, Algorithms and Applications in java, 2nd Edition, S. Sahani, Universities Press.
6. Data structures, Algorithms and Applications in C++, 2nd Edition, S. Sahani, Universities Press.
7. Data structures and Algorithm Analysis in C++, 2nd Edition, M.A. Weiss, Pearson education.



**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**

**MC3L20**

**NETWORK SECURITY LAB**

1. Write a C program that contains a string(char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and display the result.
2. Write a C program that contains a string(char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
  - a. Ceaser cipher
  - b. Substitution cipher
  - c. Hill Cipher
4. Write a C program to implement the DES algorithm logic.
5. Write a JAVA program to implement the DES algorithm logic.
6. Write a Java program that contains functions, which accept a key and input text to be encrypted/decrypted. This program should use the key to encrypt/decrypt the input by using the triple Des algorithm. Make use of Java Cryptography package.
7. Write a C/JAVA program to implement the Blowfish algorithm logic.
8. Write a C/JAVA program to implement the Rijndael algorithm logic.
9. Write the RC4 logic in Java
10. Using Java cryptography, encrypt the text "Hello world" using Blowfish. Create your own key using Java keytool.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**

MC3L30

**DATABASE SYSTEMS LAB.**

**List of Sample Problems:**

- 1) Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
- 2) Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.  
Example:- Select the roll number and name of the student who secured fourth rank in the class.
- 3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
- 4) Queries using Conversion functions (to\_char, to\_number and to\_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next\_day, add\_months, last\_day, months\_between, least, greatest, trunc, round, to\_char, to\_date)
- 5) i) Creation of simple PL/SQL program which includes declaration section, executable section and exception –handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)  
ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
- 6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
- 7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.
- 8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
- 9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
- 10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
- 11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
- 12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

**Example Problems:**

**1. Creating tables for various relations (in SQL):**

CLIENT\_MASTER (CLIENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE)

PRODUCT\_MASTER( PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNITMEASURE, QTY\_ON\_HAND, REORDERLVL, SELLPRICE, COSTPRICE)

SALESMAN\_MASTER (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, SLAMT, TGTTOGET, YTDSALES, RESALES)

SALES\_ORDER(OREDRENO, CLIENTNO, ORDERDATE, DELYADDR, SALESMANNO, DELYTYPE, BILLYN, DELYDATE, ORDERSTATUS)

SALES\_ORDER\_DETAILS(ORDERNO, PRODUCTNO, QTYORDERED, QTYDISP, PRODUCTRATE)

- i) Insert data into their respective table.
- ii) Exercise on retrieving records from table.
  - a. Find out the names of all clients.
  - b. Retrieve the entire contents of the client\_master table.
  - c. Retrieve the list of names, city and the state of all clients.
  - d. List the various products available from the Product\_master table.
  - e. List all the clients who are located in Mumbai.
  - f. Find the names of salesmen who have a salary equal to Rs.3000.
- iii. Exercise on updating records in a table
  - a. Change the city of client No 'C00005' to 'Bangalore'.
  - b. Change the BalDue of ClientNo 'C00001' to Rs.1000.
  - c. Change the cost price of 'Trousers' to Rs.950.00.
  - d. Change the city of the salesman to Pune.
- iv. Exercise on deleting records in a table
  - a. Delete all sales men from the salesman\_master whose salaries are equal to Rs.35000
  - b. Delete all products from product\_master where the quantity on hand is equal to 100.
  - c. Delete from Client\_Master where the column state holds the value 'Tamilnadu'.
- v. Exercise on altering the table
  - a. Add a column called 'Telephone' of data type 'number' and size='10' to the Client Master table.
  - b. Change the size of Sell Price column in Product\_Master to 10,2.
- vi. Exercise on deleting the table structure along with the data
  - a. Destroy the table Client\_Master along with the data.

vii. Exercise on renaming the table

- a. Change the name of the Salesman\_Master table to Sman\_mast.

2. Using the tables created previously generate the SQL statements for the operations

mentioned below. The tables in user are as follows:

Client\_Master

Product\_Master

Salesman\_Master

Sales\_Order

Sales\_Order\_Details

i) Perform the following computations on table data:

- a. List the names of all clients having 'a' as the second letter in their names.
- b. List the clients who stay in a city whose first letter is 'M'.
- c. List all clients who stay in 'Bangalore' or 'Mangalore'.
- d. List all clients whose BalDue is greater than value 10000.
- e. List all information from the Sales\_Order table for orders placed in the month of June.
- f. List the order information for Client No 'C00001' and 'C00002'.
- g. List products whose selling price is greater than 500 and less than or equal to 750.
- h. List products whose selling price is more than 500. Calculate a new selling price as, original selling price \* .15. Rename the new column in the output of the above query as new\_price.
- i. List the names, city and state of clients who are not in the state of 'Maharashtra'.
- j. Count the total no of orders.
- k. Calculate the average price of all the products.
- l. Determine the maximum and minimum products prices. Rename the output as max\_price and min\_price respectively.
- m. Count the no of products having price less than or equal to 500.
- n. List all the products whose Qty On Hand is less than reorder level.

ii) Exercise on Date Manipulation

- a. List the order number and day on which clients on placed their order.
- b. List the months (in alphabets) and date when the orders must be delivered.
- c. List the Order Date in the format 'DD-Month-YY'. e.g.12-February-02.
- d. List the date, 15 days after today's date.

iii). Exercises on using Having and Group by Clauses:

- a. Print the description and total qty sold for each product.
- b. Find the value of each product sold
- c. Calculate the average qty sold for each client that has a maximum order value of 15000.00.
- d. Find out the total of all the billed orders for the month of June.

iv). Exercises on Joins and Correlation:

- a) Find out the products, which have been sold to 'Ivan Bay Ross'.

- b) Find out the products and their quantities that will have to be delivered in the current month.
  - c) List the product no and description of constantly sold products (i.e. rapidly moving products).
  - d) Find the names of clients who have purchased 'Trousers'.
  - e) List the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.
  - f) Find the products and their quantities for the orders placed by 'Ivan Bay Ross' and 'Mamta Muzumdar'.
  - g) Find the products and their quantities for the orders placed by Client No 'C00001' and 'C00002'.
- v). Exercise on Sub-queries:
- a. Find the Product No and Description of non\_moving products i.e. Products not being sold.
  - b. List the customer Name Address1, Address2, City and Pin Code for the client who has placed order no 'O19001'.
  - c. List the client names that have placed orders before the month of May'02.
  - d. List if the product 'Lycra Top' has been ordered by any client and print the Client\_no, Name to whom it was sold.
  - e. List the names of clients who have placed orders worth Rs.10,000 or more.

### 3) Creating Views

### 4) Writing Assertions

### 5) Writing Triggers

### 6) Implementing Operations on relations (tables) using PL/SQL

Ex: Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named Areas(radius,area).

### 7) Creating FORMS.

### 8) Generating REPORTS.

## **Additional Problems:**

### **i) Databases :**

**Objective:** This lab enables the students to practice the concepts learnt in the subject Databases by developing a database for an example company named "Roadway Travels" whose description is as follows. The student is expected to practice the designing, developing and querying a database in the context of example database "Roadway travels". Students are expected to use "Mysql" database.

### **Roadway Travels**

"Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its

main office is located in Hyderabad.

The company wants to *computerize its operations* in the following areas:

- Reservations and Ticketing
- Cancellations

### **Reservations & Cancellation:**

Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family).

Cancellations are also directly handed at the booking office.

In the process of *computerization* of **Roadway Travels** you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 3. Relational Model 4. Normalization 5. Creating the database 6. Querying. *Students are supposed to work on these steps week wise and finally create a complete "Database System" to Roadway Travels.* Examples are given at every experiment for guidance to students.

### **Experiment 1: E-R Model**

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example: **Entities:**

1. BUS
2. Ticket
3. Passenger

#### **Relationships:**

1. Reservation
2. Cancellation

#### **PRIMARY KEY ATTRIBUTES:**

1. Ticket ID (Ticket Entity)
2. Passport ID (Passenger Entity)
3. Bus\_NO (Bus Entity)

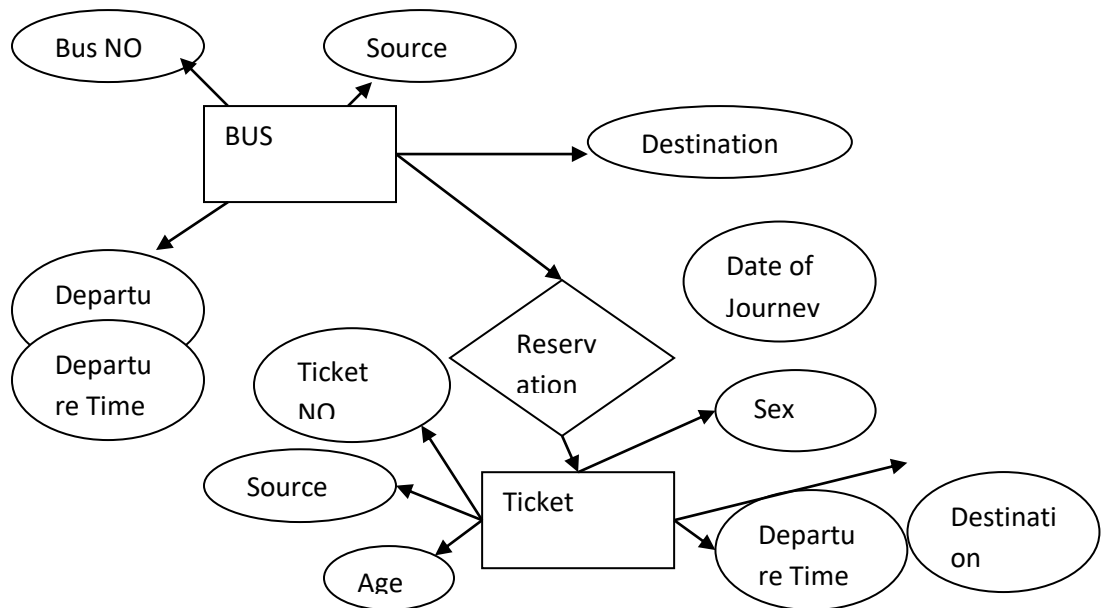
Apart from the above mentioned entities you can identify more. The above mentioned are few.

**Note:** *The student is required to submit a document by writing the Entities and Keys to the lab teacher.*

## Experiment 2: Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

### Example: E-R diagram for bus



**Note:** The student is required to submit a document by drawing the E-R Diagram to the lab teacher.

## Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

Passenger

Name	Age	Sex	Address	<u>Passport ID</u>
				Ticket_id

---

**Note:** *The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.*

#### **Experiment 4: Normalization**

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

For the above table in the First normalization we can remove the multi valued attribute Ticket\_id and place it in another table along with the primary key of passenger.

**First Normal Form: The above table can be divided into two tables as shown below.**

Passenger

Name	Age	Sex	Address	<u>Passport ID</u>

<u>Passport ID</u>	Ticket_id

You can do the second and third normal forms if required. Any how Normalized tables are given at the end.



### Experiment 5: Installation of Mysql and practicing DDL commands

Installation of MySQL. In this week you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized “Passenger” table.

```
CREATE TABLE Passenger (
  Passport_id INTEGER PRIMARY KEY,
  Name VARCHAR (50) Not NULL,
  Age Integer Not NULL,
  Sex Char,
  Address VARCHAR (50) Not NULL);
```

Similarly create all other tables.

**Note: Detailed creation of tables is given at the end.**

### Experiment 6: Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

#### Inserting values into “Bus” table:

Insert into Bus values (1234,'hyderabad', 'tirupathi');

Insert into Bus values (2345,'hyderabad', 'Banglore');

Insert into Bus values (23,'hyderabad', 'Kolkata');

Insert into Bus values (45,'Tirupathi', 'Banglore');

Insert into Bus values (34,'hyderabad', 'Chennai');

#### Inserting values into “Passenger” table:

Insert into Passenger values (1, 45,'ramesh', 45,'M', 'abc123');

Insert into Passenger values (2, 78,'geetha', 36,'F', 'abc124');

Insert into Passenger values (45, 90,'ram', 30,'M', 'abc12');

Insert into Passenger values (67, 89,'ravi', 50,'M', 'abc14');

Insert into Passenger values (56, 22, 'seetha', 32, 'F', 'abc55');

### Few more Examples of DML commands:

Select \* from Bus; (selects all the attributes and display)

UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;

### Experiment 7: Querying

In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

#### Practice the following Queries:

1. Display unique PNR\_no of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.
5. Find the names of passengers whose age is between 30 and 45.
6. Display all the passengers names beginning with 'A'
7. Display the sorted list of passengers names

### Experiment 8 and Experiment 9: Querying (continued...)

You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN),

GROUP BY, HAVING and Creation and dropping of Views.

1. Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR\_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR\_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. **Hint:** Use GROUP BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

### Experiment 10: Triggers

In this week you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

Eg: **CREATE TRIGGER updcheck BEFORE UPDATE ON passenger**

**FOR EACH ROW**

```

BEGIN
  IF NEW.TicketNO > 60 THEN
    SET New.Ticket no = Ticket no;
  ELSE
    SET New.Ticketno = 0;
  END IF;
END;

```

### Experiment 11: Procedures

In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

**Eg:CREATE PROCEDURE myProc()**

```

BEGIN
SELECT COUNT(Tickets) FROM Ticket WHERE age>=40;
End;

```

### Experiment 12: Cursors

In this week you need to do the following: Declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done

```

CREATE PROCEDURE myProc(in_customer_id INT)
BEGIN
  DECLARE v_id INT;
  DECLARE v_name VARCHAR(30);
  DECLARE c1 CURSOR FOR SELECT stdId,stdFirstname FROM students WHERE
stdId=in_customer_id;

```

```

OPEN c1;
FETCH c1 into v_id, v_name;
Close c1;
END;

```

## **Tables**

### **BUS**

Bus No: Varchar: PK (public key)

Source : Varchar

Destination : Varchar

### **Passenger**

PPNO: Varchar(15)) : PK

Name: Varchar(15)

Age : int (4)

Sex:Char(10) : Male / Female

Address: VarChar(20)

### **Passenger\_Tickets**

PPNO: Varchar(15)) : PK

Ticket\_No: Numeric (9)

### **Reservation**

PNR\_No: Numeric(9) : FK

Journey\_date : datetime(8)

No\_of\_seats : int (8)

Address : Varchar (50)

Contact\_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer

Status: Char (2) : Yes / No

Cancellation

PNR\_No: Numeric(9) : FK

Journey\_date : datetime(8)

No\_of\_seats : int (8)

Address : Varchar (50)

Contact\_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer

Status: Char (2) : Yes / No

### **Ticket**

Ticket\_No: Numeric (9): PK

Journey\_date : datetime(8)

Age : int (4)

Sex:Char(10) : Male / Female

Source : Varchar

Destination : Varchar

Dep\_time : Varchar

### **TEXT BOOKS:**

- 1.Introduction to SQL,Rick F.Vander Lans,Pearson education.
- 2.Oracle PL/SQL, B.Rosenzweig and E.Silvestrova,Pearson education.
- 3.Oracle PL/SQL Programming,Steven Feuerstein,SPD.
- 4.SQL & PL/SQL for Oracle 10g,Black Book,Dr.P.S.Deshpande,Dream Tech.
- 5.Oracle Database 11g PL/SQL Programming,M.Mc Laughlin,TMH.
- 6.SQL Fundamentals,J.J.Patrick,Pearson Education.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year I Semester**  
**MC3A10 PROFESSIONAL COMMUNICATION SKILLS**

**Course Objectives:**

- ☒ To teach the four language skills - Listening, Speaking, Reading and Writing; critical thinking skills to students.
- ☒ To enable students comprehend the concept of communication.
- ☒ To help students cultivate the habit of Reading and develop their critical reading skills.

**Course Outcomes:**

- ☒ Students are trained to convert the conceptual understanding of communication into every day practice.
- ☒ Students are expected to be ready for placements.
- ☒ Students are prepared to communicate their ideas relevantly and coherently in professional writing.

**UNIT I**

**INTRODUCTION:** Basics of Communication - Principles of Communication - Types of Communication – Stages of Communication – Verbal and Non-verbal Communication – Channels of Communication – Barriers to Effective Communication – Formal and Informal Expressions in Various Situations.

**UNIT II**

**READING & STUDY SKILLS:** Reading Comprehension – Reading Strategies - Skimming and Scanning- Intensive and Extensive Reading– Unknown Passage for Comprehension - Critical Reading of Short Stories – Study Skills – Note Making – Summarizing – Articles and Prepositions – Synonyms and Antonyms

**UNIT III**

**WRITING SKILLS:** Difference between Spoken and Written Communication- Features of Effective Writing - Formation of a Sentence – SVOs and SVOC patterns – Types of sentences- Common errors in Writing - Writing coherent sentences using connectives and conjunctions- Written Presentation Skills – Tenses – Concord – Question Tags - Practice Exercises - One Word Substitutes – Words Often Confused and Misspelt.

**UNIT IV**

**PROFESSIONAL WRITING:** Letter writing – Types, Parts and Styles of Formal Letters – Language to be used in Formal Letters – Letters of Enquiry, Complaint, and Apology with Replies – Letter of Application -Resume – E-mail – Active and Passive Voice.

**UNIT V**

REPORT WRITING: Types of Reports – Formats of Reports – Memo Format – Letter Format and Manuscript Format- Parts of Technical Report – Informational, Analytical and Project Reports – Idioms and Phrases.

**REFERENCE BOOKS:**

1. Meenakshi Raman & Sangeetha Sharma. 2012. *Technical Communication*. New Delhi: OUP
2. Rizvi, M. A. 2005. *Effective Technical Communication*. New Delhi: Tata McGraw Hill
3. Sanjay Kumar & Pushp Latha. 2012. *Communication Skills*. New Delhi: OUP
4. Er. A. K. Jain, Dr. Pravin S. R. Bhatia & Dr. A. M. Sheikh. 2013. *Professional Communication Skills*. S. Chand Publishers. New Delhi.
5. Farhathullah, T.M. 2009. *English for Business Communication*. Bangalore: Prism Publishers
6. Bikram K Das. 2011. *Functional Grammar and Spoken and Written Communication in English*. Kolkata: Orient Blackswan
7. Kiranmai Dutt, P et al. 2011. *A Course in Communication Skills*. New Delhi: CUP India
8. Krishnaswamy, N. 2000. *Modern English – A Book of Grammar, Vocabulary and Usage*. Macmillan India Pvt. Ltd
9. Ramachandran, K K. et al. 2007. *Business Communication*. New Delhi: Macmillan
10. Taylor, Ken. 2011. *50 ways to improve your Business English*. Hyderabad: Orient Blackswan

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**CORE-I INTERNET TECHNOLOGIES**

**MC4C10**

**Unit I:**

**HTML:**

Common tags- List, Tables, Images, Forms, Frames and IFrames, Cascading Style Sheets; Introduction to Java Scripts, Dynamic HTML with Java Script.

**XML:**

Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, DOM and SAX Parsers, XHTML

**Unit II:**

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads, Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies

**Unit III:**

Object Oriented Programming with PHP: Creating classes and objects, public, private and protected access, constructor and destructor, Inheritance, Overriding and overloading methods, calling base class methods, static members, interfaces, abstract classes

File Handling: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories

XML Processing: creating an XML file, using simple XML functions, extracting elements and attributes, modifying XML elements and attributes, adding and deleting elements and attributes, parsing XML file with parser functions .Simple Ajax implementation with PHP

**Unit IV:**

Introduction to Servlets: Lifecycle of a Servlet, The Servlet API, The javax.servlet Package, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions.

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages. Using Cookies-Session Tracking, Security Issues ,Database Access: Using JDBC to access Database from JSPs and Servlets

**Unit V:**



Introduction to MVC architecture, Anatomy of a simple struts application, struts-config.xml file, Presentation layer with JSP, Struts Controller class, JSP bean, html and logic tag libraries, ActionForms, DynaActionForm, Actions, Forwarding, Error Handling, Database Connection Pooling, validation framework and examples for simple data types, Internationalization

#### **TEXT BOOKS:**

1. Web Programming, building internet applications, Chris Bates 2<sup>nd</sup> edition,  
WILEY Dreamtech (UNIT 1)
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill (Unit 2,3)
3. Java Server Pages –Hans Bergsten, SPD O’Reilly (UNITs 3,4,5)
4. The World of Scripting Languages , David Barron,Wiley Publications.
5. Professional Jakarta Struts - James Goodwill, Richard Hightower, Wrox Publishers.

#### **REFERENCE BOOKS:**

1. Programming world wide web,R.W.Sebesta,Fourth edition,Pearson.
2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES , Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program , Dietel and Nieto,Pearson.
4. Jakarta Struts Cookbook , Bill Siggelkow, S P D O’Reilly.
5. Professional Java Server Programming,S.Allamaraju and othersApress(dreamtech).
6. Java Server Programming ,Ivan Bayross and others,The X Team,SPD
7. Web Warrior Guide to Web Programmimg-Bai/Ekedaw-Thomas
8. Beginning Web Programming-Jon Duckett WROX.
9. Java Server Pages, Pekowsky, Pearson.
10. Java Script,D.Flanagan,O’Reilly,SPD.

## MASTER OF COMPUTER APPLICATIONS

### II year II semester

#### MC4C20 CORE-II SOFTWARE TESTING METHODOLOGIES

##### UNIT-I

Introduction:- Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs. Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

##### UNIT-II

Transaction Flow Testing:-transaction flows, transaction flow testing techniques. Dataflow testing:- Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

##### UNIT-III

Domain Testing:-domains and paths, Nice & ugly domains, domain testing, domains and interfaces, testing, domain and interface testing, domains and testability.

Paths, Path products and Regular expressions:- path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

##### UNIT-IV

Logic Based Testing:- overview, decision tables, path expressions, kv charts, specifications.

State, State Graphs and Transition testing:- state graphs, good & bad state graphs, state testing, Testability tips.

##### UNIT-V

Graph Matrices and Application:-Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. ( Student should be given an exposure to a tool like JMeter or Win-runner).

##### TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.

##### REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing, 3<sup>rd</sup> edition, P.C.Jorgensen, Aurbach Publications(Dist.by SPD).
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley, 2<sup>nd</sup> Edition, 1999.
5. Art of Software Testing – Meyers, John Wiley.
6. Software Testing, N.Chauhan, Oxford University Press.
7. Software Testing, M.G.Limaye, TMH.
8. Software Testing, S.Desikan, G.Ramesh, Pearson.
9. Foundations of Software Testing, D.Graham & Others, Cengage Learning.
10. Foundations of Software Testing, A.P.Mathur, Pearson.

**MASTER OF COMPUTER APPLICATIONS**

**II Year II Semester**

**MC4C30**

**CORE-III**

**DATA MINING**

**Objectives:**

- To understand data mining concepts.
- To learn about various data preprocessing techniques.
- To learn about data warehousing.
- To learn about various data mining functionalities such as association rule mining, clustering, classification and outlier analysis.

**UNIT I**

**Introduction:** Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Integration of a Data Mining System with a Database or a Data Warehouse System, Issues in Data Mining.

**Data Preprocessing:** Need for Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

**UNIT II**

**Data Warehouse and OLAP Technology for Data Mining:** Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Usage of Data Warehousing Online Analytical Processing and Mining

**Data Cube Computation:** Efficient Methods for simple Data Cube Computation (Full Cube, Iceberg Cube, Closed Cube and Shell Cube), Discovery Driven exploration of data cubes, Attribute-Oriented Induction for data characterization and its implementation

**UNIT III**

**Mining Frequent Patterns, Associations and Correlations:** Basic Concepts, The Apriori algorithm for finding frequent itemsets using candidate generation, Generating association rules from frequent itemsets, Mining frequent itemsets without candidate generation, Mining various kinds of Association Rules, Correlation Analysis

**UNIT IV**

**Classification and Prediction:** Description and comparison of classification and prediction, preparing data for Classification and Prediction

Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Backpropagation

Prediction, linear and non-linear regression, evaluating accuracy of a Classifier or a Predictor

**UNIT V**

**Cluster Analysis:** Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, k-means and k-medoids methods, CLARANS, Agglomerative and divisive hierarchical clustering, chameleon dynamic modeling, DBSCAN, Grid based clustering method: STING, Conceptual Clustering, Constraint-Based Cluster Analysis, Outlier Analysis.

**TEXT BOOKS:**

1. Data Mining – Concepts and Techniques - Jiawei Han, Micheline Kamber and Jian Pei, 3<sup>rd</sup> edition, Morgan Kaufmann Publishers, ELSEVIER.
2. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.

**REFERENCES:**

1. Data Warehousing in the Real World – Sam Aanhory & Dennis Murray Pearson Edn Asia.
2. Insight into Data Mining, K.P.Soman, S.Diwakar, V.Ajay, PHI, 2008.
3. Data Warehousing Fundamentals – Paulraj Ponnaiah Wiley student Edition
4. The Data Warehouse Life cycle Tool kit – Ralph Kimball Wiley student edition
5. Building the Data Warehouse By William H Inmon, John Wiley & Sons Inc, 2005.
6. Data Mining Introductory and advanced topics –Margaret H Dunham, Pearson education
7. Data Mining Techniques – Arun K Pujari, 2<sup>nd</sup> edition, Universities Press.
8. Data Mining, V.Pudi and P.Radha Krishna, Oxford University Press.
9. Data Mining: Methods and Techniques, A.B.M Shawkat Ali and S.A.Wasimi, Cengage Learning.
10. Data Warehouse 2.0, The Architecture for the next generation of Data Warehousing, W.H.Inmon, D. Strauss, G.Neushloss, Elsevier, Distributed by SPD.

**MASTER OF COMPUTER APPLICATIONS****II Year II Semester****MC4011****SOFT COMPUTING****(OPEN ELECTIVE-I)**

To give students knowledge of soft computing theories fundamentals, i.e. Fundamentals of artificial and neural networks, fuzzy sets and fuzzy logic and genetic algorithms.

**UNIT-I**

AI Problems and Search: AI problems, Techniques, Problem Spaces and Search, Heuristic Search Techniques- Generate and Test, Hill Climbing, Best First Search Problem reduction, Constraint Satisfaction and Means End Analysis. Approaches to Knowledge Representation- Using Predicate Logic and Rules.

