

**ACADEMIC REGULATIONS,  
COURSE STRUCTURE & SYLLABI  
FOR**

**M. Tech. (SOFTWARE ENGINEERING)**

Two Year PG Day-Time Program  
(with effect from 2017 – 18)



**SCHOOL OF INFORMATION TECHNOLOGY  
(AUTONOMOUS)  
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD  
Kukatpally, Hyderabad – 500 085 TELANGANA.**

**SCHOOL OF INFORMATION TECHNOLOGY  
(AUTONOMOUS)  
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD  
Kukatpally, Hyderabad – 500 085, Telangana (India)**

**ACADEMIC REGULATIONS FOR THE AWARD OF M.C.A DEGREE BASED ON  
CHOICE BASED CREDIT SYSTEM (CBCS)  
(WITH EFFECT FROM THE ACADEMIC YEAR 2017 – 2018)**

The Master of Technology (M.Tech.) Post Graduate 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 2 Years (4 Semesters) Master of Technology (M.Tech.) Post Graduate Degree program, under Choice Based Credit System (CBCS) at its constituent Autonomous College – **SCHOOL OF INFORMATION TECHNOLOGY (SIT), JNTUH**, Hyderabad, in the following specializations

<b>S.No.</b>	<b>Specialization</b>
1	Computer Science(CS)
2	Software Engineering(SE)
3	Computer Networks & Information Security(CNIS)

## **1.0 ELIGIBILITY FOR ADMISSIONS**

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

Admissions shall be made on the basis of merit rank obtained by the qualifying candidate on the basis of GATE score or at an Entrance Test (TSPGECET) conducted by TELANGANA State Government, subject to reservations prescribed by the University time to time.

## **2.0 AWARD OF M. Tech. DEGREE**

**2.1** A student shall be declared eligible for the award of the M. Tech. Degree, if he pursues a course of study in not less than two and not more than four academic years. However, he is permitted to write the examinations for two more years after four academic years of course work, failing which he shall forfeit his seat in M. Tech. programme.

**2.2** The student shall register for all 92 credits and secure all the 92 credits.

**2.3** The minimum instruction days in each semester are 90.

## **3.0 M.TECH. PROGRAMS STRUCTURE**

- 3.1** The M.Tech. Programs of SIT-JNTUH are of Semester Pattern, with 4 Semesters constituting 2 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:** Each PG program is of 2 Academic Years (4 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:** Subjects/ Courses are to be registered by a student 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/ Semester for Theory/ Lecture (L) Courses

One Credit - for Two hours/ Week/ Semester for Laboratory/ Practical (P) Courses

#### **4.0 COURSE WORK**

- 4.1** A candidate after securing admission shall pursue the M.Tech. in a minimum period of 2 Academic Years, and a maximum period of 4 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.Tech. Degree in respective specialization.
- 4.3** Each of I Year I Semester and II Semester offers 26 Credits(I Year =  $2 \times 26 = 52C$ ) and II Year I Semester and II Semester offer 20 credits(II Year =  $2 \times 20 = 40C$ ) each, totaling to 92 Credits (92 C) for the entire M.Tech. Program.
- 4.4 The student shall register for all 92 credits and secure all the 92 credits.**

## 5 COURSE REGISTRATION

- 5.1 A 'Course Coordinator or Faculty Advisor' shall be assigned to each student, who will advise him about the M.Tech. 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 student's apriori (before the beginning of the Semester). The Registration Requests for any 'CURRENT SEMESTER' shall be completed BEFORE the commencement of SEEs (Semester End Examiantions) 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 of CHOICE with a typical total of 26 Credits per Semester (I & II Semesters): **Minimum being 22 Credits and Maximum being 30 Credits**, 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 course mandatorily and he has to 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 CGPA.
- 5.6 Choice for 'additional Subjects/ Courses' to reach the Maximum Permissible Limit of 30 Credits (above the typical 26 Credit norm) must be clearly indicated, which needs the specific approval and signature of the Course Coordinator.
- 5.7 If the Student 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, and CAN NOT be inter-changed; further, alternate choices will also not be considered. 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 Student 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 Course Coordinator/Faculty Advisor (subject to retaining a minimum of 22 C), 'within 15 Days of Time' from the beginning of the current Semester.

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

**6.1** A Subject/ Course may be offered to the Students, ONLY IF a Minimum of 1/3<sup>rd</sup> of the Section Strength) opt for the same.

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

**6.3** 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 .

**6.4** In case of options coming from Students of other Departments/ Branches/ Disciplines (not considering OPEN ELECTIVES), PRIORITY shall be given to the student of the 'Parent Department' first.

## **7 ATTENDANCE**

**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.

**7.4** 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.

**7.5** Students 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.

**7.6** 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 M.Tech 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 practicals, on the basis of Internal Evaluation and End Semester Examination.

- 8.1** Evaluation of a student 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 60 marks are allocated. The remaining 40 marks are allocated to internal evaluation.
- 8.2** In internal evaluation, 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 Semesters Examination will be conducted for 60 marks. It consists of eight Questions carries 12 marks each covering the entire syllabus. The student should answer any five questions out of eight.
- 8.4** For practical subjects, 60 marks shall be awarded based on the performance in the End Semester Examinations and 40 marks shall be awarded based on the day-to-day performance as Internal Marks.
- 8.5** A candidate shall be deemed to have secured the minimum academic requirement in a subject if he secures a minimum of 40% of marks in the Semester End Examination and a minimum aggregate of 50% of the total marks in the Semester End 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 two 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 student 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 thesis in the first 2 weeks of the II Year I 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 other faculty members from the school.
- 9.3** Only after obtaining the approval of Project Review Committee (PRC), the student 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 Departmental Academic Committee. However, the Departmental 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 44 weeks which is spread across 22 weeks in II Year I semester and 22 Weeks in II year II semester. The student can submit the final project only after 40 weeks from the date of registration after the approval of PRC.
- 9.6** Internal Evaluation of the project shall be on the basis of the seminars (Project reviews) conducted during the II Year I semester and II semesters by the Project Review Committee (PRC). In II Year I semester Two seminars for 100 marks are conducted and in II Year II semester two seminars each of 100 marks are conducted.
- 9.7** At end of the II Year I semester, a candidate shall submit status report in a spiral bound copy form. He will attend for the internal viva-voce conducted by the PRC (Internal Viva-Voce only) which is evaluated for 100 marks.
- 9.8** At end of the II Year II semester, after approval from the PRC, the soft copy of the thesis should be submitted for ANTI-PLAGIARISM for the quality check

and the plagiarism report should be included in the final thesis. If the copied information is less than 24%, then only thesis will be accepted for submission. A candidate shall submit the thesis/dissertation in a hard bound copy form. He will attend for the viva-voce conducted by the PRC (External Viva-Voce) which is evaluated for 100 marks.

- 9.9** 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.
- 9.10** The Thesis/Dissertation will be adjudicated by one external examiner selected by the competent authority.
- 9.11** If the report of the examiner is favorable, viva-voce examination shall be conducted by a board consisting of the supervisor, Head of the Department and the examiner who adjudicated the Thesis. The Board shall jointly adjudicates the project for 100 marks. 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 he fails at the second viva-voce examination, he will not be eligible for the award of the degree unless the candidate is asked to revise and resubmit.
- 9.12** If the report of the examiner is not favourable, the candidate shall revise and resubmit the Thesis, within the time frame as described by PRC. If the report of the examiner is unfavourable again, the thesis shall be summarily rejected.

## **10 Examinations and Assessment - The Grading System**

- 10.1** Marks will be awarded to indicate the performance of each student in each Theory Subject, or Lab/Practicals, or Seminar, or Project, etc., based on the % marks obtained in IE + SEE (Internal Evaluation + Semester End Examination, both taken together) as specified in Item 6 above, and a corresponding Letter Grade shall be given.
- 10.2** As a measure of the student's 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



- 10.3** A student 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.
- 10.4** A student 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.
- 10.5** A Letter Grade does not imply any specific Marks percentage and it will be the range of marks percentage.
- 10.6** In general, a student shall not be permitted to repeat any Subject/ Course (s) only for the sake of ‘Grade Improvement’ or ‘CGPA Improvement’.
- 10.7** A student earns Grade Point (GP) in each Subject/ Course, on the basis of the Letter Grade obtained by him in that Subject/ Course. The corresponding ‘Credit Points’ (CP) are computed by multiplying the Grade Point with Credits for that particular Subject/ Course.

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

- 10.8 The Student passes the Subject/ Course only when he **gets GP ≥ 5 (C Grade or above)**.
- 10.9 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

$$GPA = \frac{\sum_{i=1}^n C_i \times GP_i}{\sum_{i=1}^n C_i}$$

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  $i^{th}$  Subject, and  $GP_i$  represents the Grade Points (GP) corresponding to the Letter Grade awarded for that  $i^{th}$  Subject.

**10.10** The Cumulative Grade Point Average (CGPA) is a measure of the overall cumulative performance of a student over all Subjects in all considered for registration. The CGPA is the ratio of the Total Credit Points secured by a student 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  $j^{th}$  subject.

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

10.11 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 Student 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 (PGP), and secures the required number of **92** Credits (with CGPA  $\geq 5.0$ ), shall be declared to have 'QUALIFIED' for the award of the M.Tech. Degree in the chosen Branch of Engineering and Technology with specialization as he admitted.

### 11.2 Award of Class

After a student has satisfied the requirements prescribed for the completion of the and is eligible for the award of M. Tech. 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 student 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 student has not paid the dues, if any, to the University or if any case of indiscipline is pending against him, the result of the student 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 admission to maximum of two earlier 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 students with effect from the dates notified by the University.

## MALPRACTICES RULES

### DISCIPLINARY ACTION FOR / IMPROPER CONDUCT IN EXAMINATIONS

	Nature of Malpractices/Improper conduct	Punishment
	<i>If the candidate:</i>	
1.	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 (material shall include any marks on the body of the candidate which can be used as an aid in the subject of the examination)	Expulsion from the examination hall and cancellation of the performance in that subject only.
(a)		
(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	Expulsion from the examination hall and cancellation of performance in that subject

	to send out the question paper during the examination or answer book or additional sheet, during or after the examination.	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 words, either spoken or written or 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 students 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 student 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.	Student 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.	

### 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

**COURSE STRUCTURE (CBCS) FOR P.G. PROGRAMMES  
(2017-2018)**

**I Year**

I Semester		Course Title	Int. marks	Ext. marks	L	P	C
1.	Core I		40	60	4	--	4
2.	Core II		40	60	4	--	4
3.	Core III		40	60	4	--	4
4.	Professional Elective I		40	60	4	--	4
5.	Professional Elective II		40	60	4	--	4
6.	Lab I		40	60	--	4	2
7.	Lab II		40	60	--	4	2
8.	Lab III		40	60	--	4	2
9.	Audit Course(Theory)				4	--	4
		<b>Total Credits</b>			<b>20</b>	<b>12</b>	<b>26</b>

II Semester		Course Title	Int. marks	Ext. marks	L	P	C
1.	Core IV		40	60	4	--	4
2.	Core V		40	60	4	--	4
3.	Professional Elective III		40	60	4	--	4
4.	Professional Elective IV		40	60	4	--	4
5.	Open Elective		40	60	4	--	4
6.	Lab IV		40	60	--	4	2
7.	Lab V		40	60	--	4	2
8.	Lab VI		40	60	--	4	2
9.	Audit Course(Lab)				--	4	2
		<b>Total Credits</b>			<b>20</b>	<b>12</b>	<b>26</b>

**II Year**

I Semester		Int. marks	Ext. marks	L	P	C
1.	<b>Project Seminar – I</b>	100	--	--	--	2
2.	<b>Project Seminar – II</b>	100	--	--	--	2
3.	Project Viva-Voce(Internal Valuation)	100	--	--	32	16
		<b>Total Credits</b>			--	<b>20</b>

II Semester		Int. marks	Ext. marks	L	P	C
1	<b>Project Seminar – III</b>	100	--	--	--	2
2	<b>Project Seminar – IV</b>	100	--	--	--	2
3	Project Viva-Voce(External Valuation)	--	100	--	32	16
		<b>Total Credits</b>			--	<b>20</b>

**SCHOOL OF INFORMATION TECHNOLOGY**  
**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERAB**  
**MASTER OF TECHNOLOGY (SOFTWARE ENGINEERING)**

