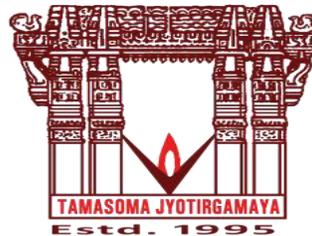


ACADEMIC REGULATIONS COURSE STRUCTURE AND DETAILED SYLLABUS

M. Tech. SOFTWARE ENGINEERING

(Applicable for the batches admitted from 2015-2016)



VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, Accredited by NAAC with 'A' Grade
NBA Accreditation for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes
Approved by AICTE, New Delhi, Affiliated to JNTUH
Recognized as "College with Potential for Excellence" by UGC
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

**VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
HYDERABAD
An Autonomous Institute**

**Academic Regulations - M.Tech. Programme
(Applicable for the batches admitted from the academic year 2015-2016)**

1. Introduction

Academic programmes of the institute are governed by rules and regulations as approved by the Academic Council of the institute.

These academic rules and regulations are effective from the academic year 2015-16, for the students admitted into two year post graduate programme offered by the college leading to Master of Technology (M. Tech.) degree in different specializations offered by the departments of Civil Engineering, Electrical and Electronics Engineering, Mechanical Engineering, Electronics and Communication Engineering, Computer Science and Engineering, Information Technology and Electronics and Instrumentation Engineering.

The M.Tech. degree of Jawaharlal Nehru Technological University Hyderabad shall be conferred on students who are admitted to the programme after fulfilling all the requirements for the award of the degree.

1.1 Eligibility for Admissions

Admission to the above program shall be made subject to the eligibility and qualifications prescribed from time to time. Admissions shall be made on the basis of GATE Rank and merit rank obtained at an Entrance Test conducted by the TSSCHE or as decided by TSSCHE subject to reservations prescribed by the university/ State Government from time to time.

2. Programmes of study

The following two year M.Tech. degree programmes of study are offered by the departments at VNR VJIET.

Department	Specializations
ME	1. Advanced Manufacturing Systems 2. Automation 3. CAD/CAM
CE	1. Highway Engineering 2. Structural Engineering 3. Geotechnical Engineering
EEE	1. Power Electronics 2. Power Systems
CSE	1. Software Engineering 2. Computer Science and Engineering
ECE	1. VLSI System Design 2. Embedded Systems
EIE	Electronics and Instrumentation
IT	Computer Networks and Information Security

- 'ENGLISH' language is used as the medium of instruction in all the above programmes.

3. Attendance requirements

Each academic year shall be divided into two semesters, each of 90 Instructions days, excluding examination, evaluation, declaration of results etc.

3.1 A student shall be eligible to appear for the semester end examinations in subject if he / she acquire a **minimum of 75% of attendance in that subject.**

3.2 Shortage of attendance up to 10% in any subject (i.e., attendance of 65% and above and below 75%) in a semester may be condoned by the Institute Academic Committee based on the rules prescribed by the Academic Council of the Institute from time to time.

3.3 A student shall get **minimum required attendance in at least three (03) theory subjects** in the present semester to get promoted to the next semester. In order to qualify for the award of the M.Tech. degree, the student shall complete all the academic requirements of the subjects, as per the course structure.

3.4 Shortage of **attendance below 65% shall in NO case be condoned.**

3.5 A stipulated fee shall be payable towards condonation of shortage of attendance.

3.6 In case the student secures less than the required attendance in any subject(s), he shall not be permitted to appear for the semester end examination in that subject(s). He shall re-register for the subject when offered next.

4. Evaluation

- i. The performance of a student in each semester shall be evaluated subject-wise with a maximum of **100 marks for theory** and **100 marks for practical subjects**. In addition, **mini-project** and **comprehensive viva-voce** shall be evaluated for **100 marks** respectively.
- ii. For theory subjects, the distribution shall be **40 marks for mid-term evaluation** and **60 marks for the semester end examination**.

❖ **Mid-Term Evaluation (40 M):**

Mid-term evaluation consists of **mid-term examination (30 M)** and **assignment/objective test/ case study/course project (10 M)**.

➤ **Mid-term examination (30 M):**

- For theory subjects, two mid-term examinations shall be conducted in each semester as per the academic calendar. Each mid-term examination shall be evaluated for 30 marks.
- Pattern of Mid-term examination:
3 X 10M = 30 M (three internal choice questions one from each UNIT shall be given, the student has to answer ONE question from each UNIT)
- There shall be TWO mid-term examinations for each subject and the average of two mid-term examinations shall be considered for calculating final mid-term examination marks in that subject.

➤ **Assignment/objective exam/ case study/course project (10 M):**

- Two assignment/objective exam/ case study/course project shall be given to the students covering the syllabus of first mid-term and second mid-term examinations respectively and evaluated for 10 marks each.

- The first assignment/objective exam/ case study/course project shall be submitted before first mid-term examination and the second one shall be submitted before second mid-term examination.
 - The average of 2 assignments shall be taken as final assignment marks.
- iii. For practical subjects, there shall be a **continuous evaluation during the semester for 40 marks and 60 marks for semester end examination**. Out of the 40 marks, **day-to-day work in the laboratory shall be evaluated for 10 marks**, and **15 marks for practical examination** and **15 marks for laboratory record**.

❖ **Semester End Examination (60 M):**

(a) Theory Courses

Question paper pattern for semester end examination (60 Marks)

- Paper shall consist of 05 questions of 10 marks each. (05X12M = 60 M)
- There shall be 01 question from each unit with internal choice.

(b) Practical Courses

Each laboratory course shall be evaluated for 60 marks. The semester end examination shall be conducted by two examiners, one Internal and other external concerned with the subject of the same / other department / Industry. The evaluation shall be as per the standard format.

- 4.1. **Evaluation of Mini-Project:** There shall be two presentations during the first year, one in each semester. For mini-project 1 and mini-project 2, a student under the supervision of a faculty member, shall collect the literature on a topic, critically review the literature, carry out the mini-project, submit it to the department in a report form and shall make an oral presentation before the departmental Project Review Committee (PRC). The Departmental PRC consists of Head of the Department, supervisor and one senior faculty member of the department. For each mini-project there shall be only internal evaluation of 100 marks. A student has to secure a minimum of 50% to be declared successful.
- 4.2. There shall be a comprehensive viva-voce in II year I semester. The comprehensive viva- Voce shall be conducted by a committee consisting of Head of the Department and two senior faculty members of the department. The comprehensive viva-voce is aimed to assess the students' understanding in various subjects studied during the M.Tech. programme of study. The comprehensive viva-voce shall be evaluated for 100 marks by the committee. There are no internal marks for the comprehensive viva-voce. A student must secure a minimum of 50% to be declared successful.
- 4.3. A student 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 mid-term evaluation taken together.
- 4.4. A student shall be given one chance to re-register, after completion of the course work, for each subject, provided the internal marks secured by a student are less than 50% and he has failed in the semester end examination. In such a case student may re-register for the subject(s) and secure required minimum attendance. Attendance in the re-registered subject(s) has to be calculated separately to become eligible to write the end examination in the re-registered subject(s). Re-registration for the subjects is allowed only if that particular re-registration subjects are the hindrance for the award of Degree. Re-registration is allowed in this case provided the student doesn't have any subject(s) yet to pass other than the re-registration subjects where the internal marks are less than 50% with prior permission.

Laboratory examination for M.Tech. courses must be conducted with two examiners, one of them being laboratory class teacher and second examiner shall be a teacher of same specialization either external or a teacher from the same department other than the teacher who conducted laboratory classes for that batch.

- 4.5. Laboratory examination for M.Tech. courses must be conducted with two examiners, one of them being laboratory class teacher and second examiner shall be a teacher of same specialization either external or a teacher from the same department other than the teacher who conducted laboratory classes for that batch.

5. Evaluation of Project / Dissertation Work.

5.1 Registration of Project Work: A student shall be permitted to register for the project work after satisfying the attendance requirement of all the subjects (theory and practical subjects).

5.2 A Project Review Committee (PRC) shall be constituted with at least four members namely HOD, PG coordinator of the M.Tech. programme, project supervisor and one senior faculty member of same specialization.

5.3 After getting permission as per 5.1, a student has to submit, in consultation with the project supervisor, the title, objective and plan of action of his project work to the Departmental PRC for its approval. Only after obtaining the approval of PRC, the student can initiate the project work.

5.4 If a student wishes to change his supervisor or topic of the project he can do so with the approval of PRC. However, the committee shall examine whether the change of topic/supervisor leads to a major change of his initial plans of project proposal. If so, the date of registration for the project work shall be the date of change of supervisor or topic as the case may be.

5.5 Internal evaluation of the project shall be on the basis of the seminars (Project reviews) conducted during the second year by the PRC. A student shall submit draft report in a spiral bound copy form.

5.6 The work on the project shall be initiated in the beginning of the second year and the duration of project is for two semesters. A student is permitted to submit Project work only after successful completion of theory and practical course with the approval of PRC not earlier than 240 days from the date of registration of the project work. For the approval of PRC the student shall submit the draft copy of thesis to the Head of the Department (Through project supervisor and PG coordinator) and shall make an oral presentation before the PRC.

The student is eligible to submit project work if he has published at least one paper covering 70% of the project work and presented his project work in Show and Tell activity.

5.7 After approval of PRC, every student has to submit three copies of the project dissertation certified by the supervisor to the Department.

5.8 The dissertation shall be adjudicated by one examiner selected by the Chief Superintendent. For this, HOD shall submit a panel of 3/ 5 examiners, who are eminent in that field with the help of the concerned guide.

5.9 If the report of the examiner is not favourable, the student shall revise and resubmit the Dissertation, within the time frame as prescribed by PRC. If the report of the examiner is unfavourable again, the dissertation shall be summarily rejected.

5.10 If the report of the examiner is favorable, viva-voce examination shall be conducted by a board consisting of the project supervisor, Head of the Department and the external examiner who adjudicated the Thesis. The Board shall jointly report students work as:

- A. Excellent**
- B. Good**
- C. Satisfactory**
- D. Unsatisfactory**

Head of the Department shall coordinate and make arrangements for the conduct of viva-voce examination. The student has to secure any one of the grades as Excellent, Good or Satisfactory on his dissertation and viva-voce. If the report of the viva-voce is unsatisfactory, the student shall retake the viva-voce examination after three months, making modifications as suggested. If he fails to get a satisfactory report at the second viva-voce examination, he has to re-register for the project work as mentioned in clause 5.1. However, the student may select a new guide or new topic or both with the approval of the PRC and submit the project dissertation with a minimum of 240 days from the date of re-registration. Of course, this shall not prejudice the clause 6.1 below.

