

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CURRICULUM AND SYLLABI FOR M.E. SE COURSES

SEMESTER 1:

S.NO	CATEGORY	SUBJECT CODE	SUBJECT NAME	L	T	P	C	MAX MARKS
1	FCBS	15MM106	Operations Research	3	2	0	4	100
2	PC	15PQ401	Advanced Data Structures	3	2	0	4	100
3	PC	15PQ402	Advanced Operating Systems	3	2	0	4	100
4	PC	15PN401	Advances in Software Engineering	3	0	0	3	100
5	PE		PE – 1	3	0	0	3	100
6	PE		PE – 2	3	0	0	3	100
7	PC Lab	15PN451	Software Requirements & Design Laboratory	0	0	4	2	100
	Total			18	6	4	23	800

SEMESTER 2:

S.NO	CATEGORY	SUBJECT CODE	SUBJECT NAME	L	T	P	C	MAX MARKS
1	PC	15PQ404	Advanced Database Technology	3	2	0	4	100
2	PC	15PN402	Software Testing	3	2	0	4	100
3	PC	15PN403	Software Project Planning and Management	3	2	0	4	100
4	PE		PE - 3	3	0	0	3	100
5	PE		PE – 4	3	0	0	3	100
6	PSC		PSC – 1	3	0	0	3	100
7	PC Lab	15PN452	Database & Software Testing Laboratory	0	0	4	2	100
8		15PN801	Technical Seminar	0	0	2	1	100
	Total			18	4	6	24	800

FCBS - Foundation Compulsory Basic Science
 PC - Programme Core
 PSC - Programme Soft Core
 PE - Programme Elective

SEMESTER 3:

S.NO	CATEGORY	SUBJECT CODE	SUBJECT NAME	L	T	P	C	MAX MARKS
1	PSC		PSC – 2	3	0	0	3	100
2	PE		PE – 5	3	0	0	3	100
3	PE		PE – 6	3	0	0	3	100
4	Project	15PN901	Project Work & Viva-voce – Phase I	0	0	12	6	100
5		15PN802	Comprehensive Viva – Voce	0	0	2	1	100
	Total			9	0	14	16	500

SEMESTER 4:

S.NO	CATEGORY	SUBJECT CODE	SUBJECT NAME	L	T	P	C	MAX MARKS
1	Project	15PN902	Project Work & Viva-voce – Phase II	0	0	24	12	100
		Total		0	0	24	12	100

FCBS - Foundation Compulsory Basic Science

PC - Programme Core

PSC - Programme Soft Core

PE - Programme Elective

Programme Soft Core Group (PSC):

S.No	SUBJECT CODE	SUBJECT NAME
1	15PN501	Software Reuse
2	15PN502	Software Agents
3	15PN503	Formal methods in Software Engineering
4	15PN504	Software Reliability
5	15PN505	Software Metrics & Quality Assurance

Programme Elective (PE) Groups:

S.No	GROUP NAME	SUBJECT CODE	SUBJECT NAME
1	Software Process and Design	15PN601	Personal and Team Software Process
		15PN602	Software Design Pattern
		15PN603	Software Risk Management and Maintenance
2	Software Architecture and Agile Software Development	15PN604	Pattern Oriented Software Architecture
		15PN605	Software Architecture & Design
		15PN606	Agile Software Process
3	Distributed Computing	15PQ607 / 15PN607	Distributed Computing
		15PQ608 / 15PN608	Service Oriented Architecture
		15PQ609 / 15PN609	XML and Web Services
4	Social Network Engineering	15PQ610 / 15PN610	Mobile Application Development
		15PQ611 / 15PN611	User Interface Design
		15PQ612 / 15PN612	Social Network Analysis
5	Data Analytics	15PQ613 / 15PN613	Data Warehousing & Mining
		15PQ614 / 15PN614	Internet of Things
		15PQ615 / 15PN615	Big Data Analytics

Note :

1. Programme Electives (PE) must be framed by having 5 domains, each possessing 3 subjects. Students should get specialized in any two or three domains.
2. List of subjects must be given in Programme Soft Core (PSC), so that students can choose any 2 subjects.
3. Students can earn extra credits by doing certification courses.