**UNIT-II**

Artificial Neural Networks: Introduction, Basic models of ANN, important terminologies, Supervised Learning Networks, Perceptron Networks, Adaptive Linear Neuron, Backpropagation Network. Associative Memory Networks. Training Algorithms for pattern association, BAM and Hopfield Networks.

**UNIT-III**

Unsupervised Learning Network- Introduction, Fixed Weight Competitive Nets, Maxnet, Hamming Network, Kohonen Self-Organizing Feature Maps, Learning Vector Quantization, Counter Propagation Networks, Adaptive Resonance Theory Networks. Special Networks-Introduction to various networks.

**UNIT-IV**

Introduction to Classical Sets ( crisp Sets)and Fuzzy Sets- operations and Fuzzy sets. Classical Relations - and Fuzzy Relations- Cardinality, Operations, Properties and composition. Tolerance and equivalence relations. Membership functions- Features, Fuzzification, membership value assignments, Defuzzification.

**UNIT-V**

Fuzzy Arithmetic and Fuzzy Measures, Fuzzy Rule Base and Approximate Reasoning Fuzzy Decision making, Fuzzy Logic Control Systems. Genetic Algorithm- Introduction and basic operators and terminology. Applications: Optimization of TSP, Internet Search Technique

**TEXT BOOKS:**

- 1 Principles of Soft Computing- S N Sivanandam, S N Deepa, Wiley India, 2007.
- 2 Soft Computing and Intelligent System Design -Fakhreddine O Karray, Clarence D Silva, Pearson Edition, 2004.

**REFERECE BOOKS :**

1. Artificial Intelligence and SoftComputing- Behavioural and Cognitive Modelling of the Human Brain- Amit Konar, CRC press, Taylor and Francis Group.
2. Artificial Intelligence – Elaine Rich and Kevin Knight, TMH, 1991, rp2008.
3. Artificial Intelligence – Patric Henry Winston – Third Edition, Pearson Education.
4. A first course in Fuzzy Logic-Hung T Nguyen and Elbert A Walker, CRC. Press Taylor and Francis Group.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**DISTRIBUTED SYSTEMS**  
**(OPEN ELECTIVE- I)**

**MC4012**

Objectives:

- To explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are;
- To list the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions;
- To recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems;

**UNIT-I**

Characterization of Distributed Systems. Design Issues, User Requirement, Network Technologies and Protocols, IPC, Client-Server Communication, Group Communication, IPC in UNIX. Remote Procedure Calling, Design issues, Implementation, Asynchronous RPC

**UNIT-II**

Distributed OS, Its kernel, Processes and Threads, Naming and Protection, Communication and Invocation, Virtual Memory, File Service components, Design issues, Interfaces, implementation techniques, SUN network file systems

**UNIT-III**

SNS – a name service model, its design issues, Synchronizing physical clocks, Logical time and logical clocks, Distributed coordination. Replication and its architectural model, Consistency and request ordering, Conversation between a client and a server, Transactions, Nested Transactions.

**UNIT-IV**

Concurrency control Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control.

Distributed Transactions and Nested Transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed Deadlocks, Transactions with replicated data, Transaction recovery, Fault tolerance, Hierarchical and group masking of faults.

**UNIT-V**

Cryptography, Authentication and key distribution, Logics of Authentication, Digital signatures.

Distributed shared memory, Design and Implementation issues, Sequential consistency and ivy, Release consistency and Munin, Overview of Distributed Operating systems Mach, Chorus.

**TEXT BOOKS:**

1. G Coulouris, J Dollimore and T Kindberg - Distributed Systems Concepts and Design, Third Edition, Pearson Education.

**REFERENCES:**

1. M Singhal, N G Shivarathri - Advanced Concepts in Operating Systems, Tata McGraw Hill Edition.
2. A.S. Tanenbaum and M.V. Steen - Distributed Systems – Principles and Paradigms, Pearson education.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**

**MC4013 ORGANIZATION STRUCTURE AND PERSONNEL MANAGEMENT**

**(OPEN ELECTIVE-I)**

**Objectives:**

One of the main objectives of this course is to provide the students with an in depth understanding of organizations. Organizations are a main tool that modern society applies to meet the challenges of innovation and resolution of complex tasks. To understand how to organize to resolve the challenges organizations meet is a main focus of this course. The course reviews the main contributions to organization research.

This is an introductory course in Organization Theory. In this course we will explore the basic theories and principles around which contemporary organizations are structured in complex, dynamic, uncertain, and competitive environments. The course examines the effects of both the internal and external environment on managerial choices for the structuring of organizations, as well as the implications that stem from these choices.

This course also covers Personnel Management and Communication. In this course we will explore the basic theories of Personnel Management and its functions. The object of the study is to do the analysis of various functions of the organization in functional area of Personnel Management so that the student should be in a position to create data bases and programmes in the content of Personnel Department.

**Outcomes:**

The course will help each student to better:

- Understand the importance of organizational structure and design on internal organizational processes and overall effectiveness.
- Understand the relationships between organization structure and the behavior of those who work in them or otherwise interact with them.
- Recognize the managerial implications of organization design and change and how these are informed by the relevant theories.
- Appreciate the impact of advanced technologies on the strategy and structure of organizations and how to address the changes implied by the adoption of these technologies.
- Acquire the knowledge and skills needed to analyze the design and structure of organizations through a combination of lectures, discussions, and cases.
- Understand the Personnel Functions like position of the personnel department in the organization.
- Understand manpower planning, job description, interviewing techniques, transfers, promotion and its policies.
- Understand the training and development and career planning and Performance Appraisal.
- Obtain and practice effective written and oral business communications skills.

**UNIT I**

Classical Theories of organization : Functional approach, classical theories of organization, division of labour, levels of authority, span of control, authority & responsibility, efficiency of management.



Behavioral theories of organization, limitations of formal organization, human relation, group behavior, committee and group making, motivation and morale.

### **UNIT II**

Personnel Function: Evaluation, objectives, principles, philosophies and policies, duties & responsibilities of the manager, position of the personnel department in the organization, line and staff relationship & the changing concept of personnel management in India.

### **UNIT III**

Manpower planning : Uses benefits problems and limitations, manpower inventory, manpower forecasting, job description, recruitment, Job specification and job selection, interviewing techniques, transfers, promotion and its policies.

Training and development : Objectives and policies planning, organizing the training department, training manager and his job, on and off the job training, techniques, career planning, objectives of performance appraisal.

### **UNIT IV**

Strategic management: Objectives, importance policies, concept of core competence capability of organizational learning, strategic levels and planning, business level strategy and functional level, PHASES OF PLANNING, SWOT, develop strategies and prepare strategic plan.

### **UNIT V**

Communication: Importance of communication, inter personnel communication barriers of communication, communication in organizations, using communication skills to manage conflicts. Impact of informational technology and fostering effective communication

### **TEXT BOOKS:**

1. L.M.Prasad, Principles and Practice of Management, Sultan Chand & Sons.
2. A.R.Aryasri, Organizational Structure and Personnel Management, TMH, 2009

### **REFERENCE BOOKS:**

1. Hellriegel, Jackson and Slocum, Edition 9, Management-A competency – Based Approach
2. L.M.Prasad, Human Resource Management.



**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**COMPUTER FORENSICS**  
**(OPEN ELECTIVE-I)**

MC4015

**Objectives:**

To understand the cyberspace

To understand the forensics fundamentals

To understand the evidence capturing process.

To understand the preservation of digital evidence.

**UNIT I**

**Computer Forensics Fundamentals:** Introduction to Computer Forensics, Use of Computer Forensics in Law Enforcement, Computer Forensics Assistance to Human Resources/Employment Proceedings, Computer Forensics Services, Benefits of Professional Forensics Methodology, Steps Taken by Computer Forensics Specialists, Who Can Use Computer Forensic Evidence?. **Types of Computer Forensics Technology :** Types of Military Computer Forensic Technology, Types of Law Enforcement Computer Forensic Technology, Types of Business Computer Forensics Technology.

**UNIT II**

**Computer Forensics Evidence and Capture: Data Recovery:** Data Recovery Defined, Data Backup and Recovery, The Role of Backup in Data Recovery, The Data-Recovery Solution, Case Histories. **Evidence Collection and Data Seizure:** Why Collect Evidence?, Collection Options, Obstacles, Types of Evidence, The Rules of Evidence, Volatile Evidence, General Procedure, Collecting and Archiving, Methods of Collection, Artifacts, Collection Steps, Controlling Contamination: The Chain of Custody.

**UNIT III**

**Duplication and Preservation of Digital Evidence:** Preserving the Digital Crime Scene, Computer Evidence Processing Steps, Legal Aspects of Collecting And Preserving Computer Forensic Evidence. **Computer Image Verification and Authentication :** Special Needs of Evidential Authentication, Practical Considerations, Practical Implementation.

**UNIT IV**

**Computer Forensics Analysis: Discovery of Electronic Evidence:** Electronic Document Discovery: A Powerful New Litigation Tool, **Identification of Data:** Timekeeping, Time Matters, Forensic Identification and Analysis of Technical Surveillance Devices. **Reconstructing Past Events:** How to

Become a Digital Detective, Useable File Formats, Unusable File Formats, Converting Files. **Networks:** Network Forensics Scenario, A Technical Approach, Destruction of Email, Damaging Computer Evidence, International Principles Against Damaging of Computer Evidence, Tools Needed for Intrusion Response to the Destruction of Data, Incident Reporting and Contact Forms.

## UNIT V

**Current Computer Forensics Tools:** Evaluating Computer Forensics Tool Needs, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software.

### TEXT BOOKS:

---

1. “Computer Forensics : Computer Crime Scene Investigation”, JOHN R. VACCA, Firewall Media.
  2. “Guide to Computer Forensics and Investigations”4e, Nelson, Phillips Enfinger, Steuart, Cengage Learning.
- 

### REFERENCES:

---

1. “Computer Forensics and Cyber Crime”, Marjie T Britz, Pearson Education.
2. “Computer Forensics”, David Cowen, Mc Graw Hill.
3. Brian Carrier , "File System Forensic Analysis" , Addison Wesley, 2005
4. Dan Farmer & Wietse Venema , "Forensic Discovery", Addison Wesley, 2005
5. Eoghan Casey , —Digital Evidence and Computer Crime —, Edition 3, Academic Press, 2011
6. Chris Pogue, Cory Altheide, Todd Haverkos ,Unix and Linux Forensic Analysis DVD ToolKit, Syngress Inc. , 2008
7. Harlan Carvey ,Windows Forensic Analysis DVD Toolkit, Edition 2, Syngress Inc. , 2009
8. Harlan Carvey ,Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry , Syngress Inc, Feb 2011
9. Eoghan Casey, Handbook of Digital Forensics and Investigation, Academic Press, 2009
10. Gonzales/ Woods/ Eddins, Digital Image Processing using MATLAB, 2nd edition, Gatesmark Publishing, ISBN 9780982085400
11. N.Efford, Digital Image Processing, Addison Wesley 2000, ISBN 0-201-59623-7
12. M Sonka, V Hlavac and R Boyle, Image Processing, Analysis and Machine Vision, PWS
13. 1999, ISBN 0-534-95393-
14. Pratt.W.K., Digital Image Processing, John Wiley and Sons, New York, 1978

**MASTER OF COMPUTER APPLICATIONS****II Year II Semester****INTERNET OF THINGS****(OPEN ELECTIVE-I)****MC4016****Objectives:**

- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- To introduce the hardware and working principles of various sensors used for IoT
- To introduce the Python Scripting Language which is used in many IoT devices
- To introduce the Raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of web based services on IoT devices

**UNIT I**

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs, IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates, Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle

**UNIT II**

Introduction to Python - Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling Python packages - JSON, XML, HTTPLib, URLLib, SMTPLib

**UNIT III**

IoT Physical Devices and Endpoints - Introduction to Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, reading input from pins.

**Unit IV**

Controlling Hardware- Connecting LED, Buzzer, Switching High Power devices with transistors, Controlling AC Power devices with Relays, Controlling servo motor, speed control of DC Motor, Using unipolar and bipolar Stepper motors

Digital input- Sensing push switch, pull-up and pull-down resistors, Rotary encoder, Using keypad, Using RTC

Sensors: Light sensor, temperature sensor with thermistor, voltage sensor, ADC and ADC, Temperature and Humidity Sensor DHT11, Read Switch, Distance Measurement with ultrasound sensor

**UNIT V**

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs Webservice – Web server for IoT, Cloud for IoT, Python web application framework Designing a RESTful web API

**TEXT BOOK:**

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759
3. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895

**MASTER OF COMPUTER APPLICATIONS****II Year II Semester****MC4P11****MOBILE COMPUTING****(PROFESSIONAL ELECTIVE-I)****UNIT I**

**Mobile Computing:** Novel Applications, Limitations of Mobile Computing, **Mobile Computing Architecture:** Programming languages, Functions of Operating Systems, Functions of Middleware for mobile Systems, Mobile Computing Architectural layers, Protocols, Layers.