6. Award of Degree and Class

A student shall be declared eligible for the award of the M.Tech. degree, if he pursues a course of study and complete it successfully for **not less than two academic years** and **not more than four academic years**.

6.1 A student, who fails to fulfil all the academic requirements for the award of the degree within four academic years from the year of his admission, for any reason whatsoever, shall forfeit his seat in M.Tech. Course.

6.2 A student shall register and put up **minimum academic requirement in all 84 credits** and earn **84 credits**. Marks obtained in all 86 credits shall be considered for the calculation of Cumulative Grade Point Average (CGPA).

6.3 CGPA System:

Method of awarding absolute grades and grade points in two year M.Tech. degree programme is as follows:

- Absolute Grading Method is followed, based on the total marks obtained in mid-term evaluation and semester end examinations.
- Grades and Grade points are assigned as given below.

Marks Obtained	Grade	Description of Grade	Grade Points(GP) Value Per Credit
>=90	O	Outstanding	10.00
>=80 and <89.99	A	Excellent	9.00
>=70 and <79.99	B	Very Good	8.00
>=60 and <69.99	C	Good	7.00
>=50 and <59.99	D	Pass	6.00
<50	F	Fail	
Not Appeared the Exam(s)	N	Absent	

The student is eligible for the award of the M.Tech degree with the class as mentioned in the following table.

CGPA	Class
≥ 8.0	First Class with Distinction
≥ 7.0 and < 8.0	First Class
≥ 6.0 and < 7.0	Second Class

➤ **Calculation of Semester Grade Points Average (SGPA):**

- The performance of each student at the end of the each semester shall be indicated in terms of SGPA. The SGPA shall be calculated as below:

$$\text{SGPA} = \frac{\text{Total earned weighted grade points in a semester}}{\text{Total credits in a semester}}$$

$$\text{SGPA} = \frac{\sum_{i=1}^p C_i * G_i}{\sum_{i=1}^p C_i}$$

Where C_i = Number of credits allotted to a particular subject 'i'

G_i = Grade point corresponding to the letter grade awarded to the subject 'i'

$i = 1, 2, \dots, p$ represent the number of subjects in a particular semester

Note: SGPA is calculated and awarded for the students who pass all the courses in a semester.

➤ **Calculation of Cumulative Grade Point Average (CGPA):**

The CGPA of a student for the entire programme shall be calculated as given below:

- Assessment of the overall performance of a student shall be obtained by calculating Cumulative Grade Point Average (CGPA), which is weighted average of the grade points obtained in all subjects during the course of study.

$$\text{CGPA} = \frac{\text{Total earned weighted grade points for the entire programme}}{\text{Total credits for the entire programme}}$$

$$\text{CGPA} = \frac{\sum_{j=1}^m C_j * G_j}{\sum_{j=1}^m C_j}$$

Where C_j = Number of credits allotted to a particular subject 'j'

G_j = Grade Point corresponding to the letter grade awarded to that subject 'j'

$j = 1, 2, \dots, m$ represent the number of subjects of the entire program.

- Grade lower than D in any subject shall not be considered for CGPA calculation. The CGPA shall be awarded only when the student acquires the required number of credits prescribed for the program.

➤ **Grade Card**

The grade card issued shall contain the following:

- a) The credits for each subject offered in that semester
- b) The letter grade and grade point awarded in each subject
- c) The SGPA/CGPA
- d) Total number of credits earned by the student up to the end of that semester.

7. Withholding of Results

If the student has not paid dues to the Institute, or if any case of indiscipline is pending against him, the result of the student may be withheld and he shall not be allowed into the next higher semester. The award or issue of the provisional certificate and the degree may also be withheld in such cases. This delay shall not prejudice clauses Nos.6.0 and 6.1.

8. Transitory Regulations

Students who have discontinued or have been detained for want of attendance or any other academic requirements, may be considered for readmission as and when they become eligible. They have to take up Equivalent subjects, as substitute subjects in place of repeated subjects as decided by the Chairman of the BoS of the respective departments. He/She shall be admitted under the regulation of the batch in which he/she is readmitted.

9. Minimum Instruction Days

The minimum instruction days for each semester shall be **90 instruction days**.

10. General

10.1 The academic regulations should be read as a whole for purpose of any interpretation.

10.2 In case of any doubt or ambiguity in the interpretation of the above rules, the decision of the Principal is final.

10.3 The Institute may change or amend the academic regulations and syllabi at any time and the changes and amendments made shall be applicable to all the students with effect from the date notified by the Institute.

10.4 Wherever the words he, him or his occur, they shall also include she, her and hers.

11. Supplementary Examination

Supplementary examinations shall be conducted along with regular semester end examinations. (During even semester regular examinations, supplementary examinations of odd Semester and during odd semester regular examinations, supplementary examinations of even semester shall be conducted).

**VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.TECH. (SOFTWARE ENGINEERING)
(R15 Regulation)**

I Year I Semester

COURSE STRUCTURE

Code	Group	Subject	L	T/P/D	Credits
SWE01	Core	Object Oriented Modelling	3	1	4
SWE02		Software Requirements and Estimation	3	1	4
SWE03		Java and Web Technologies	3	1	4
SWE11	Elective – I & Elective-II Basket	Computer Vision	3	0	3 + 3
SWE12		Information Storage Management	3	0	
SWE13		Ethical Hacking and Computer Forensics	3	0	
SWE14		Software Architecture and Design Patterns	3	0	
SWE15		Formal Methods in Software Engineering	3	0	
SWE16		Cloud Computing	3	0	
SWE31	Open Elective -I	Scripting Languages	3	0	3
CNS31		Security in E-Commerce	3	0	
ENG32		Professional and Technical Communication	3	0	
SWE51	Lab	Java and Web Technologies & OOM Laboratory	0	3	2
SWE61		Mini Project - I	0	0	4
Total			18	6	27

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

M.TECH. (SOFTWARE ENGINEERING)
(R15 Regulation)

I Year II Semester

COURSE STRUCTURE

Code	Group	Subject	L	T/P/D	Credits
SWE04	Core	Data Analytics	3	1	4
SWE05		Web Services and Service Oriented Architecture	3	1	4
SWE06		Software Quality Assurance and Testing	3	1	4
SWE21	Elective – III & Elective-IV Basket	Advanced Software Engineering Principles and Practices	3	0	3 + 3
SWE22		Software Metrics	3	0	
SWE23		Software Reengineering	3	0	
SWE24		Research Methodologies	3	0	
SWE25		Internet of Things	3	0	
SWE26		Software Defined Networks	3	0	
SWE41	Open Elective -II	Advanced Unix Programming	3	0	3
CNS41		Android Application Development	3	0	
SWE42		Soft Skills	3	0	
SWE52	Lab	Software Testing, Case Tools and Analytics Laboratory	0	3	2
SWE62		Mini Project - II	0	0	4
Total			18	6	27

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

M.TECH. (SOFTWARE ENGINEERING)
(R15 Regulation)

II Year I Semester

COURSE STRUCTURE

Code	Subject	L	P	Credits
SWE63	Comprehensive viva voce	0	0	4
SWE71	Internship / Dissertation Phase - I	0	0	8
Total				12

II Year II Semester

COURSE STRUCTURE

Code	Subject	L	P	Credits
SWE72	Dissertation Phase - II	0	0	18
Total				18

(SWE01) OBJECT ORIENTED MODELLING

Course Objectives:

The student should be made to:

- List and discuss the object oriented concepts , principles and the artifacts that are related to classes and relationships for modeling (CRC approach)
- Illustrate the communication across the objects for a specified behavior (use case) through interaction diagrams.
- Outline the various states a process/thread undergoes based on occurrence of events and the mode for deploying the components being built.
- Explain the industry software modeling practice through various processes

Course Outcomes:

After completion of this course the student is able to:

- Relate object oriented concepts representation through artifacts of UML.
- Demonstrate classes, their relationships and collaborations (CRC) (for any case study).
- Generate the list and order of activities carried out for each behavior exhibited by the system(for any case study) to design and deploy the model
- Apply unified process approach for case studies(ATM, Automation of a Library, Software Simulator application, etc)

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.

UNIT II:

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams. Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams. **Advanced structural modeling:** Advanced classes, Advanced relationships, Interfaces types and roles, packages, instances and object diagrams.

UNIT III:

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. **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.

UNIT IV:

Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams. **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.

UNIT V:

Architecture-Centric Process: architecture in brief, why we need architecture, use cases and architecture, the steps to architecture, an architecture description.

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

Case Studies: Automation of a Library, Software Simulator application.

Text Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modelling Language User Guide, Pearson Education.
2. Ivar Jacobson, Grady Booch, James Rumbaugh: The Unified Software Development Process, Pearson Education.

References:

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning, Object-Oriented Analysis and Design with the Unified Process.
5. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY dreamtech.

(SWE02) SOFTWARE REQUIREMENTS AND ESTIMATION

Course Objectives:

- Learn the concepts of software requirements.
- Understand the stages of requirements engineering.
- Know the techniques used for requirements development and modeling.
- Learn the requirements management process.
- Study and analyze the estimation methods.

Course Outcomes:

After completion of this course the student is able to:

- Identify and classify the requirements.
- Select appropriate elicitation technique for a system.
- Develop various requirements models.
- Apply the change management process.

UNIT I:

Software Requirements: What and Why: Essentials of Software requirement Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management

UNIT II:

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,

Software Requirements Modeling: Use Case Modeling, Analysis Models, Dataflow diagram, state transition diagram, class diagrams, Object analysis, Problem Frames

UNIT III:

Software Requirements Management: Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain

Requirements Management Tools: Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation,

UNIT IV:

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 V:

Effort, Schedule and Cost Estimation: Productivity, Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Algorithmic models, Cost Estimation

Software Estimation Tools: Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools

Text Books:

1. Software Requirements by Karl E. Weigers, Microsoft Press.
2. Software Requirements and Estimation by Rajesh Naik and Swapna Kishore, Tata Mc Graw Hill

References:

1. Managing Software Requirements, Dean Leffingwell & Don Widrig, Pearson Education, 2003.
2. Mastering the requirements process, second edition, Suzanne Robertson & James Robertson, Pearson Education, 2006.
3. Estimating Software Costs, Second edition, Capers Jones, Tata McGraw-Hill, 2007.
4. Practical Software Estimation, M.A. Parthasarathy, Pearson Education, 2007.
5. Measuring the software process, William A. Florac & Anita D. Carleton, Pearson ducation, 1999.