Curriculum Structure - Sample

S.No	Category Name	Actual Credit Break Up
1	Foundation Compulsory Basic Science (FCBS)	4
2	Programme Core(PC)	27
3	Programme Elective(PE)	18
4	Programme Soft Core(PSC)	6
5	Project	18
6	Technical Seminar	1
7.	Comprehensive Viva – Voce	1
Total		75

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ401	ADVANCED DATA STRUCTURES	3	2	0	4

Distribution of marks in Question Paper – Theory (70%) and Problem (30%)

1. Course objectives:

- To understand the implementation and use of advanced data structures
- To learn how to analyze the space and time requirements of a given algorithm.
- To design efficient algorithms using algorithmic techniques

2. Course pre-requisites : Data Structures and Algorithms, C++ Programming

Module – I FOUNDATIONS AND SORTING ORDER STATISTICS 9

Growth of Functions: Asymptotic notations – Standard notations and common functions - Divide-and-Conquer: The maximum-subarray problem – Strassen’s algorithm for matrix multiplication – solving recurrences:- substitution method, recursion-tree method. Probabilistic Analysis and Randomized Algorithms: The hiring problem – Indicator random variables - Randomized algorithms Heap sort- Quick sort - Sorting in Linear Time

Module – II DATA STRUCTURES 9

Elementary Data Structures: Stacks and queues - Linked lists - Implementing pointers and objects - Representing rooted trees. Hash Tables: Direct-address tables - Hash tables - Hash functions - Open addressing. Binary Search Trees: Querying a binary search tree - Insertion and deletion - Randomly built binary search trees .Red-Black Trees - Augmenting Data Structures

Module – III ADVANCED DESIGN AND ANALYSIS TECHNIQUES 9

Dynamic Programming: Matrix-chain multiplication - Elements of dynamic programming – Optimal binary search trees .Greedy Algorithms: An activity-selection problem - Elements of the greedy strategy. Amortized Analysis: Aggregate analysis - accounting method, potential method -Dynamic tables

Module – IV ADVANCED DATA STRUCTURES 9

B-Trees: Definition of B-trees - Basic operations on B-trees - Deleting a key from a B-tree . Fibonacci Heaps: Structure of Fibonacci heaps - Mergeable-heap operations - Decreasing a key and deleting a node - Bounding the maximum degree .Data Structures for Disjoint Sets: Disjoint-set operations- Linked-list representation of disjoint sets - Disjoint-set forests

Module – V GRAPH ALGORITHMS 9

Elementary Graph Algorithms: Representations of graphs - Breadth-first search - Depth-first search - Topological sort -Strongly connected components - Minimum Spanning Trees: Growing a minimum spanning tree - Kruskal and Prim algorithm - Single-Source Shortest Paths: The Bellman-Ford algorithm - Single-source shortest paths in directed acyclic graphs - Dijkstra’s algorithm - All-Pairs Shortest Paths: Shortest paths and matrix multiplication - The Floyd-Warshall algorithm - Johnson’s algorithm - Maximum Flow : Flow networks - The Ford-Fulkerson method - Maximum bipartite matching

STATE OF ART (Not for Exam)

Solving all kind of Storage, Accessing, Searching Problems, Design approaches in Cloud Computing.

Total Hours: 45+30

3. *Course outcomes:*

- *Understand the algorithmic foundations and sorting order statistics*
- *Understand the properties of various basic data structures*
- *Analyze different algorithm design techniques.*
- *Understand the properties of advanced data structures and graph algorithms.*
- *Design and employ appropriate data structures for solving real time applications.*

REFERENCES:

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, " Introduction to Algorithms", Prentice Hall of India ,third edition ,2012
2. Mark Allen Weiss, Data Structures and Algorithms in C++, Pearson, 2009.
3. E. Horowitz, S. Sahni and S. Rajasekaran, Computer Algorithms / C++, University Press, 2007.
4. Adam Drozdex, Data Structures and algorithms in C++. New Delhi: Thomson learning, 2006.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ402	ADVANCED OPERATING SYSTEMS	3	2	0	4

Distribution of marks in Question Paper – Theory (70%) and Problem (30%)

1. Course objectives:

- To provide a modern and forward-looking perspective of computer system design, based on enforcing modularity.
- To provide a solid foundation about the mechanisms that underlie operating systems, database systems, computer security, distributed systems, fault-tolerant computing, and concurrency
- To understand the engineering of many aspects of computer systems to information management
- To prepared the students to read and understand current professional literature about systems, and know what questions to ask and where to find the answers.