**UNIT II**

**Mobile Devices:** Handheld Mobile Smartphones with Multimedia Functionalities, Smartcards, Smart Sensors, **Mobile System Networks:** Cellular Network, WLAN Network and Mobile IP, Ad-hoc Networks, **Mobility Management**

**UNIT III**

**Global System For Mobile Communications (Gsm):** Mobile Services, System Architecture, Protocols, Localization & Calling, Handover, Security. **GPRS:** GPRS System Architecture, **UMTS:** UMTS System Architecture. **LTE:** Long Term Evolution

**UNIT IV**

**Mobile Network Layer:** Mobile IP: Goals, Assumptions, Entities and Terminology, IP Packet Delivery, Agent Discovery, Registration, Tunneling and Encapsulation, Optimizations, Dynamic Host Configuration Protocol (DHCP) **Mobile Transport Layer:** Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/fast recovery, Transmission /time-out freezing, Selective retransmission, Transaction oriented TCP, TCP over 2.5G/3G Wireless Networks.

**UNIT V**

**Security Issues in Mobile Computing:** Introduction, Information Security, Security Techniques and Algorithms, Security Protocols, Security Models, Security Frameworks for mobile Environment

**TEXT BOOKS:**

1. Raj Kamal, "Mobile Computing", OXFORD UNIVERSITY PRESS.
2. Asoke K Talukder, et al, "Mobile Computing", Tata McGraw Hill, 2008.

**REFERENCES:**

1. Jochen Schiller, "Mobile Communications", Pearson Education, Second Edition, 2008.
2. Dr. Sunilkumar, et al "Wireless and Mobile Networks: Concepts and Protocols", Wiley India.
3. Matthew S.Gast, "802.11 Wireless Networks", SPD O'REILLY.
4. Ivan Stojmenovic , "Handbook of Wireless Networks and Mobile Computing", Wiley, 2007.
5. Kumkum Garg, "Mobile Computing", Pearson.
6. Handbook of Security of Networks, Yang Xiao, Frank H Li, Hui Chen, World Scientific, 2011.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**INFORMATION RETRIEVAL SYSTEMS**

**MC4P12**

**(PROFESSIONAL ELECTIVE-I)**

**UNIT I**

Introduction to Information Retrieval Systems : Definition of Information Retrieval System, Objectives of Information Retrieval System, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses; Boolean retrieval. The term vocabulary and postings lists. Dictionaries and tolerant retrieval. Index construction. Index compression.

**UNIT II**

Scoring, term weighting and the vector space model. Computing scores in a complete search system. Evaluation in information retrieval. Relevance feedback and query expansion.

**UNIT III**

XML retrieval. Probabilistic information retrieval. Language models for information retrieval. Text classification. Vector space classification.

**UNIT IV**

Support vector machines and machine learning on documents. Flat clustering. Hierarchical clustering. Matrix decompositions and latent semantic indexing.

**UNIT V**

Web search basics. Web crawling and indexes. Link analysis.

**TEXT BOOKS:**

1. Introduction to Information Retrieval , Christopher D. Manning and Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2008.
2. Information Storage and Retrieval Systems: Theory and Implementation, Kowalski, Gerald, Mark T Maybury, Springer.

**REFERENCE BOOKS :**

1. Modern Information Retrieval , Ricardo Baeza-Yates, Pearson Education, 2007.
2. Information Retrieval: Algorithms and Heuristics, David A Grossman and Ophir Frieder, 2nd Edition, Springer, 2004.
3. Information Retrieval Data Structures and Algorithms, William B Frakes, Ricardo Baeza-Yates, Pearson Education, 1992.
4. Information Storage & Retrieval , Robert Korfhage , John Wiley & Sons.



**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**MANAGEMENT INFORMATION SYSTEMS**

MC4P13

**(PROFESSIONAL ELECTIVE-I)**

**UNIT I**

The meaning and role of MIS : What is MIS?, Decision support systems, systems approach, the systems view of business, MIS Organizational within the Company.

Management, Organizational Theory, and the Systems Approach: Development of Organizational Theory, Management and Organizational Behaviour, Management, Information and the Systems Approach

**UNIT II**

Conceptual system design: Define the problems, set system objectives, establish system constraints, determine information needs, determine information sources, develop alternative conceptual designs and select one, document the system concept, prepare the conceptual design report. Organising data and information: Datawarehouses, Datamart and datamining

**UNIT III**

Detailed system design: Inform and involve the organization, aim of detailed design, project management of MIS detailed design, identify dominant and trade off criteria, define the subsystems, sketch the detailed operating subsystems and information flows, determine the degree of automation of each operation, inform and involve the organization again, inputs, outputs, and processing, early system testing, software, hardware and tools, propose an organization to operate the system, document the detailed design, revisit the manager-user.

**UNIT IV**

Implementation, evaluation and maintenance of the MIS: Plan the implementation, acquire floor space and plan space layouts, organize for implementation, develop procedures for implementation, train the operating personnel, computer related acquisitions, develop forms for data collection and information dissemination, develop the files, test the system, cutover, document the system, evaluate the MIS, control and maintain the system.

**UNIT V**

Pitfalls in MIS development: Fundamental weaknesses, soft spots, in planning, design problems, implementation : the TAR PIT.

**TEXT BOOKS:**

1. Information systems for modern management, 3rd Edition by R.G Murdick, J.E Ross and J. R clagget, PHI-1994.
2. Management Information Systems, Managing the Digital Firm Edition by Kenneth C. Laudon, Jane P. Laudon, Pearson Education, 10th Edition.

**REFERENCE BOOKS:**

1. Management information Systems, 4th edition by Robert Schultheis, Mary Sumner, PHI-Seventeenth Reprint 2007.
- 2 Principles of Information systems, Sixth edition by Ralph M.Stair, George W.Reynolds, Cengage learning.
- 3 Management Information Systems, J.A.O'brien, G.M.Marakas, R.Behl, 9<sup>th</sup> Edition, TMH.
- 4 Management Information Systems, Effy Oz, Cengage Learning.

**MASTER OF COMPUTER APPLICATIONS  
II YEAR II SEMESTER**

**MC4P14**

**PROGRAMMING LANGUAGES  
(PROFESSIONAL ELECTIVE-I)**

**UNIT I**

Preliminary Concepts: Reasons for studying, concepts of programming languages, Programming domains, Language Evaluation Criteria, influences on Language design, Language categories, Programming Paradigms – Imperative, Object Oriented, functional Programming , Logic Programming. Programming Language Implementation – Compilation and Virtual Machines, programming environments.

**UNIT II**

Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization.

**UNIT III**

Expressions and Statements: Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.

Subprograms and Blocks: Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines.

**UNIT IV**

Abstract Data types: Abstractions and encapsulation, introductions to data abstraction, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95

Concurrency: Subprogram level concurrency, semaphores, monitors, message passing, Java threads, C# threads.

Exception handling : Exceptions, exception Propagation, Exception handler in Ada, C++ and Java.

**UNIT V**

Logic Programming Language : Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

Functional Programming Languages: Introduction, fundamentals of FPL, LISP, ML, Haskell, application of Functional Programming Languages and comparison of functional and imperative Languages.

Scripting Language: Pragmatics, Key Concepts, Case Study : Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library.

TEXT BOOKS:

1. Concepts of Programming Languages Robert .W. Sebesta 8/e, Pearson Education,2008.
2. Programming Languages, K. C.Louden, 2nd Edition, Cengage Learning,2003.
3. Programming Language Design Concepts, D. A. Watt, Wiley dreamtech,rp-2007.

REFERENCE BOOKS:

1. Programming Languages, 2nd Edition, A.B. Tucker, R.E. Noonan, TMH.
2. LISP, Patric Henry Winston and Paul Horn, Pearson Education.
3. Programming in Prolog, W.F. Clocksin,& C.S.Mellish, 5th Edition, Springer.
4. Programming Python, M.Lutz, 3rd Edition, O'reilly,SPD, rp-2007.
5. Core Python Programming, Chun, II Edition, Pearson Education, 2007.
6. Programming Language Pragmatics,Scott,3<sup>rd</sup> edition,ELSEVIER.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**COMPUTER GRAPHICS**  
**(PROFESSIONAL ELECTIVE-I)**

**MC4P15**

**UNIT I**

Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms.  
 Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

**UNIT II**

2-D Geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D Viewing : The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

**UNIT III**

3-D Object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-spline curves, Bezier and B-spline surfaces. Basic illumination models, polygon rendering methods.

**UNIT IV**

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

**UNIT V**

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods

Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications

**TEXT BOOKS:**

1. “Computer Graphics C version”, Donald Hearn and M. Pauline Baker, Pearson education.
2. “Computer Graphics Principles & practice”, second edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.

**REFERENCE BOOKS:**

1. “Computer Graphics Second edition”, Zhiqiang Xiang, Roy Plastock, Schaum’s outlines, Tata Mc Graw hill edition.
2. “Procedural elements for Computer Graphics”, David F Rogers, Tata Mc Graw hill, 2nd edition.
3. “Principles of Interactive Computer Graphics”, Neuman and Sproul, TMH.
4. “Principles of Computer Graphics”, Shalini, Govil-Pai, Springer.
5. “Computer Graphics”, Steven Harrington, TMH.
6. Computer Graphics, F.S.Hill, S.M.Kelley, PHI.
7. Computer Graphics, P.Shirley, Steve Marschner & Others, Cengage Learning.
8. Computer Graphics & Animation, M.C.Trivedi, Jaico Publishing House.
9. An Integrated Introduction to Computer Graphics and Geometric Modelling, R.Goldman, CRC Press, Taylor & Francis Group.
10. Computer Graphics, Rajesh K. Maurya, Wiley India.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**MC4P16 ARTIFICIAL INTELLIGENCE**  
**(PROFESSIONAL ELECTIVE-I)**

**Objectives:**

- To learn the difference between optimal reasoning Vs human like reasoning
- To understand the notions of state space representation, exhaustive search, heuristic search along with the time and space complexities
- To learn different knowledge representation techniques
- To understand the applications of AI: namely Game Playing, Theorem Proving, Expert Systems, Machine Learning and Natural Language Processing

**UNIT-I**

Introduction: What is AI? Foundations of AI, History of AI, Agents and environments, The nature of the Environment, Problem solving Agents, Problem Formulation, Search Strategies

**UNIT-II**

Knowledge and Reasoning: Knowledge-based Agents, Representation, Reasoning and Logic, Propositional logic, First-order logic, Using First-order logic, Inference in First-order logic, forward and Backward Chaining

**UNIT-III**

Learning: Learning from observations, Forms of Learning, Inductive Learning, Learning decision trees, why learning works, Learning in Neural and Belief networks

**UNIT-IV**

Practical Natural Language Processing: Practical applications, Efficient parsing, Scaling up the lexicon, Scaling up the Grammar, Ambiguity, Perception, Image formation, Image processing operations for Early vision, Speech recognition and Speech Synthesis

**UNIT-V**

Robotics: Introduction, Tasks, parts, effectors, Sensors, Architectures, Configuration spaces, Navigation and motion planning, Introduction to AI based programming Tools

**TEXT BOOKS**

1. Stuart Russell, Peter Norvig: “Artificial Intelligence: A Modern Approach”, 2nd Edition, Pearson Education, 2007

**REFERENCES**

1. Artificial Neural Networks B. Yagna Narayana, PHI
2. Artificial Intelligence , 2nd Edition, E.Rich and K.Knight (TMH).
3. Artificial Intelligence and Expert Systems – Patterson PHI.
4. Expert Systems: Principles and Programming- Fourth Edn, Giarrantana/ Riley, Thomson.
5. PROLOG Programming for Artificial Intelligence. Ivan Bratka- Third Edition – Pearson Education.
6. Neural Networks Simon Haykin PHI

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**MC4L10 INTERNET TECHNOLOGIES LAB**

**List of Sample Problems**

1. Develop static pages (using Only HTML) of an online Book store. The pages should resemble: [www.amazon.com](http://www.amazon.com) The website should consist the following pages.

Home page, Registration and user Login

User Profile Page, Books catalog

Shopping Cart, Payment By credit card

Order Conformation

2. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
3. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
4. Bean Assignments

- a. Create a JavaBean which gives the exchange value of INR(Indian Rupees) into equivalent American/Canadian/Australian Dollar value.
- b. Create a simple Bean with a label - which is the count of number of clicks. Than create a BeanInfo class such that only the “count” property is visible in the Property Window.
- c. Create two Beans-a)KeyPad .b)DisplayPad .After that integrate the two Beans to make it work as a Calculator.
- d. Create two Beans Traffic Light(Implemented as a Label with only three background colours-Red,Green,Yellow) and Automobile(Implemented as a TextBox which states its state/movement). The state of the Automobile should depend on the following Light Transition Table.

Light Transition	Automobile State
Red ---> Yellow	Ready
Yellow ---> Green	Move
Green --> Red	Stopped

5. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
6. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue



should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

7. Implement the “Hello World!” program using JSP Struts Framework.

8.Redo the problem 5 using PHP.