(SWE03) JAVA AND WEB TECHNOLOGIES

Course Objectives:

At the end of the course, student will be able to

- Identify issues related to the definition, creation and usage of HTML, DHTML, XHTML, JavaScript and CSS.
- Discuss the principles of inheritance and polymorphism and demonstrate through problem analysis assignments how they relate to the design of swings methods, abstract classes and interfaces.
- Build robust web applications using MVC architecture, Servlets, SessionManagement, filters.
- Design view components using JSP EL and JSTL and its API and integrate these with JDBC.

Course Outcomes:

After completion of this course the student is able to:

- Design and implement dynamic websites with HTML and CSS to get a good aesthetic sense of designing and latest technical know-how's.
- Investigate on knowledge of xml and develop XML Technologies such as XML Schemas, XSD, usage of xml technology in Electronic Data Interchange.
- Explore the intricacies of JavaScript for creation of dynamic effects, validate form input entry on web sites and develop appropriate client-side applications.
- Design and develop web based enterprise systems for the enterprises using technologies like Jsp, Servlet integrating it with database.

UNIT I:

HTML Common Tags: List, Tables, images, forms, Frames; Cascading Style sheets;

Introduction to Java Scripts: Objects in Java Script, Dynamic HTML with Java Script, CSS

UNIT II:

XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Review of Applets, Class, Event Handling, AWT Programming.

Introduction to Swing: JApplet, Handling Swing Controls like Icons – Labels – Buttons – Text Boxes – Combo – Boxes – Tabbed Pains – Scroll Pains – Trees – Tables Differences between AWT Controls & Swing Controls Developing a Home page using Applet & Swing.

UNIT III:

Java Beans: Introduction to Java Beans, Advantages of Java Beans, JDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API.

Web Servers: Tomcat Server installation & Testing.

Introduction to Servlets: Lifecycle of a Servlet, JSDK The Servlet API, The javax.servelet Package, Reading Servlet parameters, Reading Initialization parameters.

UNIT IV:

More on Servlets: The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues.

Introduction to JSP: The Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC architecture. AJAX.

UNIT V:

JSP Application Development: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages, Requests, and

Users Passing Control and Data between Pages – Sharing Session and Application Data – Memory Usage Considerations
Database Access Database Programming using JDBC Studying Javax.sql.* package Accessing a Database from a JSP Page Application – Specific Database Actions Deploying JAVA Beans in a JSP Page

Text Books:

1. Web Programming, building internet applications, Chris Bates 2nd edition, WILEY Dreamtech (UNIT 1,2)
2. The complete Reference Java 2 Fifth Edition ,Patrick Naughton and Herbert Schildt., TMH (Chapters:25) (UNIT 2,3)
3. Java Server Pages –Hans Bergsten, SPD O'Reilly (UNITs 3,4,5)

References:

1. Programming world wide web-Sebesta,Pearson
2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES, Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program , Dietel and Nieto PHI/Pearson.
4. Jakarta Struts Cookbook , Bill Siggelkow, S P D O'Reilly for chap 8.
5. Murach's beginning JAVA JDK 5, Murach, SPD

(SWE11) COMPUTER VISION
(Elective-1&2)

Course Objectives:

- Identify the basic concepts and techniques in computer vision.
- Train the computer to understand and interpret visual information
- Evaluate the usefulness and performance of computer vision.

Course Outcomes:

After completion of this course the student is able to:

- Implement the mathematical models of images.
- Correlate and compare the feature extractions techniques in order to extract the images.
- Synthesis principles and algorithms of computer vision to analyze the patterns and motion of images.
- Describe at least two methods for object reorganization

UNIT I:

Digital Image Formation and Low-Level Processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

UNIT II:

Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT

UNIT III:

Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

UNIT IV:

Pattern Analysis: Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

UNIT V:

Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation

Text Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.

References:

1. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004
2. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992
4. IEEE-T-PAMI (IEEE Transactions on Pattern Analysis and Machine Intelligence).
5. IJCV (International Journal of Computer Vision) - Springer

(SWE12) INFORMATION STORAGE MANAGEMENT
(Elective – 1 & 2)

Course Objectives:

- Discuss Storage Area Networks characteristics and components and list SAN vendors and their products.
- Describe the concept of RAID and different RAID levels and their suitability for different application environments, different backup and recovery topologies and their role in providing disaster recovery and business continuity capabilities
- Apply Fiber Channel protocols and how SAN components use them to communicate with each other and Identify different networked storage options for different application environments.
- Classify Cisco MDS 9000 Multilayer Directors and Fabric Switches and define the Cisco SAN-OS features and commands.

Course Outcomes:

After completion of this course the student is able to:

- Identify the functions to build data center networking for switch network.
- Discuss different types of logical and physical components of a storage infrastructure and different types of RAID implementations and their benefits.
- Demonstrate the importance of Fibre Channel protocols and how to communicate with each other and benefits of the different network storage options for different application environments.
- Identify the solutions for single points of failure in a storage infrastructure and in providing disaster recovery and business continuity capabilities

UNIT I:

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

UNIT II:

Storage Systems Architecture: Hardware and software components of the host environment, Key protocols and concepts used by each component ,Physical and logical components of a connectivity environment ,Major physical components of a disk drive and their function, logical constructs of a physical disk, access characteristics, and performance Implications, Concept of RAID and its components , Different RAID levels and their suitability for different application environments: RAID 0, RAID 1, RAID 3, RAID 4, RAID 5, RAID 0+1, RAID 1+0, RAID 6, Compare and contrast integrated and modular storage systems ,High-level architecture and working of an intelligent storage system

UNIT III:

Introduction to Networked Storage: Evolution of networked storage, Architecture, components, and topologies of FC-SAN, NAS, and IPSAN, Benefits of the different networked storage options, understand the need for long-term archiving solutions and describe how CAS fulfills the need, understand the appropriateness of the different networked storage options for different application environments

UNIT IV:

Information Availability & Monitoring & Managing Datacenter: List reasons for planned/unplanned outages and the impact of downtime, Impact of downtime, Differentiate between business continuity (BC) and disaster recovery (DR) ,RTO and RPO, Identify single points of failure in a storage infrastructure and list solutions to mitigate these failures , Architecture of backup/recovery and the different backup/recovery topologies , replication technologies and their role in ensuring information availability and business

continuity, Remote replication technologies and their role in providing disaster recovery and business continuity capabilities Identify key areas to monitor in a data center, Industry standards for data center monitoring and management, Key metrics to monitor for different components in a storage infrastructure, Key management tasks in a data center

UNIT V:

Securing Storage and Storage Virtualization: Information security, Critical security attributes for information systems, Storage security domains, List and analyzes the common threats in each domain, Virtualization technologies, block-level and filelevel virtualization technologies and processes Case Studies The technologies described in the course are reinforced with EMC examples of actual solutions. Realistic case studies enable the participant to design the most appropriate solution for given sets of criteria.

Text Books:

1. EMC Corporation, Information Storage and Management, Wiley.
2. Robert Spalding, "Storage Networks: The Complete Reference", Tata McGraw Hill, Osborne, 2003.

References:

1. Marc Farley, "Building Storage Networks", Tata McGraw Hill ,Osborne, 2001.
2. Meeta Gupta, Storage Area Network Fundamentals, Pearson Education Limited, 2002

VNR Vignana Jyothi Institute of Engineering and Technology			
I Year – I Sem. M.Tech (Software Engineering)	L	P	C
	3	0	3

(SWE13) ETHICAL HACKING AND COMPUTER FORENSICS
(Elective – 1 & 2)

Course Objectives:

- Assess and measure threats to information assets.
- Evaluate where information networks are most vulnerable.
- Describe the origin of computer forensics and the relationship between law enforcement and industry.
- Compose electronic evidence and the computing investigation process.

Course Outcomes:

After completion of this course the student is able to:

- Identify and analyze the stages an ethical hacker requires to take in order to compromise a target system.
- Implement security techniques and tools used to protect system and user data.
- Identify, analyze, and mitigate threats to internal computer systems.
- Produce various forensic tools to collect digital evidence.

UNIT I:

Introduction to Hacking: Basics of Hacking Techniques, Ethics of Hacking, Hacking techniques, Information War, Introduction to Ethical Hacking.

Password Cracking: Introduction, Password Stealing, Password Crackers

UNIT II:

Sniffers: Introduction to Sniffers, Working of a Sniffer, Sniffer Programs, Detecting a Sniffer, Protecting Against a Sniffer.

Buffer Overflows: Introduction, Types of Buffer Overflow, Methods to Cause a Buffer Overflow, Buffer Overflows: Detection and Prevention

UNIT III:

Denial-of-Service Attacks: Denial-of-Service Attacks, Flood Attacks, Software Attacks, Distributed Denial-of-Service, Prevention of DoS Attacks.

Scanning Tools: Introduction, Scanners.

UNIT IV:

Introduction to Computer Forensics & Investigations: Computer Forensics & Investigations as a profession, understanding computer investigations, data acquisition, processing crime and incident scenes, Network forensics, cell phone and mobile device forensics.

UNIT V:

Searching and Seizing Computer Related Evidence; Processing Evidence and Report Preparation;

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. Hacking tools & Technique for incident Handling-NIIT-PHI Learning
2. Computer Forensics and investigations- Nelson, Phillips Enfinger, Steuart- CENGAGE Learning

References:

1. Computer Forensics and Cyber Crime: An Introduction- Marjie T Britz- Pearson
2. Ethical Hacking and Network Defense – Michael T.Simpson- CENGAGE Learning

(SWE14) SOFTWARE ARCHITECTURE AND DESIGN PATTERN
(Elective – 1 & 2)

Course Objectives:

- Outline the fundamentals of software architecture and quality requirements of a software system.
- Define the fundamental principles and guidelines for software architecture design, architectural styles, patterns, and frameworks.
- Identify the methods, techniques, and tools for software architecture and document different software product lines.
- Relate various design patterns for software architecture system.