**I YEAR I SEMESTER**

	SCODE	SUBJECTS	L	P	MAX.MARKS		Total Marks	CREDITS
					Int.	Ext.		
<b>Core – I</b>	SE1C10	Data Structures and Algorithms	4	-	40	60	100	<b>4</b>
<b>Core – II</b>	SE1C20	Software Development Methodologies	4	-	40	60	100	<b>4</b>
<b>Core – III</b>	SE1C30	Software Requirements and Estimation	4	-	40	60	100	<b>4</b>
<b>Professional Elective – I</b>	SE1P11	Computer Networking	4	-	40	60	100	<b>4</b>
	SE1P12	Internet Technologies and Services						
	SE1P13	Android Application Development						
	SE1P14	Data Mining						
<b>Professional Elective – II</b>	SE1P21	Object Oriented Modeling	4	-	40	60	100	<b>4</b>
	SE1P22	Information Retrieval systems						
	SE1P23	Machine Learning						
	SE1P24	Distributed Systems						
	SE1P25	Cloud Computing						
<b>Lab – I</b>	SE1L10	Data Structures through Java Lab	-	4	40	60	100	<b>2</b>
<b>Lab – II</b>	SE1L20	Software Engineering Lab	-	4	40	60	100	<b>2</b>
<b>Lab - III</b>	<b>Professional Elective-I Lab</b>		-	4	40	60	100	<b>2</b>
	SE1L31	Computer Networking Lab						
	SE1L32	Internet Technologies and Services Lab						
	SE1L33	Android Application Development Lab						
	SE1L34	Data Mining Lab						
Audit Course	SE1A10	Professional Communication Skills	4	-	-	-	-	<b>4</b>
		<b>TOTAL CREDITS</b>						<b>26</b>



**MASTER OF TECHNOLOGY (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

	SCODE	SUBJECTS	L	P	MAX.MARKS		Total Marks	CREDITS
					Int.	Ext.		
<b>Core – IV</b>	SE2C10	Software Architecture and Design Patterns	4	-	40	60	100	<b>4</b>
<b>Core – V</b>	SE2C20	Software Quality Assurance and Testing	4	-	40	60	100	<b>4</b>
<b>Professional Elective – III</b>	SE2P31	Network Security	4	-	40	60	100	<b>4</b>
	SE2P32	Data Analytics						
	SE2P33	Internet Of Things						
	SE2P34	Computer Forensics						
<b>Professional Elective – IV</b>	SE2P41	Software Process and Project Management	4	-	40	60	100	<b>4</b>
	SE2P42	Component Based Software Engineering						
	SE2P43	Software Security Engineering						
	SE2P44	Business Process Management						
<b>Open Elective</b>	SE2O11	Image Processing and Pattern Recognition	4	-	40	60	100	<b>4</b>
	SE2O12	Multimedia and Rich Internet Applications						
	SE2O13	Bioinformatics						
	SE2O14	Cyber Security						
	SE2O15	Intellectual Property Rights						
	SE2O16	Network Programming						
	SE2O17	Scripting Language						
<b>Lab – IV</b>	SE2L10	Design Patterns Lab	-	4	40	60	100	<b>2</b>
<b>Lab – V</b>	SE2L20	Software Testing Lab	-	4	40	60	100	<b>2</b>
<b>Lab - VI</b>	<b>Professional Elective-III Lab</b>		-	4	40	60	100	<b>2</b>
	SE2L31	Network Security Lab						
	SE2L32	Data Analytics Lab						
	SE2L33	Internet Of Things Lab						
	SE2L34	Computer Forensics Lab						
	SE2A10	Soft Skills Lab	-	4	-	-	-	<b>2</b>
		<b>TOTAL CREDITS</b>						<b>26</b>

**MASTER OF TECHNOLOGY (SOFTWARE ENGINEERING)**

**II YEAR I SEMESTER**

SCODE	SUBJECTS	L	P	MAX.MARKS		Total Marks	CREDITS
				Int.	Ext.		
SE3S10	<b>Project Seminar – I</b>	-	-	100	-	100	2
SE3S20	<b>Project Seminar – II</b>	-	-	100	-	100	2
SE3P10	<b>Project Viva- Voce(Internal Valuation)</b>	-	32	100	-	100	16
	<b>TOTAL CREDITS</b>						<b>20</b>

**MASTER OF TECHNOLOGY (SOFTWARE ENGINEERING)  
II YEAR II SEMESTER**

SCODE	SUBJECTS	L	P	MAX.MARKS		Total Marks	CREDITS
				Int.	Ext.		
SE4S10	<b>Project Seminar – III</b>	-	-	100	-	100	2
SE4S20	<b>Project Seminar – IV</b>	-	-	100	-	100	2
SE4P10	<b>Project Viva- Voce(External Valuation)</b>	-	32	-	100	100	16
	<b>TOTAL CREDITS</b>						<b>20</b>

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1C10**

**DATA STRUCTURES AND ALGORITHMS  
(CORE – I)**

**Objectives:**

- The fundamental design, analysis, and implementation of basic data structures.
- Basic concepts in the specification and analysis of programs.
- Principles for good program design, especially the uses of data abstraction.
- Significance of algorithms in the computer field
- Various aspects of algorithm development
- Qualities of a good solution

**UNIT I**

Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation- Big Oh, Omega and Theta notations, Complexity Analysis Examples.

Data structures-Linear and non linear data structures, ADT concept, Linear List ADT, Array representation, Linked representation, Vector representation, singly linked lists -insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, circular lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

**UNIT II**

Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack, implementation of recursion, Circular queue-insertion and deletion, Dequeue ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap, java.util package-ArrayList, Linked List, Vector classes, Stacks and Queues in java.util, Iterators in java.util.

**UNIT III**

Searching–Linear and binary search methods, Hashing-Hash functions, Collision Resolution methods-Open Addressing, Chaining, Hashing in java.util-HashMap, HashSet, Hashtable.

Sorting –Bubble sort, Insertion sort, Quick sort, Merge sort, Heap sort, Radix sort, comparison of sorting methods.

**UNIT IV**

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree ADT, representations, recursive and non recursive traversals, Java code for traversals, Threaded binary trees. Graphs- Graphs terminology, Graph ADT, representations, graph traversals/search methods-dfs and bfs, Java code for graph traversals, Applications of Graphs-Minimum cost spanning tree using Kruskal's algorithm, Dijkstra's algorithm for Single Source Shortest Path Problem.

## **UNIT V**

Search trees- Binary search tree-Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees-Definition and examples only, Red Black trees – Definition and examples only, B-Trees-definition, insertion and searching operations, Trees in java.util- TreeSet, Tree Map Classes, Tries(examples only),Comparison of Search trees.

Text compression-Huffman coding and decoding, Pattern matching-KMP algorithm.

### **TEXT BOOKS:**

1. Data structures, Algorithms and Applications in Java, S.Sahni, Universities Press.
2. Data structures and Algorithms in Java, Adam Drozdek, 3<sup>rd</sup> edition, Cengage Learning.
3. Data structures and Algorithm Analysis in Java, M.A.Weiss, 2nd edition, Addison-Wesley (Pearson Education).

### **REFERENCES:**

1. Java for Programmers, Deitel and Deitel, Pearson education.
2. Data structures and Algorithms in Java, R.Lafore, Pearson education.
3. Java: The Complete Reference, 8<sup>th</sup> editon, Herbert Schildt, TMH.
4. Data structures and Algorithms in Java, M.T.Goodrich, R.Tomassia, 3<sup>rd</sup> edition, Wiley.
5. Data structures and the Java Collection Frame work,W.J.Collins, Mc Graw Hill.
6. Classic Data structures in Java, T.Budd, Addison-Wesley (Pearson Education).
7. Data structures with Java, Ford and Topp, Pearson Education.
8. Data structures using Java, D.S.Malik and P.S.Nair, Cengage learning.
9. Data structures with Java, J.R.Hubbard and A.Huray, PHI Pvt. Ltd.
10. Data structures and Software Development in an Object-Oriented Domain, J.P.Tremblay and G.A.Cheston, Java edition, Pearson Education.
11. A Practical Guide to Data Structures and Algorithms using Java, S.Goldman & K.Goldman, Chapman & Hall/CRC, Taylor & Francis Group.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1C20**

**SOFTWARE DEVELOPMENT METHODOLOGIES  
(CORE – II)**

**Objectives:**

Your studies will enable you to develop:

- a broad and critical understanding of all the processes for engineering high quality software and the principles, concepts and techniques associated with software development
- an ability to analyze and evaluate problems and draw on the theoretical and technical knowledge to develop solutions and systems
- a range of skills focused on the analysis of requirements, design and implementation of reliable and maintainable software, with strong emphasis on engineering principles applied over the whole development lifecycle
- an awareness of current research in software development, the analytical skills and research techniques for their critical and independent evaluation and their application to new problems.

**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.**Configuration Management:** Configuration Management planning, Change management, Version and release management, System building, CASE tools for configuration management.

## TEXT BOOKS:

1. Software Engineering: A practitioner's Approach, Roger S Pressman, sixth edition. McGraw Hill 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 3: Domains, Requirements and Software Design, D.Bjorner, Springer, International Edition.
9. Software Engineering Principles and Practice, Hans Van Vliet, 3<sup>rd</sup> edition, Wiley India edition.
10. Introduction to Software Engineering, R.J.Leach, CRC Press.
11. Software Engineering Fundamentals, Ali Behforooz and Frederick J.Hudson, Oxford University Press, rp2009
12. Software Engineering Handbook, Jessica Keyes, Auerbach, 2003.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1C30**

**SOFTWARE REQUIREMENTS AND ESTIMATION  
(CORE – III)**

**Objectives:**

- Students will demonstrate knowledge of the distinction between critical and non- critical systems.
- Students will demonstrate the ability to manage a project including planning, scheduling and risk assessment/management.
- Students will author a software requirements document.
- Students will demonstrate an understanding of the proper contents of a software requirements document.
- Students will author a formal specification for a software system.
- Students will demonstrate an understanding of distributed system architectures and application architectures.
- Students will demonstrate an understanding of the differences between real-time and non-real time systems.
- Students will demonstrate proficiency in rapid software development techniques.
- Students will demonstrate proficiency in software development cost estimation
- Students will author a software testing plan.

**UNIT I**

**Software Requirements: What and Why** Essential Software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management **Software Requirements Engineering** Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality.

**UNIT II**

**Software Requirements Management** Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain **Software Requirements Modeling** Use Case Modeling, Analysis Models, Dataflow diagram, state transition diagram, class diagrams, Object analysis, Problem Frames

**UNIT III**

**Software Estimation** Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation

**Size Estimation**

Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, Conversion between size measures.

**UNIT IV**

*School of IT, JNTU Hyderabad – 500 085 TELANGANA :: M. Tech. (SE) :: w.e.f. 2017 - 18*

**Effort, Schedule and Cost Estimation** What is Productivity? Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Putnam Estimation Model, Algorithmic models, Cost Estimation

## **UNIT V**

**Tools for Requirements Management and Estimation** **Requirements Management Tools:** Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation, **Software Estimation Tools:** Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools

### **TEXT BOOK:**

1. Software Requirements and Estimation by *Rajesh Naik and Swapna Kishore*, Tata Mc Graw Hill.

### **REFERENCES:**

1. Software Requirements by Karl E. Weigers, Microsoft Press.
2. Managing Software Requirements, Dean Leffingwell & Don Widrig, Pearson Education, 2003.
3. Mastering the requirements process, second edition, Suzanne Robertson & James Robertson, Pearson Education, 2006.
4. Estimating Software Costs, Second edition, Capers Jones, TMH, 2007.
5. Practical Software Estimation, M.A. Parthasarathy, Pearson Education, 2007.
6. Measuring the software process, William A. Florac & Anita D. Carleton, Pearson Education, 1999.



**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P11**

**COMPUTER NETWORKING  
(PROFESSIONAL ELECTIVE - I)**

**Objectives:**

- The objective of this course is to build a solid foundation in computer networks concepts and design
- To understand computer network architectures, protocols, and interfaces.
- The OSI reference model and the Internet architecture network applications.
- The course will expose students to the concepts of traditional as well as modern day computer networks - wireless and mobile, multimedia-based.
- Students completing this course will understand the key concepts and practices employed in modern computer networking

**UNIT –I**

What Is the Internet?, The Network Edge , The Network Core , Delay, Loss, and Throughput in Packet-Switched Networks , Protocol Layers and Their Service Models, Networks Under Attack, History of Computer Networking and the Internet.

**UNIT – II**

**Transport Layer:** Introduction and Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Connection-Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control.

**UNIT-III**

**The Network Layer:** Introduction, Virtual Circuit and Datagram Networks, What's Inside a Router?, The Internet Protocol (IP): Forwarding and Addressing in the Internet , Routing Algorithms, Routing in the Internet, Broadcast and Multicast Routing.

**UNIT – IV**

**The Link Layer:** Links, Access Networks, and LANs, Introduction to the Link Layer, Error-Detection and -Correction Techniques, Multiple Access Links and Protocols, Switched Local Area Networks, Link Virtualization: A Network as a Link Layer, Data Center Networking, Retrospective: A Day in the Life of a Web Page Request.