<b>Additional Assignment Problems for the IT Lab.:</b>
Write an HTML page including any required Javascript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show “out of range” and if it is not a number, it should show “not a number” message in the result box.
Write a java swing application that takes a text file name as input and counts the characters, words and lines in the file. Words are separated with white space characters and lines are separated with new line character.
Write a simple calculator servlet that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operation performed on the operands. It should check in a database if the same expression is already computed and if so, just return the value from database. Use MySQL or PostgreSQL.(Do the same problem using PHP)
Write an HTML page that contains a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
Write a servlet that takes name and age from an HTML page. If the age is less than 18, it should send a page with “Hello <name>, you are not authorized to visit this site” message, where <name> should be replaced with the entered name. Otherwise it should send “Welcome <name> to this site” message. (Do the same problem using PHP)
Write a calculator program in HTML that performs basic arithmetic operations (+, -, /, * and %). Use CSS to change the foreground and background color of the values, buttons and result display area separately. Validate the input strings using JavaScript regular expressions. Handle any special cases like division with zero reasonably. The screen may look similar to the following:

<b>Value 1</b>	<b>Operator</b>	<b>Value 2</b>	<b>=</b>	<b>Result</b>
<input type="text"/>	<input type="text" value="+"/> ▼	<input type="text"/>	<input "="" type="text" value="="/>	<input type="text"/>

Write a Java program that creates a calculator GUI, as shown in figure. Extra components may be added for convenience:

**Color Scheme**  ▼

^  ▼  ^

v

**Result**

The Color Scheme may be Black on White or Blue on Yellow (selectable) and accordingly all components colors must be changed. The values can be either entered or increased or decreased by a step of 10. The operators are +, -, / and \* (selectable). Once any change

takes place, the result must be automatically computed by the program.

Write a Java Application that will read an XML file that contains personal information (Name, Mobile Number, age and place). It reads the information using SAX parser. After reading the information, it shows two input Text Fields in a window, one for tag name and the other for value. Once these two values are given, it should list all the records in the XML file that match the value of the given field in a text area (result box). For example, if the two text boxes are entered with “name” and “ABCD” then it should show all the records for which name is “ABCD”? An Illustration is given below that takes a mobile number and lists all the records that have the same mobile number.

**Field**

**Value**

**Result**

Consider the following web application for implementation:

The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.

If name and password matches, serves a welcome page with user's full name.

If name matches and password doesn't match, then serves "password mismatch" page

If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application using:

1. Pure JSP
2. Pure Servlets
3. Struts Framework
4. PHP

Implement a simple arithmetic calculator with +, -, /, \*, % and = operations using Struts Framework The number of times the calculator is used should be displayed at the bottom (use session variable).

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**DATA MINING LAB**

**MC4L20**

Task 1: Credit Risk Assessment

**Description:**

The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the bank's profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
3. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

**The German Credit Data:**

Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. credit dataset (original) Excel spreadsheet version of the German credit data.

In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer !)

A few notes on the German dataset

- DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
- owns\_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
- foreign\_worker. There are millions of these in Germany (many from Turkey). It is very hard to get German citizenship if you were not born of German parents.
- There are 20 attributes used in judging a loan applicant. The goal is to classify the applicant into one of two categories, good or bad.

**Subtasks : (Turn in your answers to the following tasks)**

1. List all the categorical (or nominal) attributes and the real-valued attributes separately. (5 marks)

2. What attributes do you think might be crucial in making the credit assessment ? Come up with some simple rules in plain English using your selected attributes. (5 marks)
3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training. (10 marks)
4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly ? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy ? (10 marks)
5. Is testing on the training set as you did above a good idea ? Why or Why not ? (10 marks)
6. One approach for solving the problem encountered in the previous question is using cross-validation ? Describe what is cross-validation briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease ? Why ? (10 marks)
7. Check to see if the data shows a bias against "foreign workers" (attribute 20), or "personal-status" (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka's GUI Explorer. Did removing these attributes have any significant effect? Discuss. (10 marks)
8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.) (10 marks)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)? (10 marks)
10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees ? How does the complexity of a Decision Tree relate to the bias of the model ? (10 marks)
11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain ? Also, report your accuracy using the pruned model. Does your accuracy increase ? (10 marks)

12.(Extra Credit): How can you convert a Decision Trees into "if-then-else rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules.PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one ! Can you predict what attribute that might be in this dataset ? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and oneR. (10 marks)

### Task Resources:

- Mentor lecture on Decision Trees
- Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
- Decision Trees (Source: Tan, MSU)
- Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)
- Weka resources:
  - Introduction to Weka (html version) (download ppt version)
  - Download Weka
  - Weka Tutorial
  - ARFF format
  - Using Weka from command line

### Task 2: Hospital Management System

Data Warehouse consists Dimension Table and Fact Table.

REMEMBER The following

Dimension

The dimension object (Dimension):

\_ Name

\_ Attributes (Levels) , with one primary key

\_ Hierarchies

One time dimension is must.

### About Levels and Hierarchies

Dimension objects (dimension) consist of a set of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation. Hierarchies describe parent-child relationships among a set of levels.

For example, a typical calendar dimension could contain five levels. Two hierarchies can be defined on these levels:

H1: YearL > QuarterL > MonthL > WeekL > DayL

H2: YearL > WeekL > DayL

The hierarchies are described from parent to child, so that Year is the parent of Quarter, Quarter the parent of Month, and so forth.

#### About Unique Key Constraints

When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for each level of the hierarchy and a unique key constraint on the lowest level (Base Level)

Design a Hospital Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier, Time. Where measures are 'NO UNITS', UNIT PRICE.

Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

PATIENT (patient\_name, Age, Address, etc.,)

MEDICINE ( Medicine\_Brand\_name, Drug\_name, Supplier, no\_units, Unit\_Price, etc.,)

SUPPLIER :( Supplier\_name, Medicine\_Brand\_name, Address, etc., )

If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably.

Design the Hospital Management system data warehouse using all schemas. Give the example 4-D cube with assumption names.

**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**  
**SOFTWARE TESTING LAB**

**MC4L30**

**Objectives:**

1.To learn to use the following(or Similar) automated testing tools to automate testing:

- a) Win Runner/QTP for functional testing.
- b) LoadRunner for Load/Stress testing.
- c) Test Director for test management.
- d) JUnit,HTMLUnit,CPPUnit.

**Sample problems on testing:**

1. Write programs in 'C' Language to demonstrate the working of the following constructs:  
i) do...while ii) while....do iii) if...else iv) switch v) for
2. "A program written in 'C' language for Matrix Multiplication fails" Introspect the causes for its failure and write down the possible reasons for its failure.
3. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
4. Write the test cases for any known application (e.g. Banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Study of any testing tool (e.g. Win runner)
7. Study of any web testing tool (e.g. Selenium)
8. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
9. Study of any test management tool (e.g. Test Director)
10. Study of any open source-testing tool (e.g. Test Link)
11. Take a mini project (e.g. University admission, Placement Portal) and execute it. During the Life cycle of the mini project create the various testing documents\* and final test report document.

**Additional problems on testing:**

- 1.Test the following using JUnit and CPPUnit:  
i)Sorting problems ii)Searching problems iii)Finding gcd of two integers iv)Finding factorial of a number.
- 2.Test web based forms using HTMLUnit.
- 3.Test database stored procedures using SQLUnit.  
(Use sufficient number of test cases in solving above Problems)

\*Note: To create the various testing related documents refer to the text "Effective Software Testing Methodologies by William E. Perry"

**REFERENCE BOOKS:**

- 1.Software Testing Concepts and Tools,P.Nageswara Rao,dreamtech press.
- 2.Software Testing Tools,Dr.K.V.K.K.Prasad,dreamtech Press.
- 3.Software Testing with Visual Studio Team System 2008,S.Subashini,N.Satheesh kumar,SPD.



**MASTER OF COMPUTER APPLICATIONS**  
**II Year II Semester**

**MC4A10**

**SOFT SKILLS**

**(Activity-based)**

**Course Objectives**

- ✎ To improve the fluency of students in English
- ✎ To facilitate learning through interaction
- ✎ To illustrate the role of skills in real-life situations with case studies, role plays etc.
- ✎ To train students in group dynamics, body language and various other activities which boost their confidence levels and help in their overall personality development
- ✎ To encourage students develop behavioral skills and personal management skills
- ✎ To impart training for empowerment, thereby preparing students to become successful professionals

**Learning Outcomes**

- ☞ Developed critical acumen and creative ability besides making them industry- ready.
- ☞ Appropriate use of English language while clearly articulating ideas.
- ☞ Developing insights into Language and enrich the professional competence of the students.
- ☞ Enable students to meet challenges in job and career advancement.

**Unit 1 : INTRODUCTION**

Definition and Introduction to Soft Skills – Hard Skills vs Soft Skills – Significance of Soft/Life/Self Skills – Self and SWOT Analysis *and*

**1. Exercises on Productivity Development**

- Effective/ Assertive Communication Skills (Activity based)
- Time Management (Case Study)
- Creativity & Critical Thinking (Case Study)
- Decision Making and Problem Solving (Case Study)
- Stress Management (Case Study)

**2. Exercises on Personality Development Skills**

- Self-esteem (Case Study)
- Positive Thinking (Case Study)

- Emotional Intelligence (Case Study)
- Team building and Leadership Skills (Case Study)
- Conflict Management (Case Study)

### 3. Exercises on Presentation Skills

- Netiquette
- Importance of Oral Presentation – Defining Purpose- Analyzing the audience- Planning Outline and Preparing the Presentation- Individual & Group Presentation- Graphical Organizers- Tools and Multi-media Visuals
- One Minute Presentations (Warming up)
- PPT on Project Work- Understanding the Nuances of Delivery- Body Language – Closing and Handling Questions – Rubrics for Individual Evaluation (Practice Sessions)

### 4. Exercises on Professional Etiquette and Communication

- Role-Play and Simulation- Introducing oneself and others, Greetings, Apologies, Requests, Agreement & Disagreement....etc.
- Telephone Etiquette
- Active Listening
- Group Discussions (Case study)- Group Discussion as a part of Selection Procedure- Checklist of GDs
- Analysis of Selected Interviews (Objectives of Interview)
- Mock-Interviews (Practice Sessions)
- Job Application and Preparing Resume
- Process Writing (Technical Vocabulary) – Writing a Project Report- Assignments

### 5. Exercises on Ethics and Values

Introduction — Types of Values - Personal, Social and Cultural Values - Importance of Values in Various Contexts

- Significance of Modern and Professional Etiquette – Etiquette (Formal and Informal Situations with Examples)
- Attitude, Good Manners and Work Culture (Live Examples)
- Social Skills - Dealing with the Challenged (Live Examples)
- Professional Responsibility – Adaptability (Live Examples)
- Corporate Expectations

☞ Note: Hand-outs are to be prepared and given to students.

☞ Training plan will be integrated in the syllabus.

☞ Topics mentioned in the syllabus are activity-based.

### SUGGESTED SOFTWARE:

☞ The following software from ‘[train2success.com](http://train2success.com)’

- Preparing for being Interviewed
- Positive Thinking
- Interviewing Skills
- Telephone Skills
- Time Management
- Team Building
- Decision making

**SUGGESTED READING**

1. Alex, K. 2012. *Soft Skills*. S. Chand Publishers
2. *Management Shapers*. 2011. Collection of 28 Books by different Authors. Universities Press.
3. Sherfield, Robert M. 2005. *et al Cornerstone: Developing Soft Skills*. Pearson
4. Suresh Kumar,E; Sreehari, P. & Savithri, J. 2011. *Communication Skills and Soft Skills- An Integrated Approach*. New Delhi: Pearson
5. The ACE of Soft Skills by Gopalaswamy Ramesh & Mahadevan Ramesh. 2013. Pearson Publishers. New Delhi.
6. Patnaik, P. 2011. *Group Discussion and Interview Skills*. New Delhi: Foundation
7. Sudhir Andrews. 2009. *How to Succeed at Interviews*. New Delhi: Tata McGraw Hill
8. Sasikumar, V & Dhamija, P.V. 1993. *Spoken English - A Self-Learning Guide to Conversation Practice*. New Delhi: Tata McGraw-Hill
9. *Dixson, Richard J. Everyday Dialogues in English*. Prentice Hall India Pvt Ltd
10. Mukhopadhyay. L *et al*. 2012. *Polyskills*. New Delhi: CUP India Pvt Ltd
11. Rizvi, M. A. 2005. *Effective Technical Communication*. New Delhi: Tata McGraw Hill
12. *The Hindu Speaks on Education* by the Hindu Newspaper
13. Naterop, B. Jean and Revell, Rod. 2004. *Telephoning in English*. Cambridge: CUP

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****MC5C10 CORE-IV ANDROID APPLICATION DEVELOPMENT****Objectives:**

- To demonstrate their understanding of the fundamentals of Android operating systems
- To demonstrate their skills of using Android software development tools
- To demonstrate their ability to develop software with reasonable complexity on mobile platform
- To demonstrate their ability to deploy software to mobile devices
- To demonstrate their ability to debug programs running on mobile devices

**UNIT I**

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Eclipse platform, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools

Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes

Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

**UNIT II**

Android User Interface: Measurements – Device and pixel density independent measuring units Layouts – Linear, Relative, Grid and Table Layouts

User Interface (UI) Components – Editable and non editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers

Event Handling – Handling clicks or changes of various UI components

Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

**UNIT III**

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS

Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity

Notifications – Creating and Displaying notifications, Displaying Toasts

**UNIT IV**

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

**UNIT V**

**Advanced Topics:** Alarms – Creating and using alarms

Using Internet Resources – Connecting to internet resource, using download manager

Location Based Services – Finding Current Location and showing location on the Map, updating location

**Text Books:**

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013

**References:**

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****MC5C20****CORE-V****DATA SCIENCE****Objectives:**

To understand about big data, to learn the analytics of Big Data

To understand how data is stored and processed in Hadoop

To learn about NoSQL databases

To learn R tool and understand how data is analyzed using R features

To learn about spark and to understand what features of it are making it to overtake hadoop

**UNIT I**

Types of Digital data: Classification of Digital Data, Introduction to Big Data: What is big data, Evolution of Big Data, Traditional Business Intelligence vs Big Data, Coexistence of Big Data and Data Warehouse. Big Data Analytics: What is Big Data Analytics, What Big Data Analytics Isn't, Why this sudden Hype Around Big Data Analytics, Classification of Analytics, Greatest Challenges that Prevent Business from Capitalizing Big Data, Top Challenges Facing Big Data, Why Big Data Analytics Important, Data Science, Terminologies used in Big Data Environments.