Course Outcomes:

After completion of this course the student is able to:

- Construct and design suitable software architecture for small software systems.
- Analyse major software architectural styles, design patterns, and frameworks.
- Elaborate software architecture using various documentation approaches and architectural description languages.
- Demonstrate a number of the fundamental patterns and principles of software architectural styles and design any case study

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.

UNIT III:

Moving from One System to Many: Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT IV:

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, facade, flyweight Proxy.

UNIT V:

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

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 Ka/man, Pearson Education,2003.
2. Design Patterns, Erich Gamma, Pearson Education, 1995.

References:

1. Beyond Software architecture, Luke I-lohmann, Addison weslcy, 2003.\
2. Software Architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR,200|
3. Pattern Oriented Software Architecture, F. Buschmann & others, John Wiley & Sons.
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

(SWE15) FORMAL METHODS IN SOFTWARE ENGINEERING
(Elective – 1 & 2)

Course Objectives:

- Develop an appreciation of the strengths of formal methods for engineering secure software systems.
- Build up a solid background for the application of formal methods to various tasks of the software development process using Z.
- Gain a basic level of competence in using formal methods to model software systems and Verify software system properties
- Construct formal methods of software engineering through Z.

Course Outcomes:

After completion of this course the student is able to:

- Explore the role of formal methods in software engineering and glimpse of Z.
- Produce a design and specification for an application using Z
- Critically evaluate features of Z and build methods in construction of software systems.
- Apply and design formal methods to system applications through Z

UNIT I:

Foundations of Z: Understanding formal methods – motivation for formal methods – informal requirements to formal specifications – validating formal specifications – Overview of Z specification – basic elements of Z – sets and types – declarations – variables – expressions – operators – predicates and equations.

UNIT II:

Structures in Z: Tuples and records – relations, tables, databases – pairs and binary relations – functions – Sequences – propositional logic in Z – predicate logic in Z – Z and Boolean types – set Comprehension – lambda calculus in Z – simple formal specifications – modelling systems and Change.

UNIT III:

Z Schemas and Schema Calculus: Z schemas – schema calculus – schema conjunction and disjunction – other schema calculus Operators – schema types and bindings – generic definitions – free types – formal reasoning – Checking specifications – precondition calculation – machine-checked proofs.

UNIT IV:

Z Case Studies: Case Study: Text processing system – Case Study: Eight Queens – Case Study: Graphical User Interface – Case Study: Safety critical protection system – Case Study: Concurrency and real time systems.

UNIT V:

Z Refinement: Refinement of Z specification – generalizing refinements – refinement strategies – program derivation and verification – refinement calculus – data structures – state schemas – functions and relations – operation schemas – schema expressions – refinement case study.

Text Books:

1. Jonathan Jacky, “The way of Z: Practical programming with formal methods”, Cambridge University Press, 1996.
2. Antony Diller, “Z: An introduction to formal methods”, Second Edition, Wiley, 1994.

References:

1. Jim Woodcock and Jim Davies, “Using Z – Specification, Refinement, and Proof”, Prentice Hall, 1996.

2. J. M. Spivey, "The Z notation: A reference manual", Second Edition, Prentice Hall, 1992.
3. M. Ben-Ari, "Mathematical logic for computer science", Second Edition, Springer, 2003.
4. M. Huth and M. Ryan, "Logic in Computer Science – Modelling and Reasoning about Systems", Second Edition, Cambridge University Press, 2004.
5. Harry, Formal Methods Fact File VDM

(SWE16) CLOUD COMPUTING
(Elective – 1 & 2)

Course Objectives:

- Understand cloud computing paradigm, recognize its various forms
- Get a clear understanding of Cloud Computing fundamentals and its importance to various organizations.
- Master the concepts of IaaS, PaaS, SaaS, Public and Private clouds.
- Understand AWS and learn to develop applications in AWS.

Course Outcomes:

After completion of this course the student is able to:

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing
- Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
- Explain the core issues of cloud computing such as security, privacy, and interoperability.
- Identify problems, and explain, analyze, and evaluate various cloud computing solutions.

UNIT I:

Systems Modelling, Clustering and Virtualization: Distributed System Models and Enabling Technologies, Computer Clusters for Scalable Parallel Computing, Virtual Machines and Virtualization of Clusters and Data centres.

UNIT II:

Foundations: Introduction to Cloud Computing, Migrating into a Cloud, Enriching the 'Integration as a Service' Paradigm for the Cloud Era, the Enterprise Cloud Computing Paradigm.

UNIT III:

Infrastructure as a Service (IAAS) & Platform and Software as a Service (PAAS / SAAS): Virtual machines provisioning and Migration services, On the Management of Virtual machines for Cloud Infrastructures, Enhancing Cloud Computing Environments using a cluster as a Service, Secure Distributed Data Storage in Cloud Computing. Aneka, Comet Cloud, T-Systems', Workflow Engine for Clouds, Understanding Scientific Applications for Cloud Environments.

UNIT IV:

Monitoring, Management and Applications: An Architecture for Federated Cloud Computing, SLA Management in Cloud Computing, Performance Prediction for HPC on Clouds, Best Practices in Architecting Cloud Applications in the AWS cloud, Building Content Delivery networks using Clouds, Resource Cloud Mashups.

UNIT V:

Governance and Case Studies: Organizational Readiness and Change management in the Cloud age, Data Security in the Cloud, Legal Issues in Cloud computing, Achieving Production Readiness for Cloud Services.

Text Books:

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing , Kai Hwang, Geoffery C.Fox, Jack J.Dongarra, Elsevier, 2012.

3. Cloud Computing : A Practical Approach, Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Tata McGraw Hill, rp2011

References:

1. Cloud Computing : A Practical Approach, Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Tata McGraw Hill, rp2011.
2. Enterprise Cloud Computing, Gautam Shroff, Cambridge University Press, 2010.
3. Cloud Computing: Implementation, Management and Security, John W. Rittinghouse, James F.Ransome, CRC Press, rp2012.
4. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, George Reese, O'Reilly, SPD, rp2011.
5. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, rp2011

(SWE31) SCRIPTING LANGUAGES
(Open Elective-1)

Course Objectives:

- To appreciate the nature of scripting and the role of scripting languages
- To effectively apply knowledge of scripting to new situations and learn from the experience
- To be able to analyze requirements of software systems for the purpose of determining the suitability of implementation of PERL, PHP or PYTHON
- To design and implement PERL, PHP or PYTHON software solutions that accommodate specified requirements and constraints, based on analysis or modeling or requirements specification

Course Outcomes:

After completion of this course the student is able to:

- Distinguish between typical Scripting Languages & system and application programming languages
- Apply the syntax and semantics of languages such as PERL, PHP and PYTHON for effective scripting
- Develop a database driven PHP application
- Design and implement the appropriate software solutions using Scripting Languages

UNIT I:

Introduction to PERL and Scripting: Scripts and Programs, Origin of Scripting, Characteristics of Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines

UNIT II:

Advance Perl: Finer points of looping, pack and unpack, file system, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, security Issues.

UNIT III:

PHP Fundamentals: PHP Basics- Features Embedding PHP Code in your Web pages, Outputting the data to the browser, Data types, Variables, Constants, expressions, string interpolation, control structures Function, Creating a Function, Function Libraries, Arrays, strings and Regular Expressions.

UNIT IV:

Advanced PHP and MYSQL: PHP and Web Forms, Files, PHP Authentication and Methodologies -Hard Coded, File Based, Database Based, IP Based, Login Administration, Uploading Files with PHP, Sending Email using PHP, PHP Encryption Functions, the Mcrypt package, Introducing MYSQL, Installing and Configuring MYSQL, MYSQL Storage Engines and Datatypes, PHP's MYSQL Extension-PHP's MySQL Commands, Querying MySQL, Retrieving and Displaying Data, Retrieving Database and Table Information Retrieving Field Information.

UNIT V:

Python: Introduction to Python language, python-syntax, statements, functions, Built-in-functions and Methods, Modules in python, Exception Handling, Integrated Web Applications in Python - Building Small, Efficient Python Web Systems ,Web Application Framework.

Text Books:

1. The World of Scripting Languages, David Barren, Wiley Publications.

2. Python Web Programming, Steve Holden and David Beazley, New Riders Publications.
3. Beginning PHP and MySQL, 3rd Edition, Jason Gilmore, press Publications (Dream tech.).

References:

1. Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and PHP J. Lee and B.Ware (Addison Wesley) Pearson Education.
2. Programming Python, M. Lutz, SPD.
3. PHP 6 Fast and Easy Web Development Julie Meloni and Matt Telles, Cengage Learning Publication
4. PHP 5.1 J. Bayross and S. Shah, The X Team, SPD.
5. Core Python Programming, Chun, Pearson Education.

(CNS31) SECURITY IN E-COMMERCE
(Open Elective-1)

Course Objectives:

- To prepare student to understand current threats facing organizations that conduct business online
- To apply cryptography and related security techniques to e-commerce including secure electronic transactions
- To understand Electronic payment systems, intellectual property protection
- To know and understand security development cycle and security services

Course Outcomes:

After completion of this course the student is able to:

- Describe the various forms of electronic commerce and explain the range of threats to e-commerce security
- Identify the different areas susceptible to malicious activity
- Learn different cryptographic techniques and their technical characteristics with respect to E-commerce
- Explain how cryptography can be, and is, used to achieve security

UNIT I:

Introduction to E-Commerce: Network and E-Commerce – Types of E-Commerce – E-Commerce Business Models: B2C, B2B, C2C, P2P and M-commerce business models – E-Commerce Payment systems: Types of payment system – Credit card E-Commerce transactions – B2C E-Commerce Digital payment systems – B2B payment system.

UNIT II:

Security and Encryption: E-Commerce Security Environment – Security threats in E-Commerce environment – Policies, Procedures and Laws.

UNIT III:

Inter-organizational Trust in E-Commerce: Need – Trading partner trust – Perceived benefits and risks of Ecommerce – Technology trust mechanism in E-Commerce – Perspectives of organizational, economic and political theories of inter-organizational trust – Conceptual model of inter-organizational trust in Ecommerce participation.

UNIT IV:

Introduction to Trusted Computing Platform: Overview – Usage Scenarios – Key components of trusted Platform – Trust mechanisms in a trusted platform. Trusted platforms for organizations and individuals –Trust models and the E-Commerce domain.