**UNIT – V**

**Wireless and Mobile Networks:** Introduction, Wireless Links and Network Characteristics, WiFi: 802.11 Wireless LANs, Cellular Internet Access, **Mobility Management:** Principles,

Mobile IP, Managing Mobility in Cellular Networks, **Wireless and Mobility: Impact on Higher-Layer Protocols.**

**TEXT BOOKS:**

1. Computer Networking: A Top Down Approach , *James F. Kurose, Keith W.Ross*, 6<sup>th</sup> Edition.
2. Data Communications and Networking, *Behrouz A. Forouzan*, Fourth Edition, Tata McGraw Hill
3. High Speed Networks and Internets – Performance and Quality of Service, *William Stallings*, Second Edition, Pearson Education.
4. Top-Down Network Design, *Priscilla Oppenheimer*, Second Edition, Pearson Education (CISCO Press)

**REFERENCES:**

1. Computer Networks by Mayank Dave, Cengage.
2. Guide to Networking Essentials, *Greg Tomsho, Ed Tittel, David Johnson*, Fifth Edition, Thomson.
3. Computer Networks, *Andrew S. Tanenbaum*, Fourth Edition, Prentice Hall.
4. An Engineering Approach to Computer Networking, *S.Keshav*, Pearson Education.
5. Campus Network Design Fundamentals, *Diane Teare, Catherine Paquet*, Pearson Education (CISCO Press)
- 6 Computer Communications Networks, Mir, Pearson Education.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P12**

**INTERNET TECHNOLOGIES AND SERVICES  
(PROFESSIONAL ELECTIVE - I)**

**Objective:**

The student who has knowledge of programming with java should be able to develop web based solutions using multi-tier architecture. S/he should have good understanding of different technologies on client and server side components as Follows:

Client Side: HTML5, CSS3, Javascript, Ajax, JQuery and JSON

Server Side: Servlets, JSP

Database: MySQL with Hibernate and Connection Pooling

Framework: Struts with validation framework, Internationalization (I18N)

SOA: Service Oriented Architecture, Web services fundamentals, Axis framework for WS

**UNIT I**

**Client Side Technologies:** Overview of HTML - Common tags, XHTML, capabilities of HTML5 Cascading Style sheets, CSS3 enhancements, linking to HTML Pages, Classes in CSS Introduction to JavaScripts, variables, arrays, methods and string manipulation, BOM/DOM (Browser/Document Object Model), accessing elements by ID, Objects in JavaScript Dynamic HTML with JavaScript and with CSS, form validation with JavaScript, Handling Timer Events Simplifying scripting with JQuery, JASON for Information exchange.

**UNIT II**

**Introduction to Java Servlets:** Introduction to Servlets: Lifecycle of a Servlet, Reading request and initialization parameters, Writing output to response, MIME types in response, Session Tracking: Using Cookies and Sessions, Steps involved in Deploying an application Database Access with JDBC and Connection Pooling Introduction to XML, XML Parsing with DOM and SAX Parsers in Java Ajax - Ajax programming with JSP/Servlets, creating XML Http Object for various browsers, Sending request, Processing response data and displaying it. Introduction to Hibernate

**UNIT III**

**Introduction to JSP:** JSP Application Development: Types of JSP Constructs (Directives, Declarations, Expressions, Code Snippets), Generating Dynamic Content, Exception Handling, Implicit JSP Objects, Conditional Processing, Sharing Data Between JSP pages, Sharing Session and Application Data, Using user defined classes with jsp:useBean tag, Accessing a Database from a JSP.

## UNIT IV

**Introduction to Struts Framework:** Introduction to MVC architecture, Anatomy of a simple struts2 application, struts configuration file, Presentation layer with JSP, JSP bean, html and logic tag libraries, Struts Controller class, Using form data in Actions, Page Forwarding, validation frame work, Internationalization.

## UNIT V

**Service Oriented Architecture and Web Services:** Overview of Service Oriented Architecture – SOA concepts, Key Service Characteristics, Technical Benefits of a SOA Introduction to Web Services– The definition of web services, basic operational model of web services, basic steps of implementing web services. Core fundamentals of SOAP – SOAP Message Structure, SOAP encoding, SOAP message exchange models, Describing Web Services –Web Services life cycle, anatomy of WSDL Introduction to Axis– Installing axis web service framework, deploying a java web service on axis. Web Services Interoperability – Creating java and .Net client applications for an Axis Web Service

(Note: The Reference Platform for the course will be open source products Apache Tomcat Application Server, MySQL database, Hibernate and Axis)

### TEXT BOOKS:

1. Web Programming, building internet applications, Chris Bates 3<sup>rd</sup> edition, WILEY Dreamtech .
2. The complete Reference Java 7<sup>th</sup> Edition , Herbert Schildt., TMH.
3. Java Server Pages, Hans Bergsten, SPD, O'Reilly.
4. Professional Jakarta Struts - James Goodwill, Richard Hightower, Wrox Publishers.
5. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India, rp – 2008.
6. Understanding SOA with Web Services, Eric Newcomer and Greg Lomow, Pearson Edition – 2009
7. Java Web Service Architecture, James McGovern, Sameer Tyagi et al., Elsevier – 2009

### REFERENCES:

1. Programming the world wide web, 4<sup>th</sup> edition, R.W. Sebesta, Pearson
2. Core SERVLETS AND JAVASERVER PAGES VOLUME 1: CORE
3. TECHNOLOGIES , Marty Hall and Larry Brown Pearson
4. Internet and World Wide Web – How to program , Dietel and Nieto PHI/Pearson.
5. Jakarta Struts Cookbook , Bill Siggelkow, S P D O'Reilly.
6. Professional Java Server Programming, S. Allamaraju & others Apress(dreamtech).
7. Java Server Programming , Ivan Bayross and others, The X Team, SPD
8. Web Warrior Guide to Web Programming-Bai/Ekedaw-Cengage Learning.
9. Beginning Web Programming-Jon Duckett , WROX.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P13**

**ANDROID APPLICATION DEVELOPMENT  
(PROFESSIONAL ELECTIVE-I)**

**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

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P14**

**DATA MINING  
(PROFESSIONAL ELECTIVE - I)**

**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.



**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P21**

**OBJECT ORIENTED MODELING  
(PROFESSIONAL ELECTIVE - II)**

**Objectives:**

- Concisely define the following key terms: class, object, state, behavior, object class, class diagram, object diagram, operation, encapsulation, constructor operation, query operation, update operation, scope operation, association, association role, multiplicity, association class, abstract class, concrete class, class-scope attribute, abstract operation, method, polymorphism, overriding, multiple classification, aggregation, and composition.
- To describe the activities in the different phases of the object-oriented development life cycle.
- State the advantages of object-oriented modeling vis-à-vis structured approaches.
- Compare and contrast the object-oriented model with the E-R and EER models.
- Model a real-world application by using a UML class diagram.
- Provide a snapshot of the detailed state of a system at a point in time using a UML (Unified Modeling Language) object diagram.
- Recognize when to use generalization, aggregation, and composition relationships.
- Specify different types of business rules in a class diagram.

**UNIT I**

**Introduction to UML:** The meaning of Object Orientation, object identity, Encapsulation, information hiding, polymorphism, generosity, importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture.

**Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams.

**Class & Object Diagrams:** Terms, concepts, modeling techniques for Class & Object Diagrams.

**Collaboration Diagrams:** Terms, Concepts, depicting a message, polymorphism in collaboration diagrams, iterated messages, use of self in messages.

**Sequence Diagrams:** Terms, concepts, depicting asynchronous messages with/without priority, callback mechanism, broadcast messages.

**UNIT II**

**Basic Behavioral Modeling:** Use cases, Use case Diagrams, Activity Diagrams.

**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 III**

**The Unified process:** use case driven, architecture centric, iterative, and incremental

**The Four Ps:** people, project, product, and process

**Use case driven process:** why use case, capturing use cases, analysis, design, and implementation to realize the use cases, testing the use cases

**Architecture-centric process:** architecture in brief, why we need architecture, use cases and architecture, the steps to architecture, an architecture description.

#### **UNIT IV**

**Iterative incremental process:** iterative incremental in brief, why iterative incremental development? The iterative approach is risk driven, the generic iteration.

**The Generic Iteration workflow:** phases are the first division workflow, planning proceeds doing, risks affect project planning, use case prioritization, resource needed, assess the iteration and phases

**Inception phase:** early in the inception phase, the archetypal inception iteration workflow, execute the core workflows, requirements to test.

#### **UNIT V**

**Elaboration Phase:** elaboration phase in brief, early in the elaboration phase, the architectural elaboration iteration workflow, execute the core workflows-Requirements to test.

**Construction phase:** early in the construction phase, the archetypal construction iteration workflow, execute the core workflow.

**Transition phase:** early in the transition phase, activities in transition phase

**Case Studies:** Automation of a Library, Software Simulator application (2-floor elevator simulator)

#### **TEXT BOOKS:**

1. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson 2<sup>nd</sup> Edition, Pearson Education.
2. UML 2 Toolkit by Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado WILEY-Dreamtech India Pvt. Ltd.
3. The Unified Software Development Process by Ivar Jacobson, Grady Booch, James Rumbaugh, Pearson Education

#### **REFERENCES:**

1. Fundamentals of Object Oriented Design in UML By Meilir Page-Jones, Pearson Education
2. Object Oriented Analysis & Design By Atul Kahate, The McGraw-Hill.
3. Practical Object-Oriented Design with UML By Mark Priestley, TATA Mc Graw Hill
4. Object Oriented Analysis & Design By Brett D McLaughlin, Gary Pollice and David West, O'REILY.
5. Object-Oriented Analysis and Design using UML by Simon Bennet, Steve McRobb and Ray Farmer, 2<sup>nd</sup> Edition, TATA Mc Graw Hill.
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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P22**

**INFORMATION RETRIEVAL SYSTEMS  
(PROFESSIONAL ELECTIVE - II)**

**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.

**REFERENCES :**

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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P23**

**MACHINE LEARNING**

**(PROFESSIONAL ELECTIVE – II)**

**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 Statistical Learning, Springer Verlag, 2001

## REFERENCE BOOKS:

1. Machine Learning Methods in the Environmental Sciences, Neural Networks, William W Hsieh, Cambridge Univ Press.
2. Richard o. Duda, Peter E. Hart and David G. Stork, pattern classification, John Wiley & Sons Inc.,2001
3. Chris Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995

## Outcomes:

- Student should be able to understand the basic concepts such as decision trees and neural networks.
- Ability to formulate machine learning techniques to respective problems.
- Apply machine learning algorithms to solve problems of moderate complexity

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P24**

**DISTRIBUTED SYSTEMS  
(PROFESSIONAL ELECTIVE –II)**

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;
- To design a distributed system that fulfils requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognize when this is not possible, and explain why;
- To build distributed system software using basic OS mechanisms as well as higher-level middleware and languages.

**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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1P25**

**CLOUD COMPUTING  
(PROFESSIONAL ELECTIVE - II)**

**Objectives:**

- 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 SmartCloud, 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**

**REFERENCES:**

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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1L10**

**DATA STRUCTURES THROUGH JAVA LAB  
(LAB – I)**

**Sample Problems on Data structures:**

1. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:
  - a) Linear search
  - b) Binary search
2. Write Java programs to implement the following using arrays and linked lists
  - a) List ADT
3. Write Java programs to implement the following using an array.
  - a) Stack ADT
  - b) Queue ADT
4. Write a Java program that reads an infix expression and converts the expression to postfix form. (use stack ADT).
5. Write a Java program to implement circular queue ADT using an array.
6. Write a Java program that uses both a stack and a queue to test whether the given string is a palindrome or not.
7. Write Java programs to implement the following using a singly linked list.
  - a) Stack ADT
  - b) Queue ADT
8. Write Java programs to implement the deque (double ended queue) ADT using
  - a) Array
  - b) Singly linked list
  - c) Doubly linked list.
9. Write a Java program to implement priority queue ADT.
10. Write a Java 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.
11. Write a Java program to implement all the functions of a dictionary (ADT) using Hashing.
12. Write a Javaprogram to implement Dijkstra's algorithm for Single source shortest path problem.
13. Write Java programs that use recursive and non-recursive functions to traverse the given binary tree in
  - a) Preorder
  - b) Inorder and
  - c) Postorder.
14. Write Java programs for the implementation of bfs and dfs for a given graph.
15. Write Java programs for implementing the following sorting methods:
  - a) Bubble sort
  - b) Insertion sort
  - c) Merge sort
  - d) Binary tree sort
  - e) Heap sort

- c) Quick sort                      f) Radix sort
16. Write a Java program to perform the following operations:  
a) Insertion into a B-tree   b) Searching in a B-tree
17. Write a Java program that implements Kruskal's algorithm to generate minimum cost spanning tree.
18. Write a Java program that implements KMP algorithm for pattern matching.

### **REFERENCES:**

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. Design and Analysis of Algorithms, P.H.Dave and H.B.Dave, Pearson education.
  7. Data Structures and java collections frame work, W.J.Collins, Mc Graw Hill.
  8. Java: the complete reference, 7<sup>th</sup> editon, Herbert Schildt, TMH.
  9. 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.
  10. Java Programming, D.S.Malik, Cengage Learning.
  11. A Practical Guide to Data Structures and Algorithms using Java, S.Goldman & K.Goldman, Chapman & Hall/CRC, Taylor & Francis Group.
- ( Note: Use packages like `java.io`, `java.util`, etc)

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1L20**

**SOFTWARE ENGINEERING LAB  
(LAB – II)**

Objectives:

- To understand the software engineering methodologies involved in the phases for project development.
- To gain knowledge about open source tools used for implementing software engineering methods.
- To exercise developing product-startups implementing software engineering methods.