**UNIT II**

Hadoop: Features of Hadoop, Key advantages of hadoop, versions of hadoop, overview of hadoop ecosystem, Hadoop distributions.

Why hadoop? RDBMS vs Hadoop, Distribution computing challenges, History of hadoop, Hadoop overview, HDFS

**UNIT III**

Processing data with hadoop, interfacing with hadoop ecosystem.

NoSQL: Where it is used? What is it? Types of NoSQL Databases, Why NoSQL? Advantages of NoSQL, What we miss with NoSQL? Use of NoSQL in industry, SQL vs NoSQL.

**UNIT IV**

What is R? Why use R for analytics? How to run R? First R example, functions a short programming example, some important R data structures, vectors, matrices, lists, R programming structures.

**UNIT V**

Introduction to Spark, Scala language: values, data types, variables, expressions, conditional expressions, evaluation order, compound expressions, functions, tuple with functions, List, Length, ++, ::, sorted, reverse, sum, slice, mkString, contains, map, filter, leftfold, reduce, Map, Contains, getOrElse, WithDefault, Keys and Values, groupBy, set, mapValues, keys and values, Option(Some and None), Objects, classes, inheritance, traits

**TEXT BOOKS:**

1. BIG DATA and ANALYTICS, Seema Acharya, Subhashini Chellappan, Wiley publications (Unit I, II, III)
2. BIG DATA, Black Book™, DreamTech Press, 2015 Edition.
3. "The art of R programming" by Norman matloff, 2009.(Unit IV)
4. "Atomic Scala", 2<sup>nd</sup> edition, Bruce Eckel, Dianne Marsh. (Unit V)

**REFERENCE BOOKS:**

1. Rajiv Sabherwal, Irma Becerra- Fernandez," Business Intelligence –Practice, Technologies and Management", John Wiley 2011.

2. Lariss T. Moss, ShakuAtre, “ Business Intelligence Roadmap”, Addison-Wesley It Service.
3. Yuli Vasiliev, “ Oracle Business Intelligence : The Condensed Guide to Analysis and Reporting”, SPD Shroff, 2012.
4. “Hadoop: The definitive guide”, by O’reilly, yahoo press, 2<sup>nd</sup> edition.
5. “Introduction to R” by Sandeep Rakshit, McGrawHill Education, 2016.

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****MC5C30 CORE-VI OBJECT ORIENTED ANALYSIS AND DESIGN****UNIT I**

Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

**UNIT II**

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams. Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

**UNIT III**

Basic Behavioral Modeling-I: Interactions, Interaction diagrams.

Basic Behavioral Modeling-II: Use cases, Use case Diagrams, Activity Diagrams.

**UNIT IV**

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

**UNIT V**

Patterns and Frameworks, Artifact Diagrams. Case Study: The Unified Library application

**TEXT BOOKS:**

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education 2nd Edition
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

**REFERENCE BOOKS:**

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY-Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TMH.
5. Applying UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.
6. Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning.
7. UML and C++, R.C.Lee, and W.M.Tepfenhart, PHI.
8. Object Oriented Analysis, Design and Implementation, B.Dathan, S.Ramnath, Universities Press.
9. OODesign with UML and Java, K.Barclay, J.Savage, Elsevier.
10. Learning UML 2.0, Russ Miles and Kim Hamilton, O'Reilly, SPD.



**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**SOFTWARE PROJECT MANAGEMENT**  
**(OPEN ELECTIVE-II)**

**MC5021**

**UNIT I**

Conventional Software Management : The waterfall model, conventional software Management performance. Evolution of Software Economics : Software Economics, pragmatic software cost estimation.

**UNIT II**

Improving Software Economics : Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections. The old way and the new : The principles of conventional software engineering, principles of modern software management, transitioning to an iterative process.

**UNIT III**

Life cycle phases : Engineering and production stages, inception, Elaboration, construction, transition phases.

Artifacts of the process : The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts. Model based software architectures : A Management perspective and technical perspective.

**UNIT IV**

Work Flows of the process : Software process workflows, Inter trans workflows. Checkpoints of the Process : Major Mile Stones, Minor Milestones, Periodic status assessments. Iterative Process Planning : Work breakdown structures, planning guidelines, cost and schedule estimating, Interaction planning process, Pragmatic planning.

Project Organizations and Responsibilities : Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation : Automation Building Blocks, The Project Environment.

**UNIT V**

Project Control and Process instrumentation : The server care Metrics, Management indicators, quality indicators, life cycle expectations pragmatic Software Metrics, Metrics automation. Tailoring the Process : Process dicriminants, Example.

Future Software Project Management : Modern Project Profiles Next generation

Software economics, modern Process transitions.

Case Study : The Command Center Processing and Display System-Replacement(CCPDS-R)

**TEXT BOOKS:**

1. Software Project Management, Walker Royce, Pearson Education, 1998
2. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, Tata Mc-Graw Hill, 2006

**REFERENCE BOOKS:**

1. Applied Software Project Management, Andrew Stellman & Jennifer Greene, O'Reilly, 2006
2. Head First PMP, Jennifer Greene & Andrew Stellman, O'Reilly, 2007
3. Software Engineering Project Managent, Richard H. Thayer & Edward Yourdon, second edition, Wiley India, 2004.
4. Agile Project Management, Jim Highsmith, Pearson education, 2004

**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**DIGITAL IMAGE PROCESSING**  
**(OPEN ELECTIVE – II)**

**MC5022**

**UNIT – I**

Fundamental steps of image processing, components of an image processing of system. The image model and image acquisition, sampling and quantization, relationship between pixels, distance functions, scanner. Statistical and spatial operations, Intensity functions transformations.

**UNIT – II**

Histogram processing, smoothing & sharpening – spatial filters Frequency domain filters, homomorphic filtering, image filtering & restoration. Inverse and weiner filtering, FIR weiner filter, Filtering using image transforms.

Morphological and other area operations, basic morphological operations, opening and closing operations, dilation erosion, Hit or Miss transform, morphological algorithms, extension to grey scale images.

**UNIT- III**

Segmentation and Edge detection region operations, basic edge detection, second order detection, crack edge detection, gradient operators, compass and laplace operators, edge linking and boundary detection, thresholding, regionbased segmentation, segmentation by morphological watersheds,Color models,Color Image Processing.

**UNIT -IV**

Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression, image data compression-predictive technique, pixel coding, transfer coding theory, lossy and lossless predictive type coding, Digital Image Water marking.

**UNIT –V**

Representation and Description

Chain codes, Ploygonal approximation, Signature Boundary Segments, Skeltons, Boundary Descriptors, Regional Descriptors, Relational Descriptors, Principal components for Description, Relational Descriptors

**TEXT BOOKS :**

- 1.Digital Image Processing Third edition, Pearson Education,Rafael C. Gonzalez, Richard
- 2.Digital Image Processing,S.Jayaraman,S.Esakkirajan,T.Veerakumar,TMH.

**REFERENCE BOOKS :**

1. Image Processing, Analysis and Machine Vision, Second Edition, Milan Sonka,  
 Vaclav Hlavac and Roger Boyle,Cengage learning.

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****MC5023****SOCIAL MEDIA INTELLIGENCE****(OPEN ELECTIVE-II)****UNIT – I**

The Beginnings of Social Media Intelligence: What is Social Media monitoring? Anecdotal referencing of Social Media Comments, Text Mining, Some Simple Metrics, Using Social Media as Early Warning System.

Fundamental of Opinion Formation: Affecting Opinion versus Biasing Expression, How Do We Form Opinions?, How Do Expectations Affect Opinion?, How Do Expertise and Knowledge Influence How We Form Opinions?, Opinion Formation in a Social Context, Bandwagon behavior and Information Cascades, Implications for Social Media Intelligence.

**UNIT – II**

Why Do We Share our Opinions : Poster versus Lurkers, What Motivates Us to Post/, Posting Motivations and Selection effects, Implications for Social Media Intelligence.

The Social effects of Strangers : How Does Social Context Affect Our Behavior?, How Influential is the Social Context/, How Does Social Context Affect Opinion Expression/, Bandwagon Behavior in Opinion expression, Differentiating Our opinions, Multiple Audience Effects, /can We Trust the Wisdom of Crowds.

**UNIT – III**

Opinion Ecosystems and the Evolution Within : Life Cycle Dynamics, Preference Mismatching and Sequential Dynamics, Social Dynamics, Are Social Media Communities the Cause of Opinion Radicalization ?, Online Echo Chambers, Implications for Social Media Monitoring and Metrics.

Are Social Media Fragmenting the Population ? : Self-Organization, Birds of a Feather Flock Together, Geography No Longer Defines Our Communities, The influential Hypothesis, The New Influential, How Can We Identify Influentials, Influence in e-Commerce, Some Concluding Remarks.

**UNIT – IV**

Managing Social Media Communities for Better Social Media Intelligence: Creating an Inviting Environment, The Benefits of a Well-Managed Opinion Community (and the Costs of Not Managing the Community at All) Quality of Intelligence Depends on the Quality of the Opinion Community, Creating and Manipulating Buzz, Buzz Campaign or Fraud?, Identifying Fraudulent Opinions

Cutting Through the Online Chatter : A New Paradigm for Marketing Research, Measure What Matters, Cast a Wide Net, Analyze the Text, Understand the biases, Establish Links to Performance metrics.

**UNIT – V**

Intelligence Integration : Overview of Marketing Research Methods, Using Social Media for Marketing research, Tracking Brand Health, Understanding Market Structure, Social Shopping, Integration with Data from Other Parts of the Organization, Intelligence Dashboards.

Building Social Media Intelligence into Our Strategies : How Can Social Media Intelligence Help Integrate an Organization's Strategy?, Multichannel Strategies, Rapid Response System, Integrated CRM, Leveraging Social Data, Seeding Strategies.

Moving from Social Media monitoring to Social Media Intelligence : Social Media Intelligence today, Social Media Intelligence tomorrow, Building on the Science of Opinion, tapping into Opinion Ecosystems, Developing an Integrated Strategy.

**References:**

1. SOCIAL MEDIA INTELLIGENCE : by Wendly W.Moe, David A. Schweidel, Cambridge University, edition 2014.

**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**MACHINE LEARNING**  
**(OPEN ELECTIVE -2)**

MC5024

**Objectives:**

- To be able to formulate machine learning problems corresponding to different applications.
- To understand a range of machine learning algorithms along with their strengths and weaknesses.
- To understand the basic theory underlying machine learning.

**UNIT – I**

**Introduction:** An illustrative learning task, and a few approaches to it. What is known from algorithms? Theory, Experiment. Biology. Psychology. Overview of Machine learning, related areas and applications. Linear Regression, Multiple Regression, Logistic Regression, logistic functions

**Concept Learning:** Version spaces. Inductive Bias. Active queries. Mistake bound/ PAC model. basic results. Overview of issues regarding data sources, success criteria.

**UNIT –II**

**Decision Tree Learning:** - Minimum Description Length Principle. Occam's razor. Learning with active queries Introduction to information theory, Decision Trees, Cross Validation and Over fitting

**Neural Network Learning:** Perceptions and gradient descent back propagation, multilayer networks and back propagation.

**UNIT –III**

**Sample Complexity and Over fitting:** Errors in estimating means. Cross Validation and jackknifing VC dimension. Irrelevant features: Multiplicative rules for weight tuning.

**Support Vector Machines:** functional and geometric margins, optimum margin classifier, constrained optimization, Lagrange multipliers, primal/dual problems, KKT conditions, dual of the optimum margin classifier, soft margins, and kernels.

**Bayesian Approaches:** The basics Expectation Maximization. Bayes theorem, Naïve Bayes Classifier, Markov models, Hidden Markov Models

**UNIT—IV**

**Instance-based Techniques:** Lazy vs. eager generalization. K nearest neighbor, case- based reasoning.

**Clustering and Unsupervised Learning:** K-means clustering, Gaussian mixture density estimation, model selection

**UNIT—V**

**Genetic Algorithms:** Different search methods for induction - **Explanation-based Learning:** using prior knowledge to reduce sample complexity.