UNIT V:

E-Commerce Security: SET for E-Commerce Transactions, Business requirements for SET, SET System Participants, Dual Signature and Signature, Authentication and Message Integrity, Payment Processing. Secure Internet Programming, Security development life cycle, Internet Security Standards and Internet Security Products, Trusted Internet Security services.

Text Books:

1. Kenneth C. Laudon and Carol Guercio Trave, “E-Commerce Business Technology Society”, Pearson Education, 2005.
2. Pauline Ratnasingam, “Inter-Organizational Trust for Business-to-Business E-Commerce”, RMPress, 2005.

3. Siani Pearson, et al, "Trusted Computing Platforms: TCPA Technology in Context", Prentice Hall PTR, 2002

References:

1. Ravi Kalakota and Andrew Whinston –"Frontiers Electronic Commerce", 2nd Edition, Addison-Wesley, 1999.
2. P.T.Joseph, S.J, "E Commerce", Prentice Hall, 2nd Edition, 2008
3. Anup Gosh, "E-Commerce Security and Privacy", 2001
4. Relevant research papers from the journals

(ENG32) PROFESSIONAL AND TECHNICAL COMMUNICATION
(Open Elective-1)

Introduction:

This course aims to offer students a practical approach to professional and technical communication; and to focus specifically on verbal and written communication. Additionally, the course is designed to build confidence and; group communication and public speaking competence. Each UNIT In the syllabus is devised so as to include a writing component as well as an oral component.

Course Objectives:

- To **enable** the students to write without errors in spelling, mechanics, grammar and punctuation; resume, business letters, proposals and reports to accomplish academic as well as professional goals.
- To **train** students to write clearly, cohesively, emphatically and concisely.
- To **groom** students to speak accurately and fluently and prepare them for real world activities
- To **train** students in soft skills through group discussion to improve their EQ.

Course Outcomes:

Students will be able to:

- **Analyze** communication situations and audiences to make choices about the most effective and efficient way to communicate and deliver messages
- **Write** resume, business letters, project proposals and reports
- **SPEAK** fluently and address a large group of audience and participate in discussions.
- **NAVIGATE** through complex environments through interpersonal and collaborative skills.

UNIT I:

- Oral Communication :Self-introduction
- Applications and Covering letters
- Resume Writing
- Job Interviews

UNIT II:

- Oral Communication: Impromptu Speech
- Reading Business and Technical Texts
- Writing E-mails
- Writing Business Letters and Business Memos

UNIT III:

- Oral Communication: Group Discussions
- Summarizing and Synthesising
- Writing Abstracts

UNIT IV:

- Oral Communication : Debate
- Writing Business Proposals
- Writing Technical Proposals

UNIT V:

- Oral Communication: Making Presentations
- Interpreting Graphic Information

- Writing Business Reports
- Writing Technical Reports

Text and Materials:

1. Ashraf Rizvi, M (2005). Effective Technical Communication, Tata Mc Graw Hill Publishing Company Limited, New Delhi.
2. M. Raman and S. Sharma, Technical Communication: Principles and Practices, OUP, 2004. (Indian Edition)

References:

1. William S. Pfeiffer, (2012) Technical Communication: A Practical Approach (7th ed.) Longman
2. Burnett, Rebecca. Technical Communication. 5th Ed., Heinle, 2001.
3. Gerson Sharon J. and Steven Gerson : Technical Writing Process and Product. 3rd edition, New Jersey: Prentice Hall 1999
4. Markel, Mike. [Technical Communication: Situations and Strategies](#) (8th EDITION (2006-2007)
5. R. C. Sharma and K. Mohan, Business Correspondence and Report Writing, Third Edition, TMH, 2002. (Indian Edition)
6. Anderson, Paul V. (2003). Reports. In Paul V. Anderson's Technical Communication: A Reader-Centered Approach (5th ed..) (pp. 457-473). Boston: Heinle.

(SWE51) JAVA AND WEB TECHNOLOGIES & OOM LABORATORY

Course Objectives:

- Identify the requirements in use-case driven approach and specify the behavioural aspects exhibited in each use case for a given case study.
- Classify various classes, relationships, their responsibilities to be carried out on collaborating, based on these interactions deployment of model for a given case study can be done.
- Formulate the transition of swings from AWT and the creation of both graphical applets and standalone applications, and the use of visual components in graphical user interface design.
- Infer SQL queries and Stored Procedures of database with internal connection and apply one of those technologies to retrieve data in fastest manner in the J2EE web applications using JSP.

Course Outcomes:

After completion of this course the student is able to:

- Represent users requirements using the artifacts of UML (use case, interaction diagrams) and design the activity diagram and state diagram for a given case study.
- Develop the class and the component diagrams and finally summarises all the above artifacts to deploy the model using the deployment diagram for a given case study.
- Combine XML vocabularies and assess and evaluate the role of XML for the management and delivery of electronic information
- Explore the intricacies of JavaScript for creation of dynamic effects, validate form input entry on web sites and develop appropriate client-side applications

List of Sample Problems/Experiments:

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

Home page, Registration and user Login

User Profile Page, Books catalog

Shopping Cart, Payment By credit card

Order Conformation

**2. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

**3. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

**4. Bean Assignments

a. Create a Java Bean which gives the exchange value of INR(Indian Rupees) into equivalent American/ Canadian/Australian Dollar value.

b. Create a simple Bean with a label - which is the count of number of clicks. Then create a BeanInfo class

such that only the "count" property is visible in the Property Window.

**5. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web page using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

**6. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

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

NOTE: * - Simple Problems.

** - Moderate Problems.

*** - Complex Problems.

The student should take up the case studies of ATM system, Online Shopping System and Model it in different views i.e. Use case view, logical view, component view, Deployment view.

8. Design a Use case Diagram for ATM system, Online Shopping System.

9.a. Design a Sequence Diagram for ATM system, Online Shopping System.

9.b. Design a Collaboration Diagram for ATM system, Online Shopping System.

10. Design a Activity Diagram for ATM system, Online Shopping System.

11. Design a State Chart Diagram for ATM system, Online Shopping System.

12. Design a Class Diagram for ATM system, Online Shopping System.

13.a. Design a Component Diagram for ATM system, Online Shopping System.

13.b. Design a Deployment Diagram for ATM system, Online Shopping System.

Text Books:

1. Java Server Programming for Professionals, 2nd Edition, Bayross and others, O'reilly,SPD, 2007.
2. JDBC, Servlets, and JSP .Black Book, K. Santosh Kumar, dreamtech.
3. Core Web Programming, 2nd Edition, Volume 1, M.Hall and L.Brown, PHJPTR.
4. Core Web Programming, 2nd Edition, Volume 2, M.Hall and L.Brown, PIIPTR.

References:

1. Core Java, Volume 1, Ilorstan and Cornell, 8th Edition, Pearson Education, 2008.
2. Core Java, Volume 2, Ilorstan and Cornell, 8th Edition, Pearson Education, 2008.
3. The Unified Modeling Language Reference Manual by James Rumbaugh, Ivar Jacobson, and Grady Booch.
4. Object Oriented Analysis and Design ,J.W. Schmidt, F. Matthes, TU Hamburg-Harburglex&yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly.

(SWE04) DATA ANALYTICS

Course Objectives:

- To explore the fundamental concepts of big data analytics
- To learn to analyze the big data using intelligent techniques.
- To understand the applications using Map Reduce Concepts.
- Explore on Big Data applications Using Pig and Hive
- To understand the various search methods and visualization techniques.

Course Outcomes:

After completion of this course the student is able to:

- Work with big data platform and analyze the big data analytic techniques for useful business applications
- Design efficient algorithms for mining the data from large volumes.
- Learn to use various techniques for mining data stream.
- Analyze the Hadoop and Map Reduce technologies associated with big data analytics

UNIT I:

Introduction to Big Data: Introduction to Big Data Platform – Challenges of Conventional Systems - Web Data – Evolution of Analytic Scalability - Analytic Processes and Tools - Analysis vs Reporting - Modern Data Analytic Tools - Statistical Concepts: Sampling Distributions - Re-Sampling - Statistical Inference - Prediction Error.

UNIT II:

Data Analysis: Regression Modeling - Multivariate Analysis - Bayesian Modeling - Inference and Bayesian Networks - Support Vector and Kernel Methods - Analysis of Time Series: Linear Systems Analysis - Nonlinear Dynamics - Rule Induction - Neural Networks: Learning And Generalization - Competitive Learning - Principal Component Analysis and Neural Networks - Fuzzy Logic: Extracting Fuzzy Models from Data - Fuzzy Decision Trees - Stochastic Search Methods.

UNIT III:

Mining Data Streams, Frequent Itemsets and Clustering: Introduction To Streams Concepts – Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window Mining Frequent Itemsets - Market Based Model – Apriori Algorithm – Handling Large Data Sets in Main Memory – Limited Pass Algorithm – Counting Frequent Itemsets in a Stream – Clustering Techniques – Hierarchical – K-Means

UNIT IV:

HADOOP: History of Hadoop- The Hadoop Distributed File System – Components of Hadoop- Analyzing the Data with Hadoop- Scaling Out- Hadoop Streaming- Design of HDFS-Java interfaces to HDFSBasics- Developing a Map Reduce Application-How Map Reduce Works-Anatomy of a Map Reduce Job run-Failures-Job Scheduling-Shuffle and Sort – Task execution - Map Reduce Types and Formats- Map Reduce Features

UNIT V:

Frameworks and Visualization: Applications on Big Data Using Pig and Hive – Data processing operators in Pig – Hive services –HiveQL – Querying Data in Hive - fundamentals of HBase and ZooKeeper Visualizations - Visual Data Analysis Techniques - Interaction Techniques

Text Books:

1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.
3. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley & Sons, 2012.
4. Tom White "Hadoop: The Definitive Guide" Third Edition, O'reilly Media, 2012.