Open source Tools: StarUML / UMLGraph / Topcased

Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology for at least two real time scenarios or for the sample experiments.

1. Problem Analysis and Project Planning -Thorough study of the problem – Identify Project scope, Objectives and Infrastructure.
2. Software Requirement Analysis – Describe the individual Phases/modules of the project and Identify deliverables. Identify functional and non-functional requirements.
3. Data Modeling – Use work products – data dictionary.
4. Software Designing - Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.
5. Prototype model – Develop the prototype of the product.

**The SRS and prototype model should be submitted for end semester examination.**

**List of Sample Experiments:**

**1. Student Enrolment System.**

A University has contracted you to develop their new student records system.

The normal tasks that the system performs are as follows:

- Enrol a student at the university: A student provides his or her personal details (name, address, sex, date of birth), along with the code of the course (e.g. Bachelor of Computer Science) in which he or she wishes to enrol. A student record is created, and a unique student ID number is assigned to the student. The system automatically enrolls the student in any core first-year subjects for the course.
- Enrol a student in a subject: A student provides his or her student ID number and the

subject code of the subject in which he or she wish to enrol. The system checks that the subject requested by the student is allowed for the course in which the student is enrolled. If not, the enrolment request is rejected. The system checks what subjects (if any) are specified as prerequisites for the subject in which the student wishes to enrol. If the student has passed all the prerequisite subjects, he or she is enrolled in the desired subject. Otherwise, the enrolment request is rejected.

- Record a mark for a student: A staff member accesses the system by giving a subject code and a password for that subject. If the password is correct, the system displays the list of students enrolled in the subject to the staff member. The staff member can then specify a mark for any student on the list.

- Create a new subject: An administrator accesses the system using a password. The administrator then chooses a subject code for the new subject. The system checks that this code is not already in use in the system, and if not, creates a new subject record. The administrator then gives the subject name, the course to which it belongs, the year of the course in which it may first be taken, a flag indicating whether or not it is a core subject and the codes of any prerequisite subjects.

- Print a transcript of a student's results: An administrator accesses the system using a password. The administrator then gives the student ID number of the student for whom the transcript is to be generated. The system contacts the finance system to check whether or not the student has paid all fees. If fees have been paid, the system creates a transcript showing all the subjects in which the student has been enrolled in each year, and the mark for that subject. The header of the transcript shows the student's personal details and the course in which he or she is enrolled.

- Assign a staff member to a subject: An administrator accesses the system using a

password. The administrator then gives the subject code for the subject to which the staff member is to be assigned, and the staff ID number of the staff member.

## **2. Online Bookshop.**

A major book retailer is planning to develop a computer system to handle their new online bookshop: Booky.com. You have been chosen to do the analysis and design. The following requirements have been identified:

- Customers can search for books on the Booky.com website, either by author name, or words in the title. A list of all matching books is returned to the customer. A customer does not need to be logged-in in order to search.

- The system records all the customers of the Booky.com who have ever logged in. A customer may be an individual customer or a business customer.

- Each customer has a username and password. Business customers may have several usernames and passwords, corresponding to different divisions within the business.

- When a customer has selected a book to buy at the Booky.com website. The system prompts for the customer's username and password. The customer enters these details. The system verifies the customer's identity and retrieves the customer's name and address, then prompts for credit card details. The customer enters these details. The system checks the credit card details. The system shows the customer the book and delivery price. The customer confirms the transaction.

- The system records all books available at Booky.com. For each book, the author, title and ISBN number are recorded. The number of each book in stock is also stored, along with the number on

order by customers and the number on order from publishers. Books may be temporarily unavailable.

- All books are stored in the Booky.com warehouse. The warehouse can be contacted via a secure internet connection.
- For each customer, a permanent record of books bought by that customer is maintained. Likewise, for each book, a record of customers who have bought that book is kept.
- A customer order consists of one or more order lines, each corresponding to a particular book. A customer may choose to defer the shipment of an order until all the order lines have been filled.
- When the warehouse fills all or part of customer order, an email is sent to the customer informing them of what has been shipped.
- If a book ordered by a customer turns out to be unavailable, the corresponding order line is flagged and an email is sent to the customer informing them of the problem. At this stage the customer can cancel this order line.
- When a book corresponding to a previously-unavailable order line becomes available, an email is sent to the customer and a copy of the book is held for seven days, after which it is returned to normal stock if the customer has not confirmed the order.
- The shop keeps track of which publishers produce particular book titles. Some books may be available from more than one publisher.
- Although Booky.com will initially sell only books, it is envisaged that in future it will offer further products, such as CDs. The list of possible future products has not yet been finalized.

### 3. **Course management system (CMS)**

A **course management system (CMS)** is a collection of software tools providing an online environment for course interactions. A CMS typically includes a variety of online tools and environments, such as:

- An area for faculty posting of class materials such as course syllabus and handouts
- An area for student posting of papers and other assignments
- A gradebook where faculty can record grades and each student can view his or her grades
- An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class
- A chat tool allowing synchronous communication among class participants
- A threaded discussion board allowing asynchronous communication among participants

In addition, a CMS is typically integrated with other databases in the university so that students enrolled in a particular course are automatically registered in the CMS as participants in that course.

The Course Management System (CMS) is a web application for department personnel, Academic Senate, and Registrar staff to view, enter, and manage course information formerly submitted via paper.

Departments can use CMS to create new course proposals, submit changes for existing courses, and track the progress of proposals as they move through the stages of online approval.

### 4. **Easy Leave**

This project is aimed at developing a web based Leave Management Tool, which is of importance to either an organization or a college.

The **Easy Leave** is an Intranet based application that can be accessed throughout the organization or a specified group/Dept. This system can be used to automate the workflow of leave applications and their approvals. The periodic crediting of leave is also automated. There are features like notifications, cancellation of leave, automatic approval of leave, report generators etc in this Tool.

**Functional components of the project:**

There are registered people in the system. Some are approvers. An approver can also be a requestor. In an organization, the hierarchy could be Engineers/Managers/Business Managers/Managing Director etc. In a college, it could be Lecturer/Professor/Head of the Department/Dean/Principal etc.

Following is a list of functionalities of the system:

1. A person should be able to
  - o login to the system through the first page of the application
  - o change the password after logging into the system
  - o see his/her eligibility details (like how many days of leave he/she is eligible for etc)
  - o query the leave balance
  - o see his/her leave history since the time he/she joined the company/college
  - o apply for leave, specifying the from and to dates, reason for taking leave, address for communication while on leave and his/her superior's email id
  - o see his/her current leave applications and the leave applications that are submitted to him/her for approval or cancellation
  - o approve/reject the leave applications that are submitted to him/her
  - o withdraw his/her leave application (which has not been approved yet)
  - o Cancel his/her leave (which has been already approved). This will need to be approved by his/her Superior
  - o get help about the leave system on how to use the different features of the system
2. As soon as a leave application /cancellation request /withdrawal /approval /rejection /password-change is made by the person, an automatic email should be sent to the person and his superior giving details about the action
3. The number of days of leave (as per the assumed leave policy) should be automatically credited to everybody and a notification regarding the same be sent to them automatically

An automatic leave-approval facility for leave applications which are older than 2 weeks should be there. Notification about the automatic leave approval should be sent to the person as well as his superior

**5. E-Bidding**

Auctions are among the latest economic institutions in place. They have been used since antiquity to sell a wide variety of goods, and their basic form has remained unchanged. In this dissertation, we explore the efficiency of common auctions when values are interdependent- the value to a particular bidder may depend on information available only to others-and asymmetric. In this setting, it is well known that sealed-bid auctions do not achieve efficient allocations in general since they do not allow the information held by different bidders to be shared.

Typically, in an auction, say of the kind used to sell art, the auctioneer sets a relatively low initial price. This price is then increased until only one bidder is willing to buy the object, and the exact manner in which this is done varies. In my model a bidder who drops out at some price can "reenter" at a higher price.

With the invention of E-commerce technologies over the Internet the opportunity to bid from the comfort of ones own home has seen a change like never seen before. Within the span of a few short years, what may have began as an experimental idea has grown to an immensely popular hobby, and in some cases, a means of livelihood, the Auction Patrol gathers tremendous response everyday, all day. With the point and click of the mouse, one may bid on an item they may need or just want, and in moments they find that either they are the top bidder or someone else wants it more, and you're outbid! The excitement of an auction all from the comfort of home is a completely different experience.

Society cannot seem to escape the criminal element in the physical world, and so it is the same with Auction Patrols. This is one area wherein a question can be raised as to how safe Auction Patrols.

Proposed system

1. To generate the quick reports
2. To make accuracy and efficient calculations
3. To provide proper information briefly
4. To provide data security
5. To provide huge maintenance of records
6. Flexibility of transactions can be completed in time

## **6. Electronic Cash counter**

This project is mainly developed for the Account Division of a Banking sector to provide better interface of the entire banking transactions. This system is aimed to give a better out look to the user interfaces and to implement all the banking transactions like:

- Supply of Account Information
- New Account Creations
- Deposits
- Withdraws
- Cheque book issues
- Stop payments
- Transfer of accounts
- Report Generations.

Proposed System:

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- Readily upload the latest updates, allows user to download the alerts by clicking the URL.
- There is no risk of data mismanagement at any level while the project development is under process.

It provides high level of security with different level of authentication

## **7. Enterprise Security Services**

Verification and Validation is a part of S/W Quality Assurance. Verification refers to the set of activities that ensure correctly implements a specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

Verification: "Are we building the product right"

Validation: "Are we building the right product"



The project entitled Independent Project Metrics is an effort, to develop a tool to manage the Verification and Validation process.

The specific purpose of the Independent Verification and Validation Process of Project Metrics Tool is to bring out the various Verification and validation tasks to be performed. The scope of the Project Metrics is to cover the developed for system.

The goals of the V&V effort is to ensure that the software and the documents are developed are of high quality as expected from any mission critical software. This project generates the plan for Verification and validation process. This project maintain the document names, source code module names, version number, released date, receiving date size of document and source code modules of receiving projects for Verification and validation.

Using this application we assign the tasks/activities to different persons and also calculate the expected efforts and actual efforts. The V&V co-coordinator does this work.

Proposed System:

The general description gives an "executive overview" and is very client-oriented. It expounds on the functional and data requirements of the application. It also lists the limitations, assumptions and dependencies of the application. It also touches on the performance and quality requirements of the application and provides a solid definition of the interface

The computerization of this system would avoid the wrong interpretation and bad calculation of data .The system help the user to see any documents, source code, tasks, activities, team information with details at the click of a button. The record data is maintained and backed up such a way that data is not loss. The speed of the system could also increased

### **8.ERP**

ERP is a powerful human resource tool for maintaining employee and company information. More than a data storage program, ERP helps you manage your employees. ERP offers a wide variety of reports that give you exactly the information you need. View payroll information by department, or find everyone who is receiving company

Module Description:

1. Payroll
2. Employee
3. Employee payslip
4. Selection process
5. Reports
6. Mailing System
7. Training
8. Add Company Information

### **PROPOSED SYSTEM**

The proposed system is designed to eliminate all the drawbacks of the existing system. The system is part of a large HRMS Application and shall be responsible for maintaining information about employees,

- positions,
- company benefits,
- departments,
- new recruit checklists,
- employee achievements,
- warnings,
- evaluation reports,
- education & training,

- administration,
- work changes and several ad hoc reports.

The major advantage of the proposed system is,

- It's online, so that information is available anytime.
- High integrity and security.
- Ability to incorporate newly available data.
- It is user friendly
- Speed and accuracy is increased
- Fully automated.
- Security is associated with user authentication
- Duplication of information is curbed

### **8.Examination Branch System**

The project "**Examination Branch System** " is developed to reduce the overhead involved in the process of maintains the data and the transaction in the Examination branch . Examination branch is an intranet application for an organization. It is software which is used to perform all the examination activities like adding employees, search employees, delete employees and assign examination duties etc.

The basic framework of the project is developed in .NET. Making use of this application the administrator can perform their activities through it.

Proposed System:

- Now we can extend our project to assign duties to faculty.  
We can implement edit, update operations now.  
We can develop our project as a user friendly type

### **9.Exam Experts**

The system would be providing a number of services, automating the processes that are being done manually. The services include communication services such as mailing facility, chat service, electronic file transfer etc and office automation packages such as leave letter processing, admission management, teaching evaluation, counseling automation etc.

The aim of the project is to design a comprehensive web enabled application for management of the Examination Process. Examination system is categorized into various sections. Among those sections, this system concentrates on the work being done in section (E-X).

The section (E-X) deals with the confidential work, i.e., Coding-Decoding of answer scripts, Processing of results, Computerization of certificates etc. This is an automated section and it plays a pivotal role in the Examination Process starting from the Application Processing to the final announcement of results

This project is aimed to solve many of the problems that are in the existing system and also provide a hassle free system that is efficient and easy to use. This project concentrates mainly on Application Processing, Marks Processing and Results Processing with an easy to use interface. The system also provides a means to generate and print various types of reports.