**Dimensionality reduction:** feature selection, principal component analysis, linear discriminant analysis, factor analysis, independent component analysis, multidimensional scaling, manifold learning

**TEXT BOOKS:**

1. Tom Michel, Machine Learning, McGraw Hill, 1997
2. Trevor Hastie, Robert Tibshirani & Jerome Friedman. The Elements of Statistically Learning, Springer Verlag, 2001

**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**

**MC5O25**

**ADVANCED PROGRAMMING**  
**( OPEN ELECTIVE – 2)**

**Unit I**

Basic features of Python-Interactive execution,comments,types,variables,operators,expressions, Statements-assignment, input,print,Control flow-Conditionals,Loops,break statement,continue statement, pass statement,Functions,definition,call,scope and lifetime of variables,keyword arguments,default parameter values,variable length arguments,recursive functions,Functional programming-mapping,filtering and reduction,Lambda functions,Scope,namespaces and modules-import statement,creating own modules,avoiding namespace collisions when importing modules, module reload, LEGB rule, dir( ) function, iterators and generators,Sequences-Strings ,Lists and Tuples-basic operations and functions, iterating over sequences, List comprehensions, Packing and Unpacking of Sequences,Sets and Dictionaries- operations, regular expressions, Python program examples.

**Unit II**

Files-operations-opening, reading, writing, closing,file positions,file names and paths,functions for accessing and manipulating files and directories on disk, os module, Exceptions – raising and handling exceptions, try/except statements, finally clause, standard exceptions, Object oriented programming- classes, constructors, objects, class variables, class methods, static methods, Inheritance-is-a relationship, composition, polymorphism, overriding, multiple inheritance, abstract classes, multithreaded programming, time and calendar modules,Python program examples.

**Unit III**

GUI Programming with Tkinter , Widgets(Buttons, Canvas, Frame, Label, Menu, Entry, Text, Scrollbar, Combobox, Listbox, Scale),event driven programming-events, callbacks, binding, layout management-geometry managers:pack and grid, creating GUI based applications in Python.

**Unit IV**

Network Programming-Sockets, Socket addresses, Connection-oriented and Connectionless Sockets,socket module,urllib module,Socket object methods,Client/Server applications(TCP/IP and UDP/IP),Socketserver module, handling multiple clients, Client side scripting-Transferring files-FTP, ftplib module,ftplib.FTP class methods, sending and receiving emails- smtplib module, smtplib.SMTP class methods, poplib module, poplib.POP3 methods, Python program examples.

**Unit V**

Database Programming-SQL Databases,SQLite,sqlite3 module, connect function( ),DB-API 2.0 Connection object methods, Cursor object Attributes and methods, creating Database applications in Python, Web programming-Simple web client, urllib, urlparse modules, Server side scripting-Building CGI applications-Setting up a web server, Creating the form page, Generating the results page, Saving state information in CGI Scripts, HTTP Cookies, Creating a cookie, Using cookies in CGI scripts, Handling cookies with urllib2 module, cgi module.

**TEXT BOOKS :**

1. Exploring Python, Timothy A. Budd, McGraw Hill Publications.
2. Core Python Programming, 2<sup>nd</sup> edition, W.J.Chun, Pearson.
3. Python Programming, R.Thareja, Oxford University Press.
4. Programming Python, 3<sup>rd</sup> edition, Mark Lutz, SPD,O'Reilly.

**REFERENCE BOOKS :**

1. Introduction to Computer Science using Python, Charles Dierbach, Wiley India Edition.
2. Fundamentals of Python, K. A. Lambert, B.L. Juneja, Cengage Learning.
3. Beginning Python,2<sup>nd</sup> edition, Magnus Lie Hetland, Apress, dreamtech press.
4. Starting out with Python, 3<sup>rd</sup> edition, Tony Gaddis, Pearson.
5. Python Essential Reference, D.M.Beazley, 3<sup>rd</sup> edition, Pearson.
6. Programming in Python3, Mark Summerfield, Pearson.
7. Think Python, How to think like a computer scientist, Allen B. Downey,SPD, O'Reilly.
8. [www.python.org](http://www.python.org) web site.

**MASTER OF COMPUTER APPLICATIONS  
III YEAR I SEM**

**MC5026**

**STORAGE AREA NETWORKS  
(OPEN ELECTIVE – II)**

**Objectives:**

To understand Storage Area Networks characteristics and components.

To become familiar with the SAN vendors and their products

To learn Fibre Channel protocols and how SAN components use them to communicate with each other

To become familiar with Cisco MDS 9000 Multilayer Directors and Fabric Switches

Thoroughly learn Cisco SAN-OS features.

To understand the use of all SAN-OS commands. Practice variations of SANOS features

**UNIT I**

Introduction to Storage Technology Review data creation and the amount of data being created and understand the value of data to a business, challenges in data storage and data management, Solutions available for data storage, Core elements of a data center infrastructure, role of each element in supporting business activities

**UNIT II**

Storage Systems Architecture Hardware and software components of the host environment, Key protocols and concepts used by each component ,Physical and logical components of a connectivity environment ,Major physical components of a disk drive and their function, logical constructs of a physical disk, access characteristics, and performance Implications, Concept of RAID and its components , Different RAID levels and their suitability for different application environments: RAID 0, RAID 1, RAID 3, RAID 4, RAID 5, RAID 0+1, RAID 1+0, RAID 6, Compare and contrast integrated and modular storage systems ,High-level architecture and working of an intelligent storage system

**UNIT III**

Introduction to Networked Storage Evolution of networked storage, Architecture, components, and topologies of FC-SAN, NAS, and IP-SAN, Benefits of the different networked storage options, Understand the need for long-term archiving solutions and describe how CAS fulfills the need , Understand the appropriateness of the different networked storage options for different application environments

**UNIT IV**

Information Availability & Monitoring & Managing Datacenter List reasons for planned/unplanned outages and the impact of downtime, Impact of downtime, Differentiate between business continuity (BC) and disaster recovery (DR) ,RTO and RPO, Identify single points of failure in a storage infrastructure and list solutions to mitigate these failures, Architecture of backup/recovery and the different backup/recovery topologies , replication technologies and their role in ensuring information availability and business continuity, Remote replication technologies and their role in providing disaster recovery and business continuity capabilities Identify key areas to monitor in a data center, Industry standards for data center monitoring and management, Key metrics to monitor for different components in a storage infrastructure, Key management tasks in a data center.



**UNIT V**

Securing Storage and Storage Virtualization Information security, Critical security attributes for information systems, Storage security domains, List and analyzes the common threats in each domain, Virtualization technologies, block-level and file-level virtualization technologies and processes Case Studies

The technologies described in the course are reinforced with EMC examples of actual solutions. Realistic case studies enable the participant to design the most appropriate solution for given sets of criteria.

**TEXT BOOK:**

1. EMC Corporation, Information Storage and Management, Wiley.

**REFERENCE BOOKS:**

1. Robert Spalding, "Storage Networks: The Complete Reference", Tata McGraw Hill, Osborne, 2003.
2. Marc Farley, "Building Storage Networks", Tata McGraw Hill, Osborne, 2001.
3. Meeta Gupta, Storage Area Network Fundamentals, Pearson Education Limited, 2002.

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****CYBER SECURITY****(PROFESSIONAL ELECTIVE –II)****MC5P21****OBJECTIVES:**

To learn about cyber crimes and how they are planned

To learn the vulnerabilities of mobile and wireless devices

To learn about the crimes in mobile and wireless devices

**UNIT-I****Introduction to Cybercrime:**

Introduction, Cybercrime and Information security, who are cybercriminals, Classifications of Cybercrimes, Cybercrime: The legal Perspectives and Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes.

**Cyber offenses:** How criminals Plan Them

Introduction, How Criminals plan the Attacks, Social Engineering, Cyber stalking, Cyber cafe and Cybercrimes, Botnets: The Fuel for Cybercrime, Attack Vector, Cloud Computing.

**UNIT-II****Cybercrime:** Mobile and Wireless Devices Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implications for Organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops.**UNIT III****Cybercrimes and Cyber security:** the Legal Perspectives Introduction Cyber Crime and Legal Landscape around the world, Why Do We Need Cyber laws: The Indian Context, The Indian IT Act, Challenges to Indian Law and Cybercrime Scenario In India, Digital signatures and the Indian IT Act, Amendments to the Indian IT Act, Cybercrime and Punishment ,Cyber law, Technology and Students: Indian Scenario.**UNIT IV****Understanding Computer Forensics :**Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Chain of Custody concept, Network Forensics, Approaching a computer, Forensics Investigation, Challenges in Computer Forensics, Special Tools and Techniques,Forensics Auditing**UNIT V****Cyber Security:** Organizational Implications

Introduction, Cost of Cybercrimes and IPR issues, Web threats for Organizations, Security and Privacy Implications, Social media marketing: Security Risks and Perils for Organizations, Social Computing and the associated challenges for Organizations.

**Text book:**

1. **Cyber Security:** *Understanding Cyber Crimes, Computer Forensics and Legal Perspectives*, Nina Godbole and Sunil Belapure, Wiley INDIA.
2. Introduction to Cyber Security , Chwan-Hwa(john) Wu,J.David Irwin.CRC Press T&F Group

**Reference book:**

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****MC5P22****DISTRIBUTED DATABASES  
(PROFESSIONAL ELECTIVE-II)****UNIT I**

Features of Distributed versus Centralized Databases, Principles of Distributed Databases, Levels Of Distribution Transparency, Reference Architecture for Distributed Databases, Types of Data Fragmentation, Integrity Constraints in Distributed Databases, Distributed Database Design

**UNIT II**

Translation of Global Queries to Fragment Queries, Equivalence transformations for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parametric Queries.

Optimization of Access Strategies, A Framework for Query Optimization, Join Queries, General Queries

**UNIT III**

The Management of Distributed Transactions, A Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions

Concurrency Control, Foundation of Distributed Concurrency Control, Distributed Deadlocks, Concurrency Control based on Timestamps, Optimistic Methods for Distributed Concurrency Control.

**UNIT IV**

Reliability, Basic Concepts, Nonblocking Commitment Protocols, Reliability and concurrency Control, Determining a Consistent View of the Network, Detection and Resolution of Inconsistency, Checkpoints and Cold Restart, Distributed Database Administration, Catalog Management in Distributed Databases, Authorization and Protection

**UNIT V**

Architectural Issues, Alternative Client/Server Architectures, Cache Consistency, Object Management, Object Identifier Management, Pointer Swizzling, Object Migration, Distributed Object Storage, Object Query Processing, Object Query Processor Architectures, Query Processing Issues, Query Execution, Transaction Management, Transaction Management in Object DBMSs, Transactions as Objects

Database Integration, Scheme Translation, Scheme Integration, Query Processing Query Processing Layers in Distributed Multi-DBMSs, Query Optimization Issues Transaction Management Transaction and Computation Model, Multidatabase Concurrency Control, Multidatabase Recovery, Object Orientation and Interoperability, Object Management Architecture CORBA and Database interoperability, Distributed Component Object Model, COM/OLE and Database Interoperability, PUSH-Based Technologies

**TEXT BOOKS:**

1. Distributed Databases Principles & Systems, Stefano Ceri, Giuseppe Pelagatti, TMH.
2. Principles of Distributed Database Systems, M. Tamer Ozsu, Patrick Valduriez , Pearson Education, 2nd Edition.

**REFERENCE BOOKS:**

1. Distributed Database Systems, Chanda Ray, Pearson.
2. Distributed Database Management Systems, S.K. Rahimi and Frank.S. Haug, Wiley.

**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**DISTRIBUTED COMPUTING**  
**(PROFESSIONAL ELECTIVE-II)**

**MC5P23**

**UNIT I**

**Introduction:** The different forms of computing, The strengths and weaknesses of Distributed computing, Operating system concepts relevant to distributed computing, the architecture of distributed applications. Paradigms for Distributed Applications, choosing a Paradigm for an application (trade-offs).

**UNIT II**

**Cluster Computing:** Parallel computing overview, cluster computing – Introduction, Cluster Architecture, parallel programming models and Paradigms, Applications of Clusters.

**UNIT III**

**Grid Computing:** Introduction, Grid Computing Anatomy – Architecture, Architecture and relationship to other Distributed Technologies, Grid computing road map. Merging the Grid services Architecture with the Web Services Architecture.

**UNIT IV**

Open Grid Service Architecture – Introduction, Architecture and Goal, Sample Use cases: Commercial Data Center, National Fusion Collaboratory, Online Media and Entertainment. OGSA platform Components, Open Grid Services Infrastructure.

**UNIT V**

Globus GT 3 Toolkit – Architecture, Programming Model, A sample implementation, High Level services, OGSI.NET Middleware Solutions.

**TEXT BOOKS:**

1. Grid Computing, Joshy. Joseph & Craig Fellenstein, Pearson education, 2004
2. Distributed Computing, Principles and Applications, M.L.Liu, Pearson Education, 2004
3. High Performance Cluster Computing, Rajkumar Buyya, Pearson education.

**REFERENCE BOOKS:**

1. Grid Computing – Making the global infrastructure a reality, Fran Berman, Geoffrey C Fox, Anthony J G Hey, Wiley India, 2010
2. A Networking Approach to Grid Computing, D.Minoli, Wiley & sons, 2006
3. Grid Computing: A Practical Guide to Technology and Applications, A.Abbas, Firewall Media, 2008

**MASTER OF COMPUTER APPLICATIONS****III Year I Semester****MC5P24****CLOUD COMPUTING****(PROFESSIONAL ELECTIVE-II)****Course Description:**

Cloud computing has evolved as a very important computing model, which enables information, software, and shared resources to be provisioned over the network as services in an on-demand manner. This course provides an insight into what is cloud computing and the various services cloud is capable.

**UNIT I:**

**Computing Paradigms,** High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Biocomputing, Mobile Computing, Quantum Computing, Optical Computing, Nanocomputing.