References:

1. Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons, 2007
2. Pete Warden, "Big Data Glossary", O'Reilly, 2011.
3. Jiawei Han, Micheline Kamber "Data Mining Concepts and Techniques", Second Edition, Elsevier, Reprinted 2008

(SWE05) WEB SERVICES AND SERVICE ORIENTED ARCHITECTURE

Course Objectives:

- To define and design applications as combinations of services, and be able to discuss the emergent properties of those compositions;
- Understand concepts, technology and design of service orientation and web services
- Analyzing and designing business based on SOA principles.
- Understand Web Service Specifications and asses support by platforms like J2EE and .NET

Course Outcomes:

After completion of this course the student is able to:

- **Describe** SOA and Web Service Fundamentals and **build** Services with WS-* Specifications.
- **Assess and realize** the service Layer Abstraction.
- **Analyze and Design** the building an SOA.
- **Examine** SOA support provided by J2EE and .NET platform

UNIT I:

Evolution of Distributed Computing: Core distributed computing technologies-client/server, CORBA, JAVA RMI, Microsoft DCOM, MOM, Challenges in Distributed Computing

Introduction to Web Services: The definition of web services, basic operational model of web services, Core Web Service Standards, benefits and challenges of using web services

SOA and Web Services Fundamentals

Introducing SOA: Fundamental SOA, Common Characteristics of Contemporary SOA, Common tangible benefits of SOA, Common pitfalls of adopting SOA.

The Evolution of SOA: An SOA timeline, The continuing evolution of SOA, The roots of SOA.

Web Services and Primitive SOA: The Web Services frame work, Services, Service Descriptions, and Messaging.

UNIT II:

SOA and WS-* Extensions

Web Services and Contemporary SOA (Part-I): Message Exchange Patterns, Service Activity Coordination, Atomic transactions, Business Activities, Orchestration, and Choreography.

Web Services and Contemporary SOA (Part-II): Addressing, Reliable Messaging, Correlation, Policies, Metadata Exchange, Security, Notification and Eventing.

UNIT III:

SOA and Services - Orientation

Principles of Service-Oriented: Service - Orientation and the Enterprise, Anatomy of SOA, Common Principles of Service – Orientation, Interrelation between Principles of Service-Oriented, Service Orientation and Object Orientation, Native Web Services support for Principles of Service-Oriented. **Service Layers:** Service-Oriented and contemporary SOA, Service Layer abstraction, Application Service Layer , Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration Scenarios.

UNIT IV:

Building SOA (Planning and Analysis)

SOA Delivery Strategies: SOA delivery lifecycle phases, the top-down strategy, the bottom-up strategy, the agile strategy.

Service-Oriented Analysis (Part I-Introduction): Introduction to Service Oriented Analysis, Benefits of a Business-Centric SOA, Deriving Business Services.

Service Oriented Analysis (Part-II-Service Modelling): Service Modelling, Service Modelling guidelines, Classifying Service model logic, Contrasting Service modelling Approaches.

UNIT V:

Building SOA (Technology and Design)

Service Oriented Design (Part I-Introduction): Introduction to Service-Oriented Design, WSDL related XML Schema language basics. WSDL language basics, SOAP language basics, Service interface design tools.

Service Oriented Design (Part II-SOA Composition Guidelines): SOA Composing steps, Considerations for choosing service layers, Considerations for positioning core SOA standards, Considerations for choosing SOA extensions.

Service Oriented Design (Part III - Service Design): Service Design overview, Entity-centric business Service Design, Application Service Design, Task-centric business Service Design, Service Design guidelines.

Service Oriented Design (Part IV-Business Process Design): WS-BPEL language basics, WS-Coordination overview, Service Oriented Business process Design.

Fundamental WS-Extensions: WS-Addressing language basics, WS-Reliable Messaging language basics, WS-Policy language basics, WS-Metadata Exchange language basics, WS-Security language basics.

SOA Platforms: SOA platform basics, SOA support in J2EE and .NET, Integration considerations.

Text Books:

1. Service-Oriented Architecture: Concepts, Technology & Design, Thomas Erl, Pearson Edu.
2. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India.

References:

1. Web Services & SOA Principles and Technology, Second Edition, Michael P.Papazoglou
2. The Definitive guide to SOA, Jeff Davies & others, Apress, Dreamtech.
3. Understanding SOA with Web Services, Eric Newcomer, Greg Lomowand Pearson Education.
4. Java SOA Cook book, E. Hewitt, SPD.
5. SOA in Practice, N.M. Josuttis, SPD.

VNR Vignana Jyothi Institute of Engineering and Technology			
I Year – II Sem. M.Tech (Software Engineering)	L	P	C
	3	1	4

(SWE06) SOFTWARE QUALITY ASSURANCE AND TESTING

Course Objectives:

- Define Quality and understand various quality standards
- Recognize metrics, measurements and methodology for assessing software quality.
- Understand steps involved in testing process and identify the appropriate testing technique.
- State steps involved in testing process and testing specialized systems

Course Outcomes:

After completion of this course the student is able to:

- Identify and analyse the importance of Software Quality Assurance process and Standards
- Demonstrate Quality Metrics and it's Methodology.
- Illustrate test documentation policies and compare different testing techniques
- Outline testing process of specialized systems

UNIT I:

Software Quality Assurance Framework and Standards SQA Framework: What is Quality? Software Quality Assurance, Components of Software Quality Assurance –

Software Quality Assurance Plan: Steps to develop and implement a Software Quality Assurance Plan -

Quality Standards: ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, 6 Sigma

UNIT II:

Software Quality Assurance Metrics and Measurement Software Quality Metrics: Product Quality metrics, In-Process Quality Metrics, Metrics for Software Maintenance,

Software Quality metrics methodology: Establish quality requirements, Identify Software quality metrics, Implement the software quality metrics, analyse software metrics results, validate the software quality metrics - Software quality indicators

UNIT III:

Software Testing Strategy and Environment: Establishing testing policy, structured approach to testing, test factors, Economics of System Development Life Cycle (SDLC) Testing

Software Testing Methodology: Defects hard to find, verification and validation, functional and structural testing, workbench concept, eight considerations in developing testing methodologies, testing tactics checklist

UNIT IV:

Software Testing Techniques: Black-Box, Boundary value, Bottom-up, Branch coverage, Cause-Effect graphing, CRUD, Database, Exception, Gray-Box, Histograms, Inspections, JADs, Pareto Analysis, Prototyping, Random Testing, Risk-based Testing, Regression Testing, Structured Walkthroughs, Thread Testing, Performance Testing, White-Box Testing

Software Testing Tools: Taxonomy of Testing tools. Methodology to evaluate automated testing tools, Load Runner, QTP and Rational Testing Tools, Silk test, Java Testing Tools, JMetre.

UNIT V:

Testing Process:

Eleven Step Testing Process: Overview

Testing Specialized Systems and Applications: Testing Client/Server Web applications. Testing off the Shelf Components, Testing Security, Testing a Data Warehouse

Text Books:

1. Software Testing and continuous Quality Improvement, by William E.Lewis, Gunasekaran Vcerapillai, Second Edition, Auerbach Publications

2. Effective Methods for Software Testing, 2nd Edition, William E. Perry , Second Edition, Wiley India, 2006.
3. Metrics and Models for Software Quality Engineering, by Stephen H. Kan, by Pearson Education Publication
4. Software Testing Tools, K.V.K.K. Prasad, Dream tech press, 2008.

References:

1. Testing and Quality Assurance for Component-based Software, by Gao, Tsao and Wu, Artech House Publishers
2. Software Testing Techniques, by Boris Beizer, Second Edition, Dreamtech Press
3. Managing the Testing Process, by Rex Black, Wiley
4. Handbook of Software Quality Assurance, by G. Gordon Schulmeyer, James I. McManus, Second Edition, International Thomson Computer Press

(SWE21) ADVANCED SOFTWARE ENGINEERING PRINCIPLES AND PRACTICES
(Elective – 3 & 4)

Course Objectives:

- Students are capable of modelling and simulating the real software problems to a certain extent.
- The real time aspects of software design are discussed there by motivating the students to face real world challenges.
- Apply a systematic, disciplined, quantifiable approach to the cost-effective development, operation and maintenance of software systems to the satisfaction of their beneficiaries.
- Build solutions using different technologies, architectures and life-cycle approaches in the context of different organizational structures.

Course Outcomes:

After completion of this course the student is able to:

- The system engineering hierarchy can be analyzed and designed after studying the principles.
- Real time constraints in developing system models are understood and solved.
- An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, safety, and sustainability
- Identify and analyze the software engineering principles to develop a model

UNIT I:

System Engineering: System Engineering: Computer based systems, system engineering hierarchy, Information engineering, Information strategy planning, business area analysis, product engineering, modelling the system architecture, system modelling and simulation, system specification. Computer Based System Engineering: Emergent system properties, systems and their environment, system modelling, system engineering process, system procurement.

UNIT II:

Modern Design Concept and Principles: Design Concepts: Mapping of analysis model to design model, design process, design principles, design concepts, effective modular design, design model, design heuristics, design specification. Architectural Design Process: Transform mapping and transaction mapping, design post processing, interface design, Human computer interface design, and interface design guidelines, procedural design

UNIT III:

Real Time Software Design: Real-time systems, definition, System consideration, Real time system analysis, stimulation / Response systems, Real-time System model, system elements, Real-time programming, system design, Real-time system modelling, RTOS, process priority, process management, scheduling strategy, RT-Systems, design process, monitoring and control system, Generic architecture, data acquisitions systems

UNIT IV:

Component Based Development CBSE: Component based software engineering, components and component models; Component based software engineering process, Component development for reuse, Component based software engineering process, Component Composition.

UNIT V:

Software Reuse: Management issues, Reuse process, process model, Domain engineering, Domain analysis process, Building Reusable Components, Classifying and retrieving components, Economics of software reuse.

Text Books:

1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", - 4th edition, McGraw Hill Publications.
2. Ian Sommerville, "Software Engineering", - 6th / 7th edition – Pearson Education Publications.

References:

1. Shari Lawrence Pfleeger, "Software Engineering Theory and Practices", - 2nd Edition.
2. John W. Satzinger, Robert B Jackson, Stephen D Burd, "System Analysis and Design in Changing World", Thomson Course Technology.
3. Richard Murch, Tony Johnson, "Intelligent Software agents". Prentice Hall

(SWE22) SOFTWARE METRICS
(Elective – 3 & 4)

Course Objectives:

- To be aware of Core metrics for product, quality, process
- To familiarize with the concepts of Software quality and tools for quality metrics
- To learn more about software liability
- To understand important concepts of complexity metrics and OO metrics.

Course Outcomes:

After completion of this course the student is able to:

- Be aware of connections of software engineering measurements with the multidisciplinary history of measurement theory.
- Be aware of the social and ethical issues associated with human performance measurement .
- Assess the quality of a proposed metric.
- Understand the commercial and organizational contexts of any metric.