### **10.Application Processing System:**

This phase involves the storing of the application information and generating the required reports.

- Entry of Application forms according to center ,course order and batch
- Generating Application Id for further transactions
- Capturing of photographs of students for hall ticket processing
- Reports involving the information about students who are appearing for supplementary exams
- Generating nominal roles Reports describing the college, course, subjects and the students appearing

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1L31**

**COMPUTER NETWORKING LAB  
(PROFESSIONAL ELECTIVE-I LAB)  
(LAB-III)**

**Sample Problems on Networks (Use C/JAVA 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

**Sample Problems on Network Simulation (Use Simulation tools like NetSim/NS2, etc ):**

1. Installation of simulation tools in Linux/Windows environment
2. Introduction about discrete events simulation and its tools like NetSim, NS2, etc
3. Understand IP forwarding within a LAN and across a router
4. Study the working of spanning tree algorithm by varying the priority among the switches
5. Study the throughputs of Slow start + Congestion avoidance(Old Tahoe) and Fast Retransmit (Tahoe) Congestion Control Algorithms
6. Study how the Data Rate of Wireless LAN(IEEE 802.11b) Network varies as the distance between the Access Point and the wireless nodes is varied
7. Write a script to connect two nodes
8. Write a script for connecting three nodes considering one node as a central node.
9. Write a script to implement star topology
10. Write a script to implement a bus topology.
11. Study the working and routing table formation of Interior routing table formation of interior routing protocols , i.e. Routing Information Protocol (RIP) and Open Shortest Path First(OSPF)
12. Analyze the performance of a MANET,(running CSMA/CA(802.11b) in MAC)with increasing node density
13. Analyze the performance of a MANET,(running CSMA/CA(802.11b) in MAC)with increasing node mobility
14. Bit Stuffing and character stuffing
15. Cyclic Redundancy Check
16. Encryption and Decryption using substitution ciphers
17. Distance vector Routing
18. Link State Routing
19. Go Back N Protocol

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1L32**

**INTERNET TECHNOLOGIES AND SERVICES LAB  
(PROFESSIONAL ELECTIVE-I LAB)  
(LAB-III)**

**Objectives:**

- Write syntactically correct HTTP messages and describe the semantics of common HTTP methods and header fields
- Discuss differences between URIs, URNs, and URLs, and demonstrate a detailed understanding of http-scheme URLs, both relative and absolute
- Describe the actions, including those related to the cache, performed by a browser in the process of visiting a Web address
- Install a web server and perform basic administrative procedures, such as tuning communication parameters, denying access to certain domains, and interpreting an access log
- Write a valid standards-conformant HTML document involving a variety of element types, including hyperlinks, images, lists, tables, and forms
- Use CSS to implement a variety of presentation effects in HTML and XML documents, including explicit positioning of elements
- Demonstrate techniques for improving the accessibility of an HTML document

**List of Sample Problems:**

**i) Internet Technologies**

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. 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.
5. 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.
6. Implement the “Hello World!” program using JSP Struts Framework.

## **ii)Additional Assignment Problems**

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.

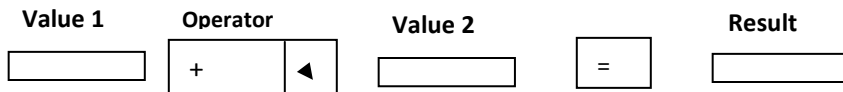
Write a simple calculator servlet that takes two numbers and an operator (+, -, /, \* and %) from an HTML page and returns the result page with the operation performed on the operands. It should check in a database if the same expression is already computed and if so, just return the value from database. Use MySQL or PostgreSQL.

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).

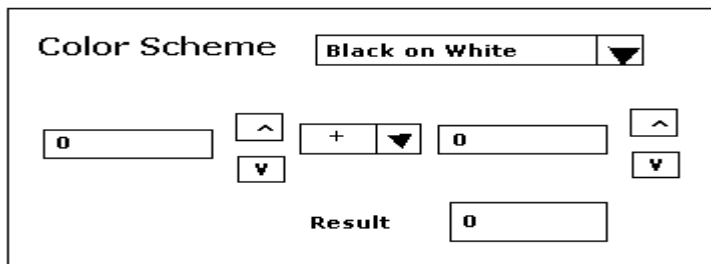
Write a servlet that takes name and age from an HTML page. If the age is less than 18, it

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.

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:



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



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.

<b>Field</b>	mobile	
<b>Value</b>	9449449449	OK
<b>Result</b>	abc, 22, Hyd def, 23, Delhi xxx, 44, Chennai	

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 in:

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

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).

### iii)Internet Technologies and Services Lab - Additional Problems

Create a web Service in Java that takes two city names from the user and returns the distance between these two from data available from a table in MySql.

Write a java and a C# client which use the above service

Write a Java program that takes a file as input and encrypts it using DES encryption. The program should check if the file exists and its size is not zero.

Write a Java program that generates a key pair and encrypts a given file using RSA algorithm.

Write a Java program that finds digest value of a given string.

Consider the following xml file for encryption

```
<?xml version="1.0"> <transaction> <from>12345</from> <to>54321</to>
<amount>10000</amount>
<secretcode>abc123</secretcode> <checksum></checksum> </transaction>
```

Replace <from> and <to> values with the RSA encrypted values represented with base64 encoding assuming that the public key is available in a file in local directory "pubkey.dat". Encrypt <secretcode> with AES algorithm with a password 'secret'. The checksum of all the field values concatenated with a delimiter character '+' will be inserted in the checksum and the xml file is written to encrypted.xml file.

Assume that a file 'config.xml', which has the following information:

```
<users>
<user> <name>abc</name> <pwd>pwd123</pwd> <role>admin</role> <md5>xxx</md5> </user>
<user> <name>def</name> <pwd>pwd123</pwd> <role>guest</role> <md5>xxx</md5> </user>
</users>
```

Replace name and role with DES encrypted values and pwd with RSA encrypted values (represent the values with base64 encoding). The public key is available in "public.key" file in current directory. Replace xxx with respective MD5 values of all the fields for each user. Write the resulting file back to config.xml.

Write an HTML page that gives 3 multiple choice (a,b,c and d) questions from a set of 5 preloaded questions randomly. After each question is answered change the color of the question to either green or blue using CSS. Finally on clicking OK button that is provided, the score should be displayed as a pop-up window. Use Java Script for dynamic content.

Write an HTML page that has 3 countries on the left side ("USA", "UK" and "INDIA") and on the right side of each country, there is a pull-down menu that contains the following entries: ("Select Answer", "New Delhi", "Washington" and "London"). The user will match the Countries with their respective capitals by selecting an item from the menu. The user chooses all the three answers (whether right or wrong). Then colors of the countries should be changed either to green or to red depending on the answer. Use CSS for changing color.

Write an HTML Page that can be used for registering the candidates for an entrance test. The fields are: name, age, qualifying examination (diploma or 10+2), stream in qualifying examination. If qualifying examination is "diploma", the stream can be "Electrical", "Mechanical" or "Civil". If the qualifying examination is 10+2, the stream can be "MPC" or "BPC". Validate the name to accept only



characters and spaces.

Write an HTML page that has two selection menus. The first menu contains the states (“AP”, “TN” and “KN”) and depending on the selection the second menu should show the following items: “Hyderabad”, “Vijayawada”, “Kurnool” for AP, “Chennai”, “Salem”, “Madurai” for TN and “Bangalore”, “Bellary”, “Mysore” for KN.

Write an HTML page that has phone buttons 0 to 9 and a text box that shows the dialed number. If 00 is pressed at the beginning, it should be replaced with a + symbol in the text box. If the number is not a valid international number (+ followed by country code and 10 digit phone number) the color of the display should be red and it should turn to green when the number is valid. Consider only “+91, +1 and +44 as valid country codes. Use CSS for defining colors.

Write an HTML page that has a text box for phone number or Name. If a number is entered in the box the name should be displayed next to the number. If 00 is pressed at the beginning, it should be replaced with a + symbol in the text box. If a name is entered in the text box, it should show the number next to the name. If the corresponding value is not found, show it in red and show it in green otherwise. Use CSS for colors. Store at least 5 names and numbers in the script for testing.

A library consists of 10 titles and each title has a given number of books initially. A student can take or return a book by entering his/her HTNo as user ID and a given password. If there are at least two books, the book is issued and the balance is modified accordingly.

- (a) Use RDBMS and implement it with JSP.
- (b) Use XML File for data and Implement it with JSP
- (c) Use RDBMS and implement it with Servlets
- (d) Use XML File for data and Implement it with Servlets

A Bus Reservation System contains the details of a bus seat plan for 40 seats in 2x2 per row arrangement, where the seats are numbered from 1 to 40 from first row to last row. The customer can visit the website and can reserve a ticket of his choice if available by entering his details (Name, Address, Gender and Age). The customer can cancel the ticket by entering the seat number and his name as entered for reservation.

- (a) Use RDBMS and implement it with JSP.
- (b) Use XML File for data and Implement it with JSP
- (c) Use RDBMS and implement it with Servlets
- (d) Use XML File for data and Implement it with Servlets.

Implement a simple messaging system with the following details:

When a student logs in with his/her HTNO and a given password, they should get all the messages posted to him/her giving the ID of sender and the actual message. Each message may be separated with a ruler. There should be a provision for the user to send a message to any number of users by giving the IDs separated with commas in the “To” text box.

- (a) Use RDBMS and implement it with JSP.
- (b) Use XML File for data and Implement it with JSP
- (c) Use RDBMS and implement it with Servlets
- (d) Use XML File for data and Implement it with Servlets.

There is an image of 600x100 size which can be logically divided into 12 button areas with labels (0-9, +, =). Write a javascript calculator program that uses this image as input virtual keyboard and three text areas for two input numbers and result of sum of these numbers. Add a CSS that can be used to change the colors of text and background of text areas and the page. The input numbers can be up to 4 digits each.

Develop a web application that takes user name and password as input and compares them with those available in an xml user database. If they match, it should display the welcome page that contains the user’s full name and last used date and time retrieved from a client cookie. On logout it stores new time to the cookie and displays a goodbye page. If authentication fails, it should store the attempt number to the client cookie and displays an error page. Add necessary CSS that takes care of the font, color of foreground and background.

A web application has the following specifications:

The first page (Login page) should have a login screen where the user gives the login name and password. Both fields must be validated on client side for a minimum length of 4 characters, name should be lower case a-z characters only and password should contain at least one digit. On submitting these values, the server should validate them with a MySQL database and if failed, show the login page along with a message saying “Login Name or Password Mismatch” in Red color below the main heading and above the form. If successful, show a welcome page with the user's full name (taken from database) and and a link to Logout. On logout, a good bye page is displayed with the total time of usage (Logout time – login time). Specify the Schema details of table and web.xml file contents.

Implement it using (a) JSP Pages (b) Servlets (c) Struts

Design a struts based web portal for an international conference with following specifications:

The welcome page should give the details of the conference and a link to login. If login fails, direct them back for re-login and also provide a link for registration. On successful registration/login, the

user will be directed to a page where s/he can see the status (accepted/rejected) of their already submitted papers followed by a form for submitting a doc file to the conference. Provide a logout button on all pages including the home page, once the user logs in. Implement validation framework to check that the user name is in the form of CCDDCC and password is in the form of (CCSDDD) (C for character, S for special character (one of @, #, \$, %, ^, & and !) and D for digit)., Database should be accessed through Connection Pool for MySql for user information. Provide scope for internationalization in future. Assume any missing information and mention it first.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1L33**

**ANDROID APPLICATION DEVELOPMENT LAB.  
(PROFESSIONAL ELECTIVE-I LAB)  
(LAB-III)**

**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.

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.

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.

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.

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.

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.

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.

Create a user registration application that stores the user details in a database table.

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.

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.

Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.

Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.

Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

Create an application that shows the given URL (from a text field) in a browser.

Develop an application that shows the current location's latitude and longitude continuously as the device is moving (tracking).

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)

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1L34**

**DATA MINING LAB  
(PROFESSIONAL ELECTIVE-I LAB)  
(LAB – III)**

**List of Sample Problems:**

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, Uinit\_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.

**Similar Tasks Can Be Framed**

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR I SEMESTER**

Subject Code: **SE1A10**

**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

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2C10**

**SOFTWARE ARCHITECTURE AND DESIGN PATTERNS  
(CORE – IV)**

**Objectives:**

After completing this course, the student should be able to:

- To understand the concept of patterns and the Catalog.
- To discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
- To understand the variety of implemented bad practices related to the Business and Integration tiers.
- To highlight the evolution of patterns.
- To learn how to add functionality to designs while minimizing complexity
- To learn what design patterns really are, and are not
- To know about specific design patterns.
- To learn how to use design patterns to keep code quality high without over design.

**UNIT I**

**Envisioning Architecture:** The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views. **Creating an Architecture:** Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

**UNIT II**

**Analyzing Architectures:** Architecture Evaluation, Architecture design decision making, ATAM, CBAM. **Moving from one system to many:** Software Product Lines, Building systems from off the shelf components, Software architecture in future.