2. Course pre-requisites: Software design and Computer hardware organization

Module - I COMPUTER SYSTEM ORGANIZATION AND NAMING SCHEMES 9

Systems : Systems and complexity - Sources of complexity- Coping with complexity Computer systems are the same, but different - Coping with complexity II . **Elements of Computer System Organization:** The three fundamental abstractions - Naming in computer systems - Organizing computer systems with names and layers - Looking back and ahead -Case study: Unix® file system layering and naming. **The Design of Naming Schemes :** Considerations in the design of naming schemes - Case study: The uniform resource locator (URL)- War stories: Pathologies in the use of names.

Module -II COMMUNICATION MODELS, FILE SYSTEM AND VIRTUALIZATION 9

Enforcing Modularity with Clients and Services: Client/service organization- Communication between client and service - Summary and the road ahead - Case study: The Internet Domain Name System (DNS) - Case study: The Network File System (NFS) . **Enforcing modularity with virtualization:** Client/server organization within a computer using virtualization - Virtual links using SEND , RECEIVE, and a bounded buffer - Enforcing modularity with domains-Virtualizing memory - Virtualizing processors using threads - Thread primitives for sequence coordination - Case study: Evolution of enforced modularity in the Intel x86 - Application: Enforcing modularity using virtual machines

Module - III FAULT TOLERANCE 9

Fault Tolerance: Reliable Systems from Unreliable Components: Faults, failures, and fault-tolerant design - Measures of reliability and failure tolerance - Tolerating active faults - Systematically applying redundancy - Applying redundancy to software and data - Wrapping up reliability Application: A fault tolerance model for CMOS RAM - War stories: fault-tolerant systems that failed

Module - IV TRANSACTIONS AND CONSISTENCY 9

Atomicity: All-or-nothing and Before-or-after: Atomicity - All-or-nothing atomicity I: Concepts - All-or-nothing atomicity II: Pragmatics - Before-or-after atomicity I: Concepts - Before-or-after atomicity II: Pragmatics - Atomicity across layers and multiple sites - Case studies: machine language atomicity - A more complete model of disk failure (Advanced topic) **Consistency :**Constraints and interface consistency

- Cache coherence - Durable storage revisited: geographically separated replicas - Reconciliation - Perspectives

Module - V SECURITY AND TRUSTED COMPUTING

Information Security : Introduction to secure systems - Authenticating principals - Authenticating messages - Message confidentiality - Security protocols - Authorization: controlled sharing - Advanced topic: Reasoning about authentication - Cryptography as a building block (Advanced topic) Case Study: Transport Layer Security (TLS) for the Web - War stories: security system breaches

STATE OF ART (Not for Exam)

Ubiquitous and Pervasive Computing, Grid Computing and Cloud Computing

Total Hours: 45+30=75

3. Course outcomes:

- *Understand the application of naming, layering, and abstraction in the file system*
- *Understand the concept of modularity using Client -Server model, Virtual memory and Virtual Processors.*
- *Understand the design principles and techniques for creating reliable systems from unreliable components, based on modularity.*
- *Able to make flawless updates to data in the presence of concurrent threads, system failures and replication.*
- *Understand cryptographic techniques, which are the basis for most network security.*

REFERENCES

1. Saltzer and Kaashoek. Morgan Kaufmann ,”Principles of Computer Systems Design “ Morghan Kauffmann Publication, 2009
2. Andrew S. Tanenbaum. Modern Operating Systems. Prentice-Hall, third edition, 2008.
3. Abraham Silberchatz, Peter B. Galvin, Greg Gagne, “ Operating Sytem Principles” 9th Edition, John Wiley & Sons, 2013.
4. Pradeep K. Sinha, “Distributed Operating Systems-Concepts and Design” 2nd Edition IEEE 1197.
5. Michael N. Nelson, Brent B. Welch, and John K. Ousterhout. Caching in the Sprite network file system. ACM Transactions on Computer Systems, 6(1):134-154, February 1988.
6. Mahadev Satyanarayanan. Scalable, secure, and highly available distributed file access. IEEE Computer, 23(5):9-21, May 1990.
7. Mendel Rosenblum and John K. Ousterhout. The design and implementation of a log-structured file system. ACM Transactions on Computer Systems, 10(1):26-52, February 1992.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN401	ADVANCES IN SOFTWARE ENGINEERING	3	0	0	3