UNIT I:

The History and Evolution of Software Metrics: Evolution of the software industry and evolution of software measurements – The cost of counting function point metrics – The paradox of reversed productivity for high-Level languages- The Varieties of functional metrics – Variations in application size and productivity rates – Future Technical Developments in Functional Metrics- Software measures and metrics not based on function points.

UNIT II:

Measuring Software Quality: Quality control and international competition – Defining quality for measurement and estimation – Five steps to software quality control- Measuring software defect removal- Measuring Defect removal efficiency – Measuring the costs of defect removal – Evaluating defect prevention methods – Measuring customer reported defects- Measuring invalid defects, The Rayleigh Model- Reliability Growth Models.

UNIT III:

Process Metrics: In-Process Metrics for Software Testing - Test Progress S Curve - Testing Defect Arrivals Over Time - Product Size Over Time - CPU Utilization - Effort/Outcome Model. Complexity Metrics and Models - Lines of Code - Halstead's Software Science - Cyclomatic Complexity. - Syntactic Constructs - Structure Metrics. Metrics for Object-Oriented Projects - Concepts and Constructs - Design and Complexity Metrics - Lorenz Metrics and Rules of Thumb - CK OO Metrics Suite - Productivity Metrics.

UNIT IV:

Mechanics of Measurement: Software Assessments – Software Baselines – Software Benchmarks- What a Baseline analysis covers – Developing or Acquiring a baseline data collection Instrument – Administering the data collection questionnaire – Analysis and aggregation of the Baseline data. Measuring and Analyzing Customer Satisfaction - Surveys - Data Collection - Sampling Methods - Analyzing Satisfaction Data. Conducting In-Process Quality Assessments - Preparation - Evaluation - Quantitative Data - Qualitative Data - Evaluation Criteria - Overall Assessment.

UNIT V:

Measurements, Metrics and Industry Leadership: Measures and metrics of industry leaders – Measures, metrics and innovation – Measurements, metrics and outsource litigation – Measurements, metrics and behavioral changes – Commercial software measurement tools. Measuring Process Maturity - Process Capability - Value of Process Improvement - Process Adoption – Process Compliance. Function Point Metrics to Measure Software Process Improvement - Software Process Improvement Sequences.

Text Books:

1. Caper Jones, “Applied Software Measurement: Global Analysis of Productivity and Quality” , Third Edition, McGraw Hill Companies, 2008.
2. Stephen H. Kan, “Metrics and Models in Software Quality Engineering”, Addison Wesley, 2011.

References:

1. Mark Lorenz, Jeff Kidd, “Object-Oriented Software Metrics”, Prentice Hall, 2000.
2. Naresh Chauhan, “Software Testing Principles and Practices”, Oxford University Press, 2010.

(SWE23) SOFTWARE REENGINEERING
(Elective – 3 & 4)

Course Objectives:

- Understand and analyze the importance and basic concepts of software reengineering.
- Identify the standards in software reengineering.
- Describe the concepts of organizing and reorganizing process.
- Analyze the need of Software Reuse Tools

Course Outcomes:

After completion of this course the student is able to:

- Illustrate the importance of software reengineering.
- Outline the standards in software reengineering.
- Use the concepts of organizing and reorganizing process in present industry.
- Compare and Contrast various Software Reuse Tools.

UNIT I:

Introduction: The Need, What is reuse, Types of reuse, Domain Analysis, Hypertext

Software Lifecycle: Requirements, Design, Implementation, Testing and Documentation, Maintenance, Standards

Management: Software Team Organization, Process Modelling, Software Project Modelling, Scheduling

UNIT II:

Reengineering Concepts:

Reengineering Objectives, Reengineering Concepts, General Models of Software Reengineering, Software Reliability Methods, Dealing with Software Reliability, Reengineering Phases and Tasks, Reengineering Implementation

UNIT III:

Enterprise & Standards:

Reuse Framework: Process Idioms and Sources, Reuse management, Asset creation, Asset management, Asset Utilization, Cost Vs Benefits, legal Issues.

Standards: Expectations, Existing Related Standards, Recommendations, Conclusion.

UNIT IV:

Organize to Reorganize:

Organizing: Indexing, Document Outlines, Domain Models, Code Organizing, Framework

Retrieving: Retrieval specification, Document Retrieval, Program Retrieval, Retrieval systems, Monitoring Retrieval.

Reorganizing: Retrieved Component suitability, Document reorganizing, Program reorganizing, Code generators, Testing and Reuse.

UNIT V:

Practical Examples:

Software Reuse tools: CASE, Practitioner, MUCH, Softclass, User Interface Generator.

Case Studies: IBM Reuse and Boblingen Experience, HP Reuse, Motorola Reuse.

Courseware Reuse: Courseware Standards, Small Company, Coordination, Education Object Economics

Text Books:

1. Reengineering Software –How to Reuse Programming to Build New, State-of-the-Art Software, 2nd edition by Roy Rada, Eric Dobby Publishing.
2. Application Software Reengineering by Afshar Alam, Tendai padenga, Pearson Education India, 2010.

References:

1. Secrets of Reverse engineering, Eldad Eilal, Willy Publishing.
2. Advancement in Software Maintenance Management:Technologies and Solutions, Macario Polo, Ed.Idea Group Publishing,2003.
3. Software Reengineering by Robert S.Arnold. IEEE Comp. Society.

VNR Vignana Jyothi Institute of Engineering and Technology			
I Year – II Sem. M.Tech (Software Engineering)	L	P	C
	3	0	3

(SWE24) RESEARCH METHODOLOGY
(Elective – 3 & 4)

Course Objectives:

- Identify an appropriate research problem in their interesting domain.
- Organize and conduct research project.
- Prepare a research project thesis report.

Course Outcomes:

After completion of this course the student is able to:

- Identify and implement the activities according to the research process.
- Select and apply the appropriate research design for a problem.
- Apply the estimation techniques, Chi-Square test and ANOVA technique.
- Prepare Research thesis report and proposals in specific domains

UNIT I:

Introduction: Research objective and motivation, Types of research, Research approaches, Significance, Research method vs. methodology, Research process

UNIT II:

Formulating a Research Problem: Literature review, Formulation of objectives, Establishing operational definitions, Identifying variables, Constructing hypotheses

UNIT III:

Research Design and Data Collection: Need and Characteristics, Types of research design, Principles of Experimental research design, Method of data collection, Ethical issues in collecting data

UNIT IV:

Sampling and Analysis of Data: Need of Sampling, Sampling distributions, Central limit theorem, Estimation: mean and variance, Selection of sample size Statistics in research, Measures of Central tendency, Dispersion, asymmetry and relationships, Correlation and Regression analysis, Displaying data

UNIT V:

Hypothesis Testing: Procedure, Hypothesis testing for difference in mean, variance limitations, Chi-square test, Analysis of variance (ANOVA), Basic principles and techniques of Writing a Research proposal

Text Books:

1. R. C. Kothari, Research Methodology: Methods and Techniques, 2nd edition, New Age International Publisher, 2009
2. Ranjit Kumar, Research Methodology: A Step-by-Step Guide for Beginners, 2nd Edition, SAGE, 2005

References:

1. Trochim, William M. The Research Methods Knowledge Base, 2nd Edition. Internet WWW page, at URL: <<http://www.socialresearchmethods.net/kb/>> (version current as of October 20, 2006).
2. (Electronic Version): StatSoft, Inc. (2012). Electronic Statistics Textbook. Tulsa, OK:StatSoft. WEB: <http://www.statsoft.com/textbook/>. (Printed Version): Hill, T. & Lewicki, P. (2007). STATISTICS: Methods and Applications. StatSoft, Tulsa, OK.

(SWE25) INTERNET OF THINGS
(Elective – 3 & 4)

Course Objectives:

- Introduce the current vision of the Internet of Things and its impact on the world.
- To provide an appreciation for the standardization of IoT protocols that is necessary for IoT to become reality.
- Implement IoT applications in real time scenario.

Course Outcomes:

After completion of this course the student is able to:

- Establish knowledge in a concise manner how the Internet of things work.
- Identify and interpret design methodology of IoT platform.
- Exhibit the knowledge of interfacing Python with embedded board- Raspberry Pi.
- Illustrate the Networking model of IoT

UNIT I:

Introduction to Internet of Things: Introduction, physical design of IoT, logical design of IoT-functional blocks, communicational models, communication APIs, IoT enabling technologies, IoT levels, deployment templates.

UNIT II:

Domain Specific IoTs: Introduction, home automation, cities, environment, energy, retail, logistics, agriculture, industry.

UNIT III:

Developing Internet of Things: Introduction, IoT design methodology, Case Study on IoT System for Weather Monitoring, Motivation for using Python.

UNIT IV

Hardware and Software for IoT: Logical design using Python-data types, control flow, functions, packages, file handling, classes, Python packages of Interests for IoT

UNIT V:

IPv6 for Smart Object Networks and the Internet of the Things: Introduction, The depletion of the IPv4 address space, NAT : A solution to IPv4 Address Exhaustion, Architectural discussion.

Text Books:

1. Arshdeep Bahga, Vijay Madiseti, Internet of Things: A hands on approach, paperback-2014.
2. Jean- Philippe Vasseur, Adam Dunkels , Interconnecting Smart Objects with IP :The Next Internet . paperback-Import,2010

References:

1. Adrian Mcewen, Hakin Cassimally Designing the Internet of Things Paperback – 25 Jul 2015.
2. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols Hardcover – Import, 6 Jan 2012.
3. Keoh, Sye Loong, Sahoo Subhendu Kumar, and Hannes Tschofenig. "Securing the internet of things: A standardization perspective." *Internet of Things Journal, IEEE* 1.3 (2014): 265-275.
4. Ortiz, Antonio M., et al. "The cluster between internet of things and social networks: Review and research challenges." *Internet of Things Journal, IEEE* 1.3 (2014): 206-215.