**UNIT III**

**Patterns:** Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage. **Creational and Structural patterns:** Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight.

**UNIT IV**

**Behavioural patterns:** Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy. template method, visitor.

## **UNIT V**

**Case Studies:** A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development,

### **TEXT BOOKS:**

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick Kazman, Pearson Education, 2003.
2. Design Patterns, Erich Gamma, Pearson Education.

### **REFERENCES:**

1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
5. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
6. J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003.
7. Design Patterns in C#, Steven John metsker, Pearson education, 2004.
8. Pattern Oriented Software Architecture, F.Buschmann &others, John Wiley & Sons.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2C20**

**SOFTWARE QUALITY ASSURANCE AND TESTING  
(CORE – V)**

**Objectives:**

The student should be able to:

- To understand software testing and quality assurance as a fundamental component of software life cycle
- To define the scope of SW T&QA projects
- To efficiently perform T&QA activities using modern software tools
- To estimate cost of a T&QA project and manage budgets
- To prepare test plans and schedules for a T&QA project
- To develop T&QA project staffing requirements
- To effectively manage a T&QA project

**UNIT I**

**Software Quality Assurance and Standards:** The Software Quality challenge, What is Software Quality, Software Quality factors, The components of Software Quality Assurance system, Software Quality Metrics, Costs of Software Quality, Quality Management Standards, Management and its role in Software Quality Assurance, SQA unit and other actors in SQA system. - **(Chapters: 1-4, 21-23, 25, 26) of T3 Quality Standards: ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma and other latest quality standards (Refer Internet and R11, R12, R13).**

**UNIT II**

**Software Testing Strategy and Environment:** Minimizing Risks, Writing a Policy for Software Testing, Economics of Testing, Testing-an organizational issue, Management Support for Software Testing, Building a Structured Approach to Software Testing, Developing a Test Strategy **Building Software Testing Process:** Software Testing Guidelines, workbench concept, Customizing the Software Testing Process, Process Preparation checklist - **(Chapters: 2,3) of T1** **Software Testing Techniques:** Dynamic Testing – Black Box testing techniques, White Box testing techniques, Static testing, Validation Activities, Regression testing -**(Chapters: 4, 5, 6, 7, 8) of T2**

**UNIT III**

**Software Testing Tools:** Selecting and Installing Software Testing tools – **(Chapter 4) of T1.** Automation and Testing Tools - **(Chapter 15) of T2** Load Runner, Win runner and Rational Testing Tools, Silk test, Java Testing Tools, JMetra, JUNIT and Cactus. **(Refer Internet and R9, R10)**



## UNIT IV

**Testing Process Seven Step Testing Process – I:** Overview of the Software Testing Process, Organizing of Testing, Developing the Test Plan, Verification Testing, Validation Testing. (Chapters 6, 7, 8, 9, 10) of T1

## UNIT V

**Seven Step Testing Process – II:** Analyzing and Reporting Test results, Acceptance and Operational Testing, Post-Implementation Analysis **Specialized Testing Responsibilities:** Software Development Methodologies, Testing Client/Server Systems (Chapters 12, 13, 14, 15) of T1.

### TEXT BOOKS:

1. Effective Methods for Software Testing, Third edition, *William E. Perry*, Wiley India, 2009
2. Software Testing – Principles and Practices, *Naresh Chauhan*, Oxford University Press, 2010.
3. Software Quality Assurance – From Theory to Implementation, *Daniel Galin*, Pearson Education, 2009.

### REFERENCES:

1. Testing Computer Software, Cem Kaner, Jack Falk, Hung Quoc Nguyen, Wiley India, rp2012.
2. Software Testing – Principles, Techniques and Tools, *M.G.Limaye*, Tata McGraw-Hill, 2009.
3. Software Testing - A Craftsman's approach, *Paul C. Jorgensen*, Third edition, Auerbach Publications, 2010.
4. Foundations of Software Testing, *Aditya P. Mathur*, Pearson Education, 2008.
5. Software Testing and Quality Assurance – Theory and Practice, *Kshirasagar Naik, Priyadashi Tripathy*, Wiley India, 2010.
6. Software Testing, *Ron Patton*, Second edition, Pearson Education, 2006.
7. Software Testing and Analysis – Process, Principles and Techniques, *Mauro Pezze, Michal Young*, Wiley India, 2008.
7. Software Testing Techniques, Boris Beizer, Second edition, Wiley India, 2006
8. Foundations of Software Testing, Dorothy Graham, et al., Cengage learning, 2007, rp 2010.
9. Software Testing - Effective Methods, Tools and Techniques, *Renu Rajani, Pradeep Oak*, Tata McGraw-Hill, rp2011.
10. Software Automation Testing Tools for Beginners, *Rahul Shende*, Shroff Publishers and Distributors, 2012.
11. Software Testing Tools, *K.V.K.K. Prasad*, Dream Tech Press, 2008.
12. Software Testing Concepts and Tools, *Nageswara Rao Pusuluri*, Dream Tech press, 2007.
13. Software Quality Assurance, *Milind Limaye*, Tata McGraw-Hill, 2011.
14. Software Quality – Theory and Management, *Alan C. Gillies*, Second edition, Cengage

- Learning, 2009.
15. Software Quality – A Practitioner’s approach, *Kamna Malik, Praveen Choudhary*, Tata McGraw-Hill, 2008.
  16. Software Quality Models and Project Management in a Nutshell, *Shailesh Mehta*, Shroff Publishers and Distributors, 2010.
  17. Software Quality Engineering – Testing, Quality Assurance and Quantifiable Improvement, *Jeff Tian*, Wiley India, 2006.
  18. Software Quality, *Mordechai Ben-Menachem/Garry S. Marliss*, Cengage Learning, 2010.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P31**

**NETWORK SECURITY  
(PROFESSIONAL ELECTIVE - III)**

**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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P32**

**DATA ANALYTICS  
(PROFESSIONAL ELECTIVE-III)**

**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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P33**

**INTERNET OF THINGS  
(PROFESSIONAL ELECTIVE - III)**

**Objectives:**

- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- 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**

IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCOZF, YANG- NETCONF, YANG, SNMP NETOPEER

**UNIT III**

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 IV**

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

**UNIT V**

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

**TEXT BOOKS:**

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



**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P34**

**COMPUTER FORENSICS  
(PROFESSIONAL ELECTIVE-III)**

**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, Stuart, 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

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P41**

**SOFTWARE PROCESS AND PROJECT MANAGEMENT  
(PROFESSIONAL ELECTIVE – IV)**

**Objectives:**

At the end of the course, the student shall be able to:

- To describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
- To compare and differentiate organization structures and project structures.
- To implement a project to manage project schedule, expenses and resources with the application of suitable project management tools.

**UNIT I**

**Software Process Maturity** : Software maturity Framework, Principles of Software Process Change, Software Process Assessment, The Initial Process, The Repeatable Process, The Defined Process, The Managed Process, The Optimizing Process. **Process Reference Models** Capability Maturity Model (CMM), CMMi, PCMM, PSP, TSP.

**UNIT II**

**Software Project Management Renaissance** Conventional Software Management, Evolution of Software Economics, Improving Software Economics, The old way and the new way. **Life-Cycle Phases and Process artifacts** Engineering and Production stages, inception phase, elaboration phase, construction phase, transition phase, artifact sets, management artifacts, engineering artifacts and pragmatic artifacts, model based software architectures.

**UNIT III**

**Workflows and Checkpoints of process** Software process workflows, Iteration workflows, Major milestones, Minor milestones, Periodic status assessments. **Process Planning** Work breakdown structures, Planning guidelines, cost and schedule estimating process, iteration planning process, Pragmatic planning.

**UNIT IV**

**Project Organizations** Line-of- business organizations, project organizations, evolution of organizations, process automation. **Project Control and process instrumentation** The seven core metrics, management indicators, quality indicators, life-cycle expectations, Pragmatic software metrics, metrics automation.

## UNIT V

**CCPDS-R Case Study and Future Software Project Management Practices** Modern Project Profiles, Next-Generation software Economics, Modern Process Transitions.

### TEXT BOOKS:

1. Managing the Software Process, *Watts S. Humphrey*, Pearson Education.
2. Software Project Management, *Walker Royce*, Pearson Education.

### REFERENCES:

1. Effective Project Management: Traditional, Agile, Extreme, Robert Wysocki, Sixth edition, Wiley India, rp2011.
2. An Introduction to the Team Software Process, Watts S. Humphrey, Pearson Education, 2000
3. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, TMH, 2006
4. Applied Software Project Management, Andrew Stellman & Jennifer Greene, O'Reilly, 2006.
5. Head First PMP, Jennifer Greene & Andrew Stellman, O'Reilly, 2007
6. Software Engineering Project Management, Richard H. Thayer & Edward Yourdon, 2<sup>nd</sup> edition, Wiley India, 2004.
7. The Art of Project Management, Scott Berkun, SPD, O'Reilly, 2011.
8. Applied Software Project Management, Andrew Stellman & Jennifer Greene, SPD, O'Reilly, rp2011.
9. Agile Project Management, Jim Highsmith, Pearson education, 2004.
10. Process Improvement essentials, James R. Perse, O'Reilly, 2006

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P42**

**COMPONENT BASED SOFTWARE ENGINEERING  
(PROFESSIONAL ELECTIVE – IV)**

**Objectives:**

- To understand the essentials of component-based software engineering
- To know the main characteristics of components and component models
- To be aware of software development processes for component-based systems
- To be aware of the mutual relations between software architecture and component models

**UNIT I**

Component definition - Definition of a Software Component and its elements, The Component Industry Metaphor, Component Models and Component Services, An example specification for implementing a temperature regulator Software Component. The Case for Components- The Business Case for components, COTS Myths and Other Lessons Learned in Component-Based Software Development.

**UNIT II**

Planning Team Roles for CBD, Common High-Risk Mistakes, CBSE Success Factors: Integrating Architecture, Process, and Organization. Software Engineering Practices - Practices of Software Engineering, From Subroutines to Subsystems: Component-Based Software Development, Status of CBSE in Europe.

**UNIT III**

The Design of Software Component Infrastructures - Software Components and the UML, Component Infrastructures, Business Components, Components and Connectors, An OPEN process for CBD, Designing Models of Modularity and Integration. Software Architecture, Software Architecture Design Principles, Product-Line Architectures.

**UNIT IV**

The Management of Component-Based Software Systems - Measurement and Metrics for Software Components, Implementing a Practical Reuse Program for Software Components, Selecting the Right COTS Software, Building instead of Buying, Software Component Project Management, The Trouble with Testing Components, Configuration Management and Component Libraries, The Evolution, Maintenance, and Management of CBS.

**UNIT V**

Component Technologies - Overview of the CORBA Component Model, Overview of COM+, Overview of the EJB Component Model, Bonobo and Free Software GNOME Components,

Choosing between COM+, EJB, and CCM, Software Agents as Next Generation Software Components.

**TEXT BOOKS:**

1. Component - Based Software Engineering, G.T. Heineman and W.T. Council, Addison- Wesley, Pearson Education.

**REFERENCES:**

1. Component Software, C.Szyperski, D.Gruntz and S.Murer, Pearson Education.
2. Software Engineering, Roger S. Pressman, 6<sup>th</sup> edition, Tata McGraw-Hill.
3. Software Engineering, Ian Sommerville, seventh edition, Pearson education, 2004.
4. Software Engineering Principles and Practice, Hans Van Vliet, 3<sup>rd</sup> edition, Wiley India.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P43**

**SOFTWARE SECURITY ENGINEERING  
(PROFESSIONAL ELECTIVE – IV)**

**Objectives:**

- Students will demonstrate knowledge of the distinction between critical and non-critical systems.
- Students will demonstrate the ability to manage a project including planning, scheduling and risk assessment/management.
- Students will author a software requirements document.
- Students will demonstrate an understanding of the proper contents of a software requirements document.
- Students will author a formal specification for a software system.
- Students will demonstrate an understanding of distributed system architectures and application architectures.
- Students will demonstrate an understanding of the differences between real-time and non-real time systems.
- Students will demonstrate proficiency in rapid software development techniques.
- Students will be able to identify specific components of a software design that can be targeted for reuse.
- Students will demonstrate proficiency in software development cost estimation.
- Students will author a software testing plan.

**UNIT – I**

**Security a software Issue:** introduction, the problem, Software Assurance and Software Security, Threats to software security, Sources of software insecurity, Benefits of Detecting Software Security

**What Makes Software Secure:** Properties of Secure Software, Influencing the security properties of software, Asserting and specifying the desired security properties?