Distribution of marks in Question Paper – Theory (70%) and Problem (30%)

1. Course Objectives:

- To have a clear understanding of Software Engineering concepts.
- To gain knowledge of the Analysis and System Design concepts.
- To learn how to manage change during development.
- To learn the SOA and AOP concepts.

2. Course Pre-requisites : Software Engineering , Object Oriented Analysis and Design

Module – I INTRODUCTION **9**

System Concepts – Software Engineering Concepts - Software Life Cycle– Development Activities – Managing Software Development – Unified Modeling Language – Project Organization – Communication.

Module – II ANALYSIS **9**

Requirements Elicitation – Use Cases – Unified Modeling Language, Tools – Analysis Object Model (Domain Model) – Analysis Dynamic Models – Non-functional requirements – Analysis Patterns.

Module – III SYSTEM DESIGN **9**

Overview of System Design – Decomposing the system -System Design Concepts – System Design Activities – Addressing Design Goals – Managing System Design.

Module – IV IMPLEMENTATION AND MANAGING CHANGE **9**

Reusing Pattern Solutions –Specifying Interfaces – Mapping Models to Code – Testing Rationale Management –Configuration Management – Project Management -real time interface design.

Module – V ASPECT ORIENTED SOFTWARE DEVELOPMENT **9**

AO Design Principles -Separations of Concerns, Subject Oriented Decomposition, Traits, Aspect Oriented Decomposition, Theme Approach, Designing Base and Crosscutting Themes, Aspect-Oriented Programming using Aspect-J.

Total Hours: 45

3. Course Outcomes:

- A clear understanding of Software Engineering concepts.
- Knowledge gained of Analysis and System Design concepts.
- Ability to manage change during development.
- Basic idea of the SOA and AOP concepts.

REFERENCES:

1. Bernd Bruegge, Alan H Dutoit, Object-Oriented Software Engineering, 2nd ed, Pearson Education, 2004.
2. Stephen Schach, Software Engineering 7th ed, McGraw-Hill, 2007.
3. Aspect-Oriented Software Development, Robert E. Filman, TzillaElrad, Siobhan Clarke, and Mehmet Aksit, October 2006.
4. Aspect-Oriented Analysis and Design: The Theme Approach, (The Addison-Wesley Object Technology Series), Siobhan Clarke and Elisa Baniassad, March 2005.
5. Analysis Patterns, Martin Fowler, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1997

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN451	SOFTWARE REQUIREMENTS AND DESIGN LABORATORY	0	0	4	2

1. Course learning objectives :

- Be exposed to requirements engineering activities.
- Be exposed to software design activities.
- Be familiar and develop various requirements documents based on IEEE standards.
- Be familiar and develop various design documents based on IEEE standards.

2. Course pre-requisites : NIL

3. Expected Level of Output : Practical

4. Department Offered : Computer Science and Engineering

5. Nature of the Course : Group 5 - Practical

(Please choose any one)

A. Group 1 – 100 % Descriptive

B. Group 2 – 100% Analytical

C. Group 3 – __% Descriptive & __% Analytical

D. Group 4 – Programming

E. Group 5 – Practical

F. Group 6 – Project

G. Group 7 – 20 % Descriptive & 80% Programming

H. Group 8 –

I. Group 9 –

Continuous Internal Assessment (CIA) : 40 Marks

Semester End Examination (SEE) : 60 Marks

6. List of Experiments:

1. The students should develop all the necessary requirements based on IEEE standards or any other standardized standards and should prepare requirement document and design document after completion.
2. Use any open source software for requirements elicitation, requirements analysis and requirements validation.
3. Use any open source software for performing software design based on the requirements obtained in 2 for each system.