**UNIT II:**

**Cloud Computing Fundamentals:** Motivation for Cloud Computing, The Need for Cloud Computing, Defining Cloud Computing, Definition of Cloud computing, Cloud Computing Is a Service, Cloud Computing Is a Platform, Principles of Cloud computing, Five Essential Characteristics... Four Cloud Deployment Models

**UNIT III:**

Cloud Computing Architecture and Management: Cloud architecture, Layer, Anatomy of the Cloud, Network Connectivity in Cloud Computing, Applications, on the Cloud, Managing the Cloud, Managing the Cloud Infrastructure Managing the Cloud application, Migrating Application to Cloud, Phases of Cloud Migration Approaches for Cloud Migration.

**UNIT IV:**

**Cloud Service Models:** Infrastructure as a Service, Characteristics of IaaS, Suitability of IaaS, Pros and Cons of IaaS, Summary of IaaS Providers, Platform as a Service, Characteristics of PaaS, Suitability of PaaS, Pros and Cons of PaaS, Summary of PaaS Providers, Software as a Service, Characteristics of SaaS, Suitability of SaaS, Pros and Cons of SaaS, Summary of SaaS Providers. Other Cloud Service Models

**UNIT V:**

**Cloud Service Providers:** EMC, EMC IT, Captiva Cloud Toolkit, Google, Cloud Platform, Cloud Storage, Google Cloud Connect, Google Cloud Print, Google App Engine, Amazon Web Services, Amazon Elastic Compute Cloud, Amazon Simple Storage Service, Amazon Simple Queue Service, Microsoft, Windows Azure, Microsoft Assessment and Planning Toolkit, SharePoint, IBM, Cloud Models, IBM Smart Cloud,

SAP Labs,SAP HANA Cloud Platform,Virtualization Services Provided by SAP,Salesforce,Sales Cloud, Service Cloud: Knowledge as a Service,Rackspace,VMware,Manjrasoft,Aneka Platform

**TEXT BOOKS:**

1. **Essentials of cloud Computing : K.Chandrasekhran , CRC press, 2014**

**REFERENCE BOOKS:**

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing , Kai Hwang, Geoffery C.Fox, Jack J.Dongarra, Elsevier, 2012.
3. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, rp2011.



**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**E-COMMERCE**  
**(PROFESSIONAL ELECTIVE-II)**

**Objectives:**

Identify the major categories and trends of e-commerce applications.  
 Identify the essential processes of an e-commerce system.  
 Identify several factors and web store requirements needed to succeed in e-commerce.  
 Discuss the benefits and trade-offs of various e-commerce clicks and bricks alternatives.  
 Understand the main technologies behind e-commerce systems and how these technologies interact.  
 Discuss the various marketing strategies for an online business.  
 Define various electronic payment types and associated security risks and the ways to protect against them.

**UNIT - I**

Electronic Commerce-Frame work, anatomy of E-Commerce applications, E-Commerce Consumer applications, E-Commerce organization applications. Consumer Oriented Electronic commerce - Mercantile Process models.

**UNIT - II**

Electronic payment systems - Digital Token-Based, Smart Cards, Credit Cards, Risks in Electronic Payment systems. Inter Organizational Commerce - EDI, EDI Implementation, Value added networks.

**UNIT - III**

Intra Organizational Commerce - work Flow, Automation Customization and internal Commerce, Supply chain Management. Corporate Digital Library - Document Library, digital Document types, corporate Data Warehouses.

**UNIT- IV**

Advertising and Marketing - Information based marketing, Advertising on Internet, on-line marketing process, market research. Consumer Search and Resource Discovery - Information search and Retrieval, Commerce Catalogues, Information Filtering.

**UNIT - V**

Multimedia - key multimedia concepts, Digital Video and electronic Commerce, Desktop video processing, Desktop video conferencing.

**TEXT BOOK:**

1. Frontiers of electronic commerce – Kalakata, Whinston, Pearson.

**REFERENCES BOOKS:**

1. E-Commerce fundamentals and applications Hendry Chan, Raymond Lee, Tharam Dillon, Ellizabeth Chang, John Wiley.
2. E-Commerce, S.Jaiswal – Galgotia.
3. E-Commerce, Efrain Turbon, Jae Lee, David King, H.Michael Chang.
4. Electronic Commerce – Gary P.Schneider – Thomson.
5. E-Commerce – Business, Technology, Society, Kenneth C.Taudon, Carol Guyerico Traver.

**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**SEMANTIC ANALYTICS**  
**(PROFESSIONAL ELECTIVE-II)**

**MC5P26**

**Objectives:**

- To learn how machines understand the natural language
- To learn the concepts of ontology and RDF programming
- To explore the semantic web and its applications
- To introduce text analytics techniques for semantic Analysis

**UNIT 1:**

**Introduction to Semantic Analysis-** Semantic analysis definition, Semantic Technology, Basic Concepts and Issues in Natural Language Semantic, Theories and Approaches to Semantic Representation, Logical Approaches, Discourse Representation Theory, Pustejovsky's Generative Lexicon, Natural Semantic Metalanguage, Object-Oriented Semantics, Relational Issues in Lexical Semantics.

**UNIT 2:**

**Principles of Systematic Lexicography** -Reconstruction of the "naïve" or language picture of the world, The unified and integrated theory of linguistic description, Lexicographic types, Lexico-semantic paradigms, Interaction of meanings in the texts, Interaction of lexical meanings, Semantic interaction of lexical and grammatical items.

**UNIT 3:**

**Ontology Technology-** Ontology definitions and basics, The Value of Ontology, Sense Relations and Ontologies, Ontology Classification Based on Context Dependency, Ontology Maturity, Maturing Process, Fine-Grained Lexical-Semantic, Relation between ontologies, modularization of ontologies, operation on ontologies.

**UNIT 4:**

**Semantic Web & Semantic Interoperability** -Semantic Web Architecture, Resource Description Framework (RDF), RDF Graph and Syntax, RDF Elements, RDF schema, Web Ontology Language (OWL). RDF query language SPARQL, Interoperability definition, Semantic Interoperability, Semantic Interoperability Architecture.

**UNIT 5:**

**Practical Text Analysis For Semantic Analytics-** The Fundamentals of Content Analysis, Planning for Text Analytics, Text Preprocessing, Semantic Space Representation and Latent Semantic Analysis, Cluster Analysis: Modeling Groups in Text, Probabilistic Topic Models, Classification Analysis: Machine Learning Applied to Text, Modeling Text Sentiment: Learning and Lexicon Models.

**Text Books:**

1. Book: Practical Text Analytics Maximizing the Value of Text Data Author Names: Murugan Anandarajan, Chelsey Hill, Thomas Nolan  
(<https://link.springer.com/book/10.1007/978-3-319-95663-3>)
2. Book: Semantic Analysis, Author Names: Cliff Goddard, Andrea C Schalley link: ([file:///C:/Users/admin/Downloads/Goddard\\_Schalley\\_NLP\\_2010.PDF](file:///C:/Users/admin/Downloads/Goddard_Schalley_NLP_2010.PDF))

**References**

1. Book: Principles of Systematic Lexicography, Juri D . Apresjan
2. Primary health information standard system based on semantic interoperability author: [Xia Zhao](#),<sup>1</sup> [Xiaohua Li](#),

**MASTER OF COMPUTER APPLICATIONS**  
**III Year I Semester**  
**MC5L10 ANDROID APPLICATION DEVELOPMENT LAB**

**Objectives:**

To learn how to develop Applications in android environment.

To learn how to develop user interface applications.

To learn how to develop URL related applications.

The student is expected to be able to do the following problems, though not limited.

1. Create an Android application that shows Hello + name of the user and run it on an emulator. (b) Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.
2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a “Back” button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.
4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.
5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message.

7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.
10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.
12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.
13. Create an application that shows the given URL (from a text field) in a browser.
14. Develop an application that shows the current location's latitude and longitude continuously as the device is moving (tracking).
15. Create an application that shows the current location on Google maps.

Note:

Android Application Development with MIT App Inventor: For the first one week, the student is advised to go through the App Inventor from MIT which gives insight into the various properties of each component.

The student should pay attention to the properties of each components, which are used later in Android programming. Following are useful links:

1. <http://ai2.appinventor.mit.edu>
2. [https://drive.google.com/file/d/0B8rTtW\\_91YclTWF4czdBMEpZcWs/view](https://drive.google.com/file/d/0B8rTtW_91YclTWF4czdBMEpZcWs/view)

**MC5L20**

**MASTER OF COMPUTER APPLICATIONS  
III Year I Semester  
DATA SCIENCE LAB**

Objectives

- To make students understand learn about a Big Data –R Programming , way of solving problems.
- To teach students to write programs in Scala to solve problems.

**Introduction to R Programming:**

What is R and RStudio? R is a statistical software program. It has extremely useful tools for data exploration, data analysis, and data visualization. It is flexible and also allows for a dvanced programming. RStudio is a user interface for R, which provides a nice environment for working with R.

1. Write an R program to evaluate the following expression  $ax+b/ax-b$ .
2. Write an R program to read input from keyboard(hint: readLine()).
3. Write an R program to find the sum of n natural numbers:  $1+2+3+4+\dots+n$
4. Write an R program to read n numbers.
  - (i) Sum of all even numbers
  - (ii) Total number of even numbers.
5. Write an R program to read n numbers.
  - (i) Total number of odd numbers
  - (ii) Sum of all odd numbers
6. Write an R program to to obtain
  - (i)sum of two matrices A and B
  - (ii) subtraction of two matrices A and B
  - (iii) Product of two matrices.
7. Write an R program for “declaring and defining functions “
8. Write an R program that uses functions to add n numbers reading from keyboard
9. Write an R program uses functions to swap two integers.
10. Write an R program that use both recursive and non-recursive functions for implementing the Factorial of a given number, n .
11. Write an R program to reverse the digits of the given number .{ example 1234 to be written as 4321 }
12. Write an R program to implement
  - (i)Linear search
  - (ii) Binary Search.
13. Write an R program to implement
  - (i)Bubble sort
  - (ii) selection sort .
14. Write a R program to implement the data structures

- (i) Vectors (ii) Array (iii) Matrix (iv) Data Frame (v) Factors
15. Write a R program to implement scan(),merge(), read.csv() and read.table() commands.
  16. Write an R program to implement “Executing Scripts” written on the note pad, by calling to the R console.
  17. Write a R program ,Reading data from files and working with datasets
    - (i) Reading data from csv files, inspection of data .
    - (ii) Reading data from Excel files .
  18. Write a R program to implement Graphs
    - (i)Basic high-level plots (ii)Modifications of scatter plots
    - (iii)Modifications of histograms, parallel box plots .

#### PART -2

#### Introduction to Scala Programming:

1. Write a scala program to demonstrate val and var
2. write a scala program to read data from keyboard
3. write a scala program to implement
  - (i) single dimensional array (ii) multi - dimensional array.
4. Write a scala program to implement classes, methods , creating objects
5. Write a scala program to returning a value to the main program.
6. Write a scala program to implement method overloading(Function Overloading)
7. Write a scala program to implement
  - (i)single inheritance (ii) multi level
8. Write a scala program to implement method overriding.
9. Write a scala program to implement Hierarchical inheritance
10. write a scala program to implement traits
11. Write a scala program to implement multiple inheritance
12. write a scala program to implement abstract classes.
13. write a scala program to implement from Collection : Vectors.
14. write a scala program to implement from Collection :MAPS.
15. Write a scala program to implement from collection : LIST.
16. write a scala program implement the statement “ traits can be inherited”.
17. write a scala program implement the statement “ abstract classes” can be

inherited”.

18. write a scala program implement the statement “ abstract classes” can be inherited”.

**Suggested Books for Lab:**

1. Big data – Black Book : 2015 edition: dreamtech press. Pg.(490- 642)
2. Introducing to programming and problem solving by scala,mark c.lewis, lisa l.lacher. CRC press,second edition .

Suggested Links:

1. <https://www.tutorialspoint.com/scala/>
2. <https://www.tutorialspoint.com/r/>

**MASTER OF COMPUTER APPLICATIONS  
III Year I Semester**



**Objectives:**

1. The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.
2. Student has to take up another case study of his/her own interest and do the same what ever mentioned in first problem. Some of the ideas regarding case studies are given in reference books which were mentioned in theory syllabus can be referred for some idea.

**Problems on UML:****Unified Modeling Language Lab:**

Students are divided into batches of 5 each and each batch has to draw the following diagrams using UML for an ATM system whose description is given below.

UML diagrams to be developed are:

1. Use Case Diagram.
2. Class Diagram.
3. Sequence Diagram.
4. Collaboration Diagram.
5. State Diagram
6. Activity Diagram.
7. Component Diagram
8. Deployment Diagram.
9. Test Design.

**Description for an ATM System**

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

The ATM must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
3. A customer must be able to make a transfer of money between any two accounts linked to the card.
4. A customer must be able to make a balance inquiry of any account linked to the card.
5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.

The ATM will provide the customer with a printed receipt for each successful transaction

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a

customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

#### REFERENCE BOOKS

- 1.Learning UML 2.0,Russ Miles and Kim Hamilton,O'Reilly,SPD.
- 2.Mastering UML with Rational Rose,W.Boggs&M.Boggs,Wiley India.