**UNIT – II**

**Requirements Engineering for secure software:** Introduction, the SQUARE process Model, Requirements elicitation and prioritization

**UNIT – III**

**Secure Software Architecture and Design:** Introduction, software security practices for architecture and design: architectural risk analysis, software security knowledge for architecture and design: security principles, security guidelines and attack patterns

**Secure coding and Testing:** Code analysis, Software Security testing, Security testing considerations throughout the SDLC

**UNIT – IV**

**Security and Complexity:** System Assembly Challenges: introduction, security failures, functional and attacker perspectives for security analysis, system complexity drivers and security

**UNIT – V**

**Governance and Managing for More Secure Software:** Governance and security, Adopting an enterprise software security framework, How much security is enough?, Security and project management, Maturity of Practice

**TEXT BOOK:**

1. Software Security Engineering: Julia H. Allen, Pearson Education

**REFERNCES:**

1. Developing Secure Software: Jason Grembi, Cengage Learning
2. Software Security : Richard Sinn, Cengage Learning



**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2P44**

**BUSINESS PROCESS MANAGEMENT  
(PROFESSIONAL ELECTIVE – IV)**

**Objectives:**

- To recognize the role of business processes within an Infinity based application
- To understand the importance of parameter sets to a business process
- To learn common patterns and best practices for formatting and restricting the output from a business process
- To understand the difference between a business process and a business process instance
- To learn how data processing occurs within a business process
- To list the Infinity SDK software developer responsibilities for building and supporting the functionality required for a business process
- To describe the database tables used by Blackbaud Enterprise CRM to manage business processes
- To describe how a QueryViewSpec can be used to define the output format for a business process
- To describe how a selection can be used to limit the rows processed by a business process

**UNIT I**

**UNDERSTANDING BPM - I:**How can we demystify business process management?What is business process management?Why is it important to improve business process before automating them?When should you do BPM – what are the main drivers and triggers?Who should be involved in BPM?

**UNIT II**

**UNDERSTANDING BPM - II:**Why are organizational strategy and process architecture important in BPM implementation?How do you sell BPM technology to the organization?What are the critical success factors in a BPM project?What are the critical implementation aspects for a BPM solution?

**UNIT III**

**FRAMEWORK - I:**Framework overview, Guidelines on how to use the framework, Organization strategy phase, Process architecture phase, Launch pad phase, Understand phase, Innovate phase.

**UNIT IV**

**FRAMEWORK – II:**

People phase, Develop phase, Implement phase, Realize value phase, Sustainable performance phase, Essentials introduction, Project management, People change management, Leadership.

## **UNIT V**

**BPM AND THE ORGANIZATION:**BPM maturity, Embedding BPM within the organization.

### **TEXT BOOKS:**

1. Business Process Management, Practical guidelines to successful implementations, John Jeston and Johan Nelis, Second edition, Elsevier, 2009.
2. Management by Process, A roadmap to sustainable Business Process Management, John Jeston and Johan Nelis, Elsevier, 2009.

### **REFERENCES:**

1. Business Process Management Systems, Strategy and Implementation, James F. Chang, Auerbach Publications, Taylor and Francis group, 2005

## **M.TECH (SOFTWARE ENGINEERING)**

### **I YEAR II SEMESTER**

Subject Code: **SE2011 IMAGE PROCESSING AND PATTERN RECOGNITION**  
**(OPEN ELECTIVE)**

#### **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.

#### **UNIT – II**

Statistical and spatial operations, Intensity functions transformations, 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, smoothing splines and interpolation. 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. 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 –IV**

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

#### **UNIT V**

**Pattern Recognition Fundamentals:** Basic Concepts of pattern recognition, Fundamental problems in pattern recognition system, design concepts and methodologies, example of automatic pattern recognition systems, a simple automatic pattern recognition model

**Pattern classification:** Pattern classification by distance function: Measures of similarity, Clustering criteria, K-means algorithm, Pattern classification by likelihood function: Pattern classification as a Statistical decision problem, Bayes classifier for normal patterns.

#### **TEXT BOOKS :**

1. Digital Image Processing Third edition, Pearson Education, Rafael C. Gonzalez, Richard E. Woods
2. Pattern recognition Principles: Julius T. Tou, and Rafel C. Gonzalez, Addison-Wesly Publishing Company

## REFERENCES :

1. Image Processing, Analysis and Machine Vision, Second Edition, Milan Sonka, Vaclav Hlavac and Roger Boyle, Cengage learning.
2. Digital Image Processing, W.K.Pratt, 4<sup>th</sup> edition John Wiley & Sons.
3. Fundamentals of digital image processing, A.K. Jain, PHI
4. Pattern classification, Richard Duda, Hart and David Stork John Wiley publishers.
5. Digital Image Processing, S.Jayaraman, S.Esakkirajan, T.Veerakumar, TMH.
6. Pattern Recognition, R.Shinghal, Oxford University Press.
7. Digital Image Processing, S.Sridhar, Oxford University Press.

## **M.TECH (SOFTWARE ENGINEERING)**

### **I YEAR II SEMESTER**

Subject Code: **SE2012 MULTIMEDIA AND RICH INTERNET APPLICATIONS**  
**(OPEN ELECTIVE)**

#### **Objectives:**

This course aims to further develop students' competency in producing dynamic and creative graphic solutions for multimedia productions. It provides students with the basic concepts and techniques of interactive authoring. It also introduces students with the advanced scripting skills necessary for implementing highly interactive, rich internet applications using multimedia technologies and authoring tools. Students will develop aesthetic value and competencies in multimedia authoring. Artistic visual style and layout design are stressed, as well as the editing and integration of graphic images, animation, video and audio files. The course allows students to master industry-wide software and technologies to create highly interactive, rich internet applications.

#### **UNIT I**

##### **Introduction to Multimedia**

Internet and Multimedia communications, Multimedia Networks, Multimedia Applications, Multimedia Information representation- Digitization Principles, Text, Images, Audio and Video, Compression Methods-Basic Coding Methods – Run Length coding, Huffman coding, Arithmetic coding, Discrete Cosine Transform, Differential PCM, Motion Compensated Prediction, Video Compression – JPEG, H.261, MPEG-1 Video, MPEG 2 and 3 Video, H.263, Wavelet and Fractal Image Compression, Audio Compression.

#### **UNIT II**

##### **Multimedia Applications in Networks.**

Introduction, Application Level Framing, Audio/Video Conferencing-Session Directories, Audio/Video Conferencing, Adaptive Applications, Receiver Heterogeneity, Real Time Application with Resource Reservation, Video Server, Applications requiring reliable multicast – White Board, Network Text Editor for Shared Text Editing, Multi Talk, Multicast file transfer, Multimedia Applications on the World Wide Web – Multicast Web Page Sharing, Audio/Video Streams in the www, Interactive Multiplayer Games.

#### **UNIT III**

##### **Web 2.0**

What is web 2.0, Search, Content Networks, User Generated Content, Blogging, Social Networking, Social Media, Tagging, Social Marking, Rich Internet Applications, Web Services, Mashups, Location Based Services, XML, RSS, Atom, JSON, and VoIP, Web 2.0 Monetization and Business Models, Future of the Web.

## **UNIT IV**

### **Rich Internet Applications (RIAs) with Adobe Flash and Flex**

Adobe Flash- Introduction, Flash Movie Development, Learning Flash with Hands-on Examples, Publish your flash movie, Creating special effects with Flash, Creating a website splash screen, action script, web sources.

Adobe Flex 2- Introduction, Flex Platform Overview, Creating a Simple User Interface, Accessing XML data from your application, Interacting with Server Side Applications, Customizing your User Interface, Creating Charts and Graphs, Connection Independent RIAs on the desktop -Adobe Integrated Runtime (AIR), Flex 3 Beta.

## **UNIT V**

### **Ajax- Enabled Rich Internet Application**

Introduction, Traditional Web Applications Vs Ajax Applications, Rich Internet Application with Ajax, History of Ajax, Raw Ajax example using xml httprequest object, Using XML, Creating a full scale Ajax Enabled application, Dojo ToolKit.

### **TEXT BOOKS:**

1. Multimedia Communications: Protocols and Applications, Franklin F Kuo, J.Joaquin Garcia, Wolf gang Effelsberg, Prentice Hall Publications.
2. Multimedia Communications: Applications, Networks, Protocols and Standards, Fred Halsall, Addison Wesley Publications.
3. AJAX, Rich Internet Applications, and Web Development for Programmers, Paul J Deitel and Harvey M Deitel, Deitel Developer Series, Pearson education.

### **REFERENCES:**

1. Professional Adobe Flex 2, Rich Tretola, Simon barber and Renaun Erickson, Wrox, Wiley.
2. Multimedia Information Networking, Nalin K Sharda, PHI Learning.
3. Multimedia Computing, Communications & Applications, Ralf Steinmetz and Klara Nahrstedt, Pearson Education.
4. Multimedia Communication Systems: techniques, standards and networks, K.R.Rao, Bojkovic and Milovanovic, PHI Learning.
5. Programming Flex 3, C.Kazoun and J.Lott, SPD.
6. Dojo, J.E.Harmon, Pearson Education.
7. Adobe Flex 3: Training from the Source, Tapper & others, Pearson Education.
8. Principles of Multimedia, R.Parekh, TMH.
9. Mastering Dojo, R.Gill, C.Riecke and A.Russell, SPD.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2013**

**BIOINFORMATICS  
(OPEN ELECTIVE)**

**UNIT-I**

Introduction to Bioinformatics and Biological Databases, Sequence alignment, Pairwise Sequence alignment, multiple sequence alignment, database Similarities.

**UNIT-II**

Molecular phylogenetics: Basics, gene phylogene Vs Systems Phylogene, Tree construction methods and programs, advanced Statistical approaches, profiles and Hidden markow models.

**UNIT-III**

Gene and promoter prediction: Gene Prediction, promoter and regulatory element prediction, RNA structure prediction, protine motives and domain prediction

**UNIT-IV**

Structural Bioinformatics: Basics, Protine structure Visualization, comparision, classofication, protein secondary structure prediction, protein tertiary structure prediction.

**UNIT-V**

Genomics and Proteomics: Genome Mapping, Assembly, comparison, functional genomics, proteomics.

**TEXT BOOKS:**

1. Essential Bioinformatics: Jin Xiong 2006, Cambridge University Press.

**M.TECH (SOFTWARE ENGINEERING)**  
**I YEAR II SEMESTER**  
**CYBER SECURITY**  
**(OPEN ELECTIVE)**

Subject Code: **SE2014**

**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.

**UNIT-II**

**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 III**

**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.

**UNIT IV**

**Tools and Methods Used in Cybercrime: Introduction,** Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks

**UNIT V**

**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



## **TEXT BOOKS:**

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

## **REFERENCES**

1. **Cyber Security Essentials**, James Graham, Richard Howard and Ryan Otson, CRC Press.

**M.TECH (SOFTWARE ENGINEERING)**

**I YEAR II SEMESTER**

Subject Code: **SE2015 INTELLECTUAL PROPERTY RIGHTS**

**(OPEN ELECTIVE)**

**UNIT-I**

Introduction to Intellectual Property Law – The Evolutionary Past - The IPR Tool Kit- Para - Legal Tasks in Intellectual Property Law Ethical obligations in Para Legal Tasks in Intellectual Property Law - Introduction to Cyber Law – Innovations and Inventions Trade related Intellectual Property Right

**UNIT-II**

Introduction to Trade mark – Trade mark Registration Process – Post registration Procedures – Trade mark maintenance - Transfer of Rights - Inter partes Proceeding – Infringement - Dilution Ownership of Trade mark – Likelihood of confusion - Trademarks claims – Trademarks Litigations – International Trade mark Law

**UNIT-III**

Introduction to Copyrights – Principles of Copyright Principles -The subjects Matter of Copy right – The Rights Afforded by Copyright Law – Copy right Ownership, Transfer and duration – Right to prepare Derivative works – Rights of Distribution – Rights of Perform the work Publicity Copyright Formalities and Registrations - Limitations - Copyright disputes and International Copyright Law – Semiconductor Chip Protection Act

**UNIT -IV**

The law of patents-patent searches –Patent ownership and transfer-Patent infringement- International Patent Law

**UNIT-V**

Introduction to Trade Secret – Maintaining Trade Secret – Physical Security – Employee Limitation - Employee confidentiality agreement - Trade Secret Law - Unfair Competition – Trade Secret Litigation – Breach of Contract – Applying State Law

**TEXT BOOKS:**

1. Debirag E.Bouchoux: “Intellectual Property” 4e . Cengage learning, New Delhi
2. M.Ashok Kumar and Mohd.Iqbal Ali: “Intellectual Property Right” Serials Pub.
3. Cyber Law. Texts & Cases, South-Western’s Special Topics Collections
4. Prabhuddha Ganguli: ‘ Intellectual Property Rights’ Tata Mc-Graw –Hill, New Delhi
5. J Martin and C Turner “Intellectual Property” CRC Press
6. Richard Stimm “ Intellectual Property” Cengage Learning

**M.TECH (SOFTWARE ENGINEERING)**  
**I YEAR II SEMESTER**  
**NETWORK PROGRAMMING**  
**(OPEN ELECTIVE )**

Subject Code: **SE2016**

**Objectives:**

- To understand Linux utilities
- To understand file handling, signals
- To understand IPC, network programming in Java
- To understand processes to communicate with each other across a Computer Network.