1. ONLINE SHOPPING MALL

PROJECT DESCRIPTION:

The Online Shopping Mall (OSM) application enables vendors to set up online shops, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. Also on the agenda is designing an online shopping

site to manage the items in the shop and also help customers purchase them online without having to visit the shop physically.

The online shopping mall will showcase a complete shopping experience in a small package. This project envisages bridging the gap between the seller, the retailer and the customer. A very high flexibility is being maintained in the design process so that this project can take the following path:

- A multiple merchant venue with each merchant having his/her own window which the customer can visit to browse and subsequently buy the products.
- Maintaining the deliverable goods as well as services through single or multiple windows is also on the agenda.

Target Users:

Mall Administrator: The Mall Administrator is the super user and has complete control over all the activities that can be performed. The application notifies the administrator of all shop creation requests, and the administrator can then approve or reject them. The administrator also manages the list of available product categories. The administrator can also view and delete entries in the guestbook.

Shop Owner: Any user can submit a shop creation request through the application. When the request is approved by the Mall Administrator, the requester is notified, and from there on is given the role of Shop Owner. The Shop Owner is responsible for setting up the shop and maintaining it. The job involves managing the sub-categories of the items in the shop. Also, the shop owner can add or remove items from his shop. The Shop Owner can view different reports that give details of the sales and orders specific to his shop. The Shop Owner can also decide to close shop and remove it from the mall.

Mall Customer/Guests: A Mall Customer can browse through the shops and choose products to place in a virtual shopping cart. The shopping cart details can be viewed and items can be removed from the cart. To proceed with the purchase, the customer is prompted to login. Also, the customer can modify personal profile information (such as phone number and shipping address) stored by the application. The customer can also view the status of any previous orders.

EMPLOYEES:

- Purchase department under a Purchase manager to overlook purchasing activities if warehousing needs arise.
- Sales department under a Sales manager who will look after the sale of products and services.
- Accounts department under an Accounts manager to look after the accounting activities of the enterprise.

2. BANKING SYSTEM

PROJECT DESCRIPTION:

A bank has several automated teller machines (ATMs), which are geographically distributed and connected via a wide area network to a central server. Each ATM machine has a card reader, a cash dispenser, a keyboard/display, and a receipt printer. By using the ATM machine, a customer can withdraw cash from either checking or savings account, query the balance of an account, or transfer funds from one account to another. A transaction is initiated when a customer inserts an ATM card into the card reader. Encoded on the magnetic strip on the back of the ATM card are the card number, the start date, and the expiration date.

Assuming the card is recognized, the system validates the ATM card to determine that the expiration date has not passed, that the user-entered PIN (personal identification number) matches the PIN maintained by the system, and that the card is not lost or stolen. The customer is allowed three attempts to enter the correct PIN; the card is confiscated if the third attempt fails. Cards that have been reported lost or stolen are also confiscated. If the PIN is validated satisfactorily, the customer is prompted for a withdrawal, query, or transfer transaction. Before withdrawal transaction can be approved, the system determines that sufficient funds exist in the requested account, that the maximum daily limit will not be exceeded, and that there are sufficient funds available at the local cash dispenser.

If the transaction is approved, the requested amount of cash is dispensed, a receipt is printed containing information about the transaction, and the card is ejected. Before a transfer transaction can be approved, the system determines that the customer has at least two accounts and that there are sufficient funds in the account to be debited. For approved query and transfer requests, a receipt is printed and card ejected. A customer may cancel a transaction at any time; the transaction is terminated and the card is ejected. Customer records, account records, and debit card records are all maintained at the server.

3. CAMPUS MANAGEMENT SYSTEM

PROJECT DESCRIPTION:

The Campus Management System; is fully computerized information organization, storage and retrieval system that could provide us any information about an Institute just at the click of a mouse. The most fascinating asset about a computerized College fee Manager is that it enables us to explore any institute related information at any time on demand and that too in an absolutely user friendly environment that could be accessed even by a layman very easily

OBJECTIVES AND GOALS:

- To automate the functions at a Higher Education Institute, the main missions of this software are as under
- To provide user-friendly interface to the college administrator
- To minimize the typing errors during data entry
- To search record of a particular object (course, student, faculty etc.)
- To update the record of an object
- To generate various reports for management
- To print various reports
- To reduce the typing work by keeping maximum information available on the screen
- To reduce the expenditure involving stationery items such as paper, ledgers, fee receipt book etc.
- To provide consistent, updated and reliable data at any time on demand.
- To analyse, plan and forecast the inflation or recession graph of the in college in the near future based on the college's record of revenue sources and expenditure.
- To provide the most important feature of maintaining the valuable back-up of the critical data.
- To be bestowed with the latest security facilities provided by the modern computerized DBMS.