(SWE26) SOFTWARE DEFINED NETWORKS
(Elective – 3 & 4)

Course Objectives:

- This course provides a comprehensive introduction to Software Defined Networking (SDN) and presents SDN in context with more familiar network services and challenges
- It also offers a unique perspective of the business case and technology motivations for considering SDN solutions.
- It Identifies the impact of SDN on traffic management and the potential for network service growth
- Instills the knowledge needed to manage current and future demand and provisioning for SDN
- It provides students with the basic concepts and explains the importance of virtualization, particularly the impact of virtualization on servers and networks

Course Outcomes:

After completion of this course the student is able to:

- Articulate the main concepts, Virtual and Physical Memory Mapping and Software Defined Networking.
- Identify the design, implementation of SDN and Service providers of SDN.
- Explain the core issues of Openflow such as Switch Specification, ports, and packet processing pipeline.
- Demonstrate Enterprise Networks ,SDN and Transport Networks, SDN and Optical Transport Networks

UNIT I:

Introduction to Software Defined Networking: Virtualization, Virtual Memory, Virtual Memory Operation, Virtual and Physical Memory Mapping, Server Virtualization, Storage Virtualization, Software Defined Networking, Network Limitations, Network Control Plane.

UNIT II:

SDN Implementation: Introduction, SDN Implementation, SDN Design, Separation of the Control and Data Planes, Edge-Oriented Networking, SDN Operation, Service Providers and SDN

UNIT III:

Openflow: Introduction, Overview of the OpenFlow Switch Specification,OpenFlow Ports,OpenFlow Packet-Processing Pipeline,OpenFlow Channel, Message Handling,OpenFlow Channel Connections, Controller Modes, Auxiliary Connection Use for Performance and Reliability, Flow Table Synchronization, Bundle Messages,OpenFlow Configuration-and-Management Protocol, Remote Configuration and The OpenFlow Conformance Testing Program

UNIT IV:

SDN Controllers: Network Programmability, The Management Interface, The Application-Network Divide Modern Programmatic Interfaces, Virtualization and Data Plane I/O, Services Engineered Path Service Locations and Chaining.

UNIT V:

SDN Evolution: Introduction, SDN and Enterprise Networks, SDN and Transport Networks, SDN and Optical Transport Networks, Increasing WAN Utilization with SDN,SDN Scalability Issues, Controller Designs for Scalability, Potential SDN Scalability Issues, Network Types, SDN Management ,Load Adaptation.

Text Books:

1. Software Defined Networking: Design and Deployment, 1st Edition Patricia A. Morreale and James M. Anderson CRC press.
2. SDN: Software Defined Networks Thomas D. Nadeau and Ken Gray Orielly media

References:

1. Software Defined Networking with OpenFlow by Siamak Azodolmolky wiley Publications
2. Software Defined Networks: A Comprehensive Approach by Paul Goransson, Chuck Black Publisher Morgan Kaufmann
3. Software Quality, Martin Wiec/orek & Dirk Meyerhoff, Springer, 2001.
4. Software Testing, Srinivasan Desikan & Gopaldaswamy Ramesh, Pearson Education, 2006.
5. Software Quality, Mordechai Ben-Menachem/Garry S. Marliss. Thomson Learning publication, 1997

(SWE41) ADVANCED UNIX PROGRAMMING
(Open Elective - 2)

Course Objectives:

- Apply the concepts of files and Directories to manage the Linux Environment.
- Provide knowledge in working with the core operating systems Concept, Signals in Linux Environment
- Design the Inter process communication by using the IPC techniques
- Build Client-Server Environment using sockets.

Course Outcomes:

After completion of this course the student is able to:

- Analyse the files and directories in Linux environment.
- Implement system programs to control the processes using signals.
- Develop programs to provide Inter process communication to avoid classical IPC problems.
- Design a client-server application using sockets and RMI

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 Network Programming ,W.R. Stevens, PHI.(Units II,III,IV)
3. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.(UNIT I)
4. An Introduction to Network Programming with Java, Jan Graba, Springer, rp 2010.(Unit V)

References:

1. Java Network Programming,3rd edition, E.R. Harold, SPD, O'Reilly.(Unit V)
2. Linux System Programming, Robert Love, O'Reilly, SPD.
3. Advanced Programming in the UNIX environment, 2nd Edition, W.R.Stevens, Pearson Education.
4. UNIX for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education.
5. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones, Wrox, Wiley India Edition.

(CNS41) ANDROIDAPPLICATION DEVELOPMENT
(Open Elective - 2)

Course Objectives:

- Mobile Application Development course exposes the students to essentials of mobile apps development.
- The core modules of this subject include designing, developing, testing, signing, packaging and distributing high quality mobile apps.
- This course aims to teach mobile app development using Android as the development platform.

Course Outcomes:

After completion of this course the student is able to:

- Appreciate the Mobility landscape
- Familiarize with Mobile apps development aspects
- Design and develop mobile apps, using Android as development platform, with key focus on user experience design, native data handling and background tasks and notifications.
- Perform testing, signing, packaging and distribution of mobile apps

UNIT I:

Introduction to Mobile

A brief history of Mobile, The Mobile Ecosystem, Why Mobile?, Types of Mobile Applications, Mobile Information Architecture, Mobile Design, Mobile 2.0, Mobile Web development, Small Computing Device Requirements.

UNIT II:

Introduction to Android: History of Mobile Software Development, The Open Handset Alliance, The Android Platform, Android SDK, Eclipse Installation, Android Installation, Building a Sample Android application

UNIT III:

Android Application Design Essentials: Anatomy of an Android applications, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions, Managing Application resources in a hierarchy, Working with different types of resources

UNIT IV:

Android User Interface Design Essentials: User Interface Screen elements, Designing User Interfaces with Layouts, Drawing and Working with Animation

UNIT V:

Using Common Android APIs: Using Android Data and Storage APIs, Managing data using SQLite, Sharing Data between Applications with Content Providers, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, Deploying Android Application to the World

Text Books

1. J2ME: The Complete Reference, James Keogh, Tata McGrawHill
2. Lauren Darcey and Shane Conder, "Android Wireless Application Development", Pearson Education, 2nd ed. (2011)

References:

1. Reto Meier, "Professional Android 2 Application Development", Wiley India Pvt Ltd

2. Mark L Murphy, "Beginning Android", Wiley India Pvt Ltd
3. Sayed Y Hashimi and Satya Komatineni, "Pro Android", Wiley India Pvt Ltd
4. developer.android.com (web)
5. Android Application Development All in one for Dummies by Barry Burd, Edition: I

(SWE42) SOFT SKILLS
(Open Elective - 2)

Course Objectives:

- Develop inter personal skills and be an effective goal oriented team player.
- Produce professionals with idealistic, practical and moral values.
- Emphasize problem solving skills and communication.
- Reconstruct attitude and understand its influence on behavior.

Course Outcomes:

After completion of this course the student is able to:

- Express what Soft Skills is
- Discover the significance of soft skills in the working environment
- Employ leadership skills and teamwork skills
- Plan and schedule the time and stress management

UNIT I:

Self Analysis: SWOT Analysis, Who am I, Attributes, Importance of Self Confidence, Self Esteem

UNIT II:

Attitude: Factors influencing Attitude, Challenges and lessons from Attitude

Change Management: Exploring Challenges, Risking Comfort Zone, Managing Change

UNIT III:

Motivation: Factors of motivation, self talk, Intrinsic & Extrinsic Motivators.

Goal Setting: Wish List, SMART Goals, Blue print for success, Short Term, Long Term, Life Time Goals

UNIT IV:

Time Management: Value of time, Diagnosing Time Management, Weekly Planner To do list, Prioritizing work.

Stress Management: Causes of Stress and its impact, how to manage & distress, Understanding the Circle of control, Stress Busters.

UNIT V:

Creativity: Out of box thinking, Lateral Thinking Presentation

Leadership: Skills for a good Leader, Assessment of Leadership Skills

Team Work: Necessity of Team Work Personally, Socially and Educationally

Text Books:

1. Nitin Bhatnagar, Mamata Bhatnagar, "Effective Communication and Soft Skills", Pearson Publications.
2. Barun K. Mithra "Personality Development and Soft Skills", Oxford Publications.

References:

1. INSIGHT, 2012, Career Development Centre, SRM Publications.
2. Covey Sean, "Seven Habits of Highly Effective Teens", New York, Fireside Publishers, 1998.
3. Carnegie Dale, "How to win Friends and Influence People", New York: Simon & Schuster, 1998.
4. Thomas A Harris, "I am ok, You are ok", New York-Harper and Row, 1972.
5. Daniel Coleman, "Emotional Intelligence", Bantam Book, 2006.

(SWE52) SOFTWARE TESTING, CASE TOOLS AND ANALYTICS LABORATORY

Course Objectives:

- Discuss the overview of data analytics and interpret the data analytics life cycle.
- Illustrate the various data analytic methods using R.
- Define the process of test documentation.
- Identify the usage of automated testing tools.

Course Outcomes:

After completion of this course the student is able to:

- Analyze and differentiate the analytical methods using R
- Apply advanced analytic techniques in data mining.
- Generate various test documents.
- Identify and apply appropriate automated testing tool

Software Testing List of Experiments

1. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
2. Write the test cases for any known application (e.g. Banking application)
3. Create a test plan document for any application (e.g. Library Management System)
4. Study of any testing tool (e.g. QTP)
5. Study of any web testing tool (e.g. Selenium)
6. Study of any bug tracking tool (e.g. Bugzilla)
7. Study of any test management tool (e.g. Quality Centre)
8. 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.

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

Data Analytics List of Experiments

1. Data Analytics overview and Data Analytics life cycle.
2. Basic Data Analytic methods using R
 - a. Introduction to R – look at the data
 - b. Analysing and Exploring the Data
 - c. Statistics for Model Building and Evaluation
3. Apply clustering techniques to cluster the data.
4. Apply various Association rule mining algorithms.
5. Apply different classification techniques to classify the given data.

VNR Vignana Jyothi Institute of Engineering & Technology

I Year – II Sem. M.Tech (Software Engineering)	L	P	C
	0	0	4

(CSE62) MINI PROJECT – II

A mini project work shall be carried out on any topic of Software Engineering and a seminar should be given on the same along with a brief report.

VNR Vignana Jyothi Institute of Engineering & Technology

II Year – I Sem. M.Tech (Software Engineering)	L	P	C
	0	0	4

(SWE63) COMPREHENSIVE VIVA-VOCE

VNR Vignana Jyothi Institute of Engineering & Technology

II Year – I Sem. M.Tech (Software Engineering)	L	P	C
	0	0	8

(SWE71) INTERNSHIP / DISSERTATION PHASE- I

VNR Vignana Jyothi Institute of Engineering & Technology

II Year – II Sem. M.Tech (Software Engineering)	L	P	C
	0	0	18

(SWE72) DISSERTATION PHASE - II