**UNIT – I**

Linux Utilities- File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking utilities, Filters, Text processing utilities and Backup utilities. Bourne again shell(bash) - Introduction, 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. Review of C programming concepts-arrays, strings (library functions), pointers, function pointers, structures, unions, libraries in C.

**UNIT - II**

Files- File Concept, File types File System Structure, Inodes, File Attributes, file I/O in C using system calls, kernel support for files, 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. File and Directory management – Directory contents, Scanning Directories- Directory file APIs. Process- Process concept, Kernel support for process, process attributes, process control – process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process.

**UNIT – III**

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. Interprocess Communication - Introduction to IPC mechanisms, 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, Introduction to message queues, semaphores and shared memory. Message Queues- Kernel support for messages, UNIX system V APIs for messages, client/server example. Semaphores- Kernel support for semaphores, UNIX system V APIs for semaphores.

## **UNIT – IV**

Shared Memory- Kernel support for shared memory, UNIX system V APIs for shared memory, client/server example. Network IPC - Introduction to Unix Sockets, IPC over a network, Client-Server model, Address formats (Unix domain and Internet domain), Socket system calls for Connection Oriented - Communication, Socket system calls for Connectionless-Communication, Example-Client/Server Programs- Single Server-Client connection, Multiple simultaneous clients, Socket options – setsockopt, getsockopt, fcntl.

## **UNIT-V**

Network Programming in Java- Network basics, TCP sockets, UDP sockets (datagram sockets), Server programs that can handle one connection at a time and multiple connections (using multithreaded server), Remote Method Invocation (Java RMI)-Basic RMI Process, Implementation details-Client-Server Application.

## **TEXT BOOKS:**

1. Unix System Programming using C++, T.Chan, PHI.(Units II,III,IV)
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.(Unit I)
3. An Introduction to Network Programming with Java, Jan Graba, Springer, 2010.(Unit V)
4. Unix Network Programming, W.R. Stevens, PHI.(Units II,III,IV)
5. Java Network Programming, 3<sup>rd</sup> edition, E.R. Harold, SPD, O'Reilly.(Unit V)

## **REFERENCES:**

1. Linux System Programming, Robert Love, O'Reilly, SPD.
2. Advanced Programming in the UNIX environment, 2<sup>nd</sup> Edition, W.R.Stevens, Pearson Education.
3. UNIX for programmers and users, 3<sup>rd</sup> Edition, Graham Glass, King Ables, Pearson Education.
4. Beginning Linux Programming, 4<sup>th</sup> Edition, N.Matthew, R.Stones, Wrox, Wiley India Edition.
5. Unix Network Programming The Sockets Networking API, Vol.-I, W.R.Stevens, Bill Fenner, A.M.Rudoff, Pearson Education.
6. Unix Internals, U.Vahalia, Pearson Education.
7. Unix shell Programming, S.G.Kochan and P.Wood, 3<sup>rd</sup> edition, Pearson Education.
8. C Programming Language, Kernighan and Ritchie, PHI

**M.TECH (SOFTWARE ENGINEERING)**  
**I YEAR II SEMESTER**  
**SCRIPTING LANGUAGE**  
**(OPEN ELECTIVE)**

Subject Code: **SE2017**

**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.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2L10**

**DESIGN PATTERNS LAB  
(LAB – IV)**

***Learning Objective***

After completing this course, the student should be able to:

- Understand the concept of patterns and the Catalog.
- Discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
- Understand the variety of implemented bad practices related to the Business and Integration tiers.
- Highlight the evolution of patterns.
- How to add functionality to designs while minimizing complexity
- What design patterns really are, and are not
- About specific design patterns.
- What code qualities you need to maintain to keep code flexible.
- How to use design patterns to keep code quality high without overdesign.

**List of Sample Problems:**

1. Design and execute an application, where “abstract factory” pattern is applied.
2. Design and execute an application, where “adapter” pattern is applied.
3. Design and execute an application, where “bridge” pattern is applied.
4. Design and execute an application, where “command” pattern is applied.
5. Design and execute an application, where “composite” pattern is applied.
6. Design and execute an application, where “facade” pattern is applied.
7. Design and execute an application, where “observer” pattern is applied.
8. Design and execute an application, where “proxy” pattern is applied.
9. Design and execute an application, where “strategy” pattern is applied.
10. Specify and implement the most convenient pattern to apply to the following scenarios:
  - a) A component on the web tier requires access to business components.
  - b) There a need to provide several buttons on a web form which executes different actions.
  - c) Messages need to be sent to citizens each time a typhoon approaches.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2L20**

**SOFTWARE TESTING LAB  
(LAB – V)**

**Objectives:**

The student should be able to:

- To understand software testing and quality assurance as a fundamental component of software life cycle
- To define the scope of SW T&QA projects
- To efficiently perform T&QA activities using modern software tools
- To estimate cost of a T&QA project and manage budgets
- To prepare test plans and schedules for a T&QA project
- To develop T&QA project staffing requirements
- To effectively manage a T&QA project

**Software Testing Objectives:**

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. Refer Page no 115 in Text book 2(Foundations of software testing by Rex Black,Erik Van Veenendaal,Dorothy Graham) for the described scenario and observe the given
  - i. Equivalence Partitioning /Boundary Value Analysis
  - ii. Decision Tables
  - iii. State transition
  - iv. Statement and decision testing.consider any other scenario of your choice and do the same.
7. Refer Page no 158 in Text book 2(Foundations of software testing by Rex Black,Erik Van Veenendaal,Dorothy Graham) for the described scenario and observe the given **Incident Report** and consider any other scenario of your choice and do the same.
8. Study of any testing tool (e.g. Win runner)
9. Study of any web testing tool (e.g. Selenium)
10. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
11. Study of any test management tool (e.g. Test Director)
12. Study of any open source-testing tool (e.g. Test Link)
13. 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”**

**TEXT BOOKS:**

1. Software Testing Concepts and Tools, P.Nageswara Rao, Dream Tech Press, 2007.
2. Foundations of software testing by Rex Black, Erik Van Veenendaal, Dorothy Graham
3. Software Testing Concepts and Tools by Nageshwara Rao Pusuluri, Dream Tech Press
4. Software Testing Tools, *K.V.K.K. Prasad*, Dream Tech Press, 2008.
5. Software Testing with Visual Studio Team System 2008, S.Subashini, N.Satheesh kumar, Shroff Publishers Distributors.
6. Software Automation Testing Tools for Beginners, *Rahul Shende*, Shroff Publishers and Distributors, 2012.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2L31**

**NETWORK SECURITY LAB  
(PROFESSIONAL ELECTIVE-III LAB)  
(LAB – VI)**

**Objectives:**

- The Network Security Lab tries to present several hands-on exercises to help reinforce the students knowledge and understanding of the various network security aspects.
- To implement of cryptographic algorithms.

**The following exercises are based on the cryptographic algorithms. They can be implemented using C, C++, Java, etc.**

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 displays 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.
11. Implement DES-2 and DES-3 using Java cryptography package.
12. Write a Java program to implement RSA algorithm.
13. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties(Alice) and the JavaScript application as the other party(Bob)
14. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
15. Calculate the message digest of a text using the MD5 algorithm in JAVA.
16. Explore the Java classes related to digital certificates.
17. Create a digital certificate of your own by using the Java keytool.
18. Write a Java program to encrypt users passwords before they are stored in a database table, and to retrieve them whenever they are to be brought back for verification.
19. Key generation(public and private key pair) can be performed using Java. Write a program which can do this.
20. Write a program in java, which performs a digital signature on a given text.

21. Study phishing in more detail. Find out which popular bank sites have been phished and how.

**TEXT BOOK:**

1. Build Your Own Security Lab, Michael Gregg, Wiley India

***Learning Objective***

After completing this course, the student should be able to:

- Understand the concept of patterns and the Catalog.
- Discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
- Understand the variety of implemented bad practices related to the Business and Integration tiers.
- Highlight the evolution of patterns.
- How to add functionality to designs while minimizing complexity
- What design patterns really are, and are not
- About specific design patterns.
- What code qualities you need to maintain to keep code flexible.
- How to use design patterns to keep code quality high without overdesign.

**List of Sample Problems:**

11. Design and execute an application, where “abstract factory” pattern is applied.
12. Design and execute an application, where “adapter” pattern is applied.
13. Design and execute an application, where “bridge” pattern is applied.
14. Design and execute an application, where “command” pattern is applied.
15. Design and execute an application, where “composite” pattern is applied.
16. Design and execute an application, where “facade” pattern is applied.
17. Design and execute an application, where “observer” pattern is applied.
18. Design and execute an application, where “proxy” pattern is applied.
19. Design and execute an application, where “strategy” pattern is applied.
20. Specify and implement the most convenient pattern to apply to the following scenarios:
  - d) A component on the web tier requires access to business components.
  - e) There a need to provide several buttons on a web form which executes different actions.
  - f) Messages need to be sent to citizens each time a typhoon approaches.

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2L32**

**DATA ANALYTICS LAB  
(PROFESSIONAL ELECTIVE-III LAB)  
(LAB-VI)**

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 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/>

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2L33**

**INTERNET OF THINGS LAB  
(PROFESSIONAL ELECTIVE-III LAB)  
(LAB – VI)**

Following are some of the programs that a student should be able to write and test on an Raspberry Pi, but not limited to this only.

1. Start Raspberry Pi and try various Linux commands in command terminal window:

ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, ping etc.

2. Run some python programs on Pi like:

Read your name and print Hello message with name

Read two numbers and print their sum, difference, product and division.

Word and character count of a given string

Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard

input

Print a name  $n$  times, where name and  $n$  are read from standard input, using for and while loops.

Handle Divided by Zero Exception.

Print current time for 10 times with an interval of 10 seconds.

Read a file line by line and print the word count of each line.

3. Light an LED through Python program

4. Get input from two switches and switch on corresponding LEDs

5. Flash an LED at a given on time and off time cycle, where the two times are taken from a file.

6. Flash an LED based on cron output (acts as an alarm)
7. Switch on a relay at a given time using cron, where the relay's contact terminals are connected to a load.
8. Access an image through a Pi web cam.
9. Control a light source using web page.
10. Implement an intruder system that sends an alert to the given email.
11. Get the status of a bulb at a remote place (on the LAN) through web.
12. Get an alarm from a remote area (through LAN) if smoke is detected.

The student should have hands on experience in using various sensors like temperature, humidity, smoke, light, etc. and should be able to use control web camera, network, and relays connected to the Pi.



**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2L34**

**COMPUTER FORENSICS LAB  
(PROFESSIONAL ELECTIVE-III LAB)  
(LAB – VI)**

To perform the following tasks for the lab, Internet facility and open source tools should be provided.

1. Use a Web search engine, such as Google or Yahoo!, and search for companies specializing in computer forensics. Select three and write a two-to three-page paper comparing what each company does.(Project 1-1)
2. Search the Internet for articles on computer crime prosecutions. Find at least two. Write one to two pages summarizing the two articles and identify key features of the decisions you find in your search. (Project 1-5)
3. Use a Web search engine, search for various computer forensics tools.
4. Preparing and processing of investigations. Try to examine and identify the evidences from the drives. (Project 2-1)
5. Extracting of files that have been deleted.( Project 2-4)
6. Illustrate any Data acquisition method and validate. Use an open source data acquisition tool.
7. You're investigating an internal policy violation when you find an e-mail about a serious assault for which a police report needs to be filed. What should you do? Write a two-page paper specifying who in your company you need to talk to first and what evidence must be turned over to the police.(Project 5-2)
8. Create a file on a USB drive and calculate its hash value in FTK Imager. Change the file and calculate the hash value again to compare the files.(Project 5-4)
9. Compare two files created in Microsoft Office to determine whether the files are different at the hexadecimal level. Keep a log of what you find. (Project 6-1)
10. Illustrate the analysis of forensic data.
11. Illustrate the validating of forensic data.
12. Locate and extract Image (JPEG) files with altered extensions.(Project 10-1)
13. Examine or Investigate an E-mail message.

**TEXT BOOKS:**

- 
1. "Computer Forensics and Investigations", Nelson, Phillips Enfinger, Stuart, 3<sup>rd</sup> Edition, Cengage Learning.
-

**M.TECH (SOFTWARE ENGINEERING)  
I YEAR II SEMESTER**

Subject Code: **SE2A10**

**SOFT SKILLS LAB  
(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**’

- 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. Naterop, B. Jean and Revell, Rod. 2004. *Telephoning in English*. Cambridge: CUP
3. Patnaik, P. 2011. *Group Discussion and Interview Skills*. New Delhi: Foundation
4. Rizvi, M. A. 2005. *Effective Technical Communication*. New Delhi: Tata McGraw Hill
5. Sasikumar, V & Dhamija, P.V. 1993. *Spoken English - A Self-Learning Guide to Conversation Practice*. New Delhi: Tata McGraw-Hill
6. Sudhir Andrews. 2009. *How to Succeed at Interviews*. New Delhi: Tata McGraw Hill
7. Vivekananda: His Call to the Nation : a Compilation R.K. Math Publication