PROJECT BUILDING BLOCKS:

Enrolment Management , Portal management , Admissions/Recruiting , Faculty Information, Student Services, Student Portal, Hostel management, Parking and Security, Student Health , Student Placement ,Campus Incidents, Faculty Portal , Forum Portal , Student Billing ,Alumni Portal.

4. AIR TRAFFIC CONTROL SYSTEM

Air traffic control is a closed loop activity in which pilots state the intent by filing flight plans. Controllers then plan traffic flow based on the total number of flight plans and, when possible, given clearance to pilots to fly according to their plans. When planning conflicts arise, controllers resolve them by clearing pilots to fly alternatives to their plans to avoid the conflicts. If unpredicted atmospheric conditions (e.g., wind speed or direction) or pilot actions cause deviations from conflict-free planned routings, controllers issue clearances for tactical maneuvers that solve any resultant problem, albeit not necessarily in a way that furthers the pilot's goal of reaching the planned destination at a certain time.

PROBLEM FORMULATION:

Design an air traffic control system (ATCS) that is fault tolerant and scalable, according to the specific requirements listed in the following sections. The primary objective of the ATCS is to provide separation services for aircraft that are flying in controlled air space, or where poor

visibility prevents from maintaining visual separation. Aircraft are separated from one another and from terrain hazards.

SPECIFIC SOFTWARE REQUIREMENTS:

The requirements of ATCSs include real-time aspects. The ATCS is a "dynamic" real-time system. Its loading will vary significantly over time, and has no upper bound. Loading scenarios can vary significantly, hence the average loading of the ATCS is not a highly useful metric for schedulability and other analyses. Although an upper bound could possibly be imposed artificially, this may not be a cost-effective solution, since pre-allocation of computing resources for such a worst case would lead to very poor resource utilization. A dynamic resource management policy is thus preferred.

5. CAFETERIA ORDERING SYSTEM

The Cafeteria Ordering System is a new system that replaces the current manual and telephone processes for ordering and picking up lunches in the Process Impact cafeteria. **Patron:** A Patron is a Process Impact employee at the corporate campus in TidalPark, Chennai, who wishes to order meals to be delivered from the company cafeteria.

There are about 600 potential Patrons, of which an estimated 400 are expected to use the Cafeteria Ordering System. Patrons will sometimes order multiple meals for group events or guests. An estimated 90 percent of orders will be placed using the corporate Intranet, with 10 percent of orders being placed from home. All Patrons have Intranet access from their offices. Some Patrons will wish to set up meal subscriptions, either to have the same meal to be delivered every day or to have the day's meal special delivered automatically. A Patron must be able to override a subscription for a specific day.

Cafeteria Staff: The Process Impact cafeteria currently employs about 20 Cafeteria Staff, who will receive orders from the Cafeteria Ordering System, prepare meals, and package them for delivery, print delivery instructions, and request delivery. Most of the Cafeteria Staff will need to be trained in the use of the computer, the Web browser, and the Cafeteria Ordering System. **Menu Manager:** The Menu Manager is a cafeteria employee, perhaps the cafeteria manager, who is responsible for establishing and maintaining daily menus of the food items available from the cafeteria and the times of day that each item is available. Some menu items may not be available for delivery. The Menu Manager will also define the cafeteria's daily specials. The Menu Manager will need to edit the menus periodically to reflect planned food items that are not available or price changes.

Meal Deliverer: As the Cafeteria Staff prepare orders for delivery, they will print delivery instructions and issue delivery requests to the Meal Deliverer, who is either another cafeteria employee or a contractor. The Meal Deliverer will pick up food and delivery instructions for each meal and deliver it to the Patron. The Meal Deliverer's primary interactions with the system will be to reprint the delivery instructions on occasion and to confirm that a meal was (or was not) delivered.

7. Expected outcome of the course:

- *Demonstrate requirements engineering skills including the ability to formulate problems and their solutions.*
- *Develop various software requirements documents based on IEEE Standard using any open source software.*
- *Demonstrate software design skills and provide a solution for a requirement problem.*
- *Develop various software design documents based on IEEE standard using any open source software.*

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ404	ADVANCED DATABASE TECHNOLOGY	3	2	0	4

Distribution of marks in Question Paper – Theory (70%) and Problem (30%)

1. Course objectives:

- To understand database systems, data models, database languages
- To be able to design a database system by understanding the concepts of functional dependencies and normalization
- To acquire knowledge on parallel and distributed databases and its query processing.
- To understand the principles of object oriented databases.
- To gain knowledge about the emerging database technologies.

2. Course pre-requisites : NIL

Module- I INTRODUCTION TO DATABASE SYSTEMS AND RELATIONAL MODEL 9

File systems - Database systems - Database systems architecture - Database Languages - Data Dictionary - Database Administration and control-Data models - Entity-Relationship model – entities, entity types, various types of attributes, relationships, relationship types and extended E-R features, ER diagram notation, examples- Reduction of ER model to relational schema.

Module- II STRUCTURED QUERY LANGUAGE AND NORMALIZATION 9

Fundamentals of SQL- Domains and Integrity constraints-Views-Triggers -Procedures and functions - Normalization and database design-Functional Dependencies-Desirable properties of Decomposition - First, Second, Third Normal Forms, Boyce/Codd Normal Form- Multi-valued Dependencies and Fourth Normal Form.

Module- III PARALLEL AND DISTRIBUTED DATABASES 9

Database System Architectures: Centralized and Client-Server Architectures -Parallel Systems- Distributed Systems. Parallel Databases: I/O Parallelism - Inter and Intra Query Parallelism - Inter and Intra operation Parallelism- Distributed Database Concepts - Distributed Data Storage-Distributed Transactions-Commit Protocols - Concurrency Control - Distributed Query Processing- Recovery Concepts. Query optimization- Database Tuning.

Module- IV OBJECT ORIENTED AND OBJECT RELATIONAL DATABASES 9

Concepts for Object Databases: Object Identity - Object structure - Type Constructors-Methods-Type and Class Hierarchies -Inheritance - Complex Objects- Object Oriented Languages -ODMG Model - Object Relational Databases - Nested Relations- Complex Types and Object Orientation- Querying With Complex Types.

Module- V INTELLIGENT DATABASES 9

Introduction to intelligent databases - Temporal - Spatial databases - Multimedia databases - Embedded database – NoSQL database – In-memory database

STATE OF ART (Not for Exam)

Challenges in NoSQL database.

Total Hours: 45+30

3. *Course outcomes:*

- Ability to understand and model database systems
- Ability to design a database system
- Ability to design parallel and distributed databases
- Apply the object oriented concepts in databases
- Ability to understand and analyze various intelligent databases

REFERENCES:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan- “Database System Concepts”, Sixth Edition, McGraw-Hill, 2010.
2. Ramez Elmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Fifth Edition, Pearson Education, 2006.
3. C.J.Date, A.Kannan, S.Swamynathan -“An Introduction to Database System”, Eighth Edition, Pearson education, 2006.
4. Won Kim, MIT Press , “Introduction to Object Oriented Databases”, MIT Press,2003.
5. Thomas Cannolly and Carolyn Begg, Database Systems, A Practical Approach to Design, Implementation and Management. New Delhi: Pearson Education, 2009.
6. V.S.Subramanian, “Principles of Multimedia Database Systems”, Harcourt India Pvt Ltd., 2001.
7. Raghu Ramakrishnan,Johannes Gehrke,” Database Management Systems”,Third Edition,Mc.Graw Hill,2008.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN402	SOFTWARE TESTING	3	2	0	4

Distribution of marks in Question Paper – Theory (70%) and Problem (30%)

1. Course objectives:

- To know the behavior of the testing techniques to detect the errors in the software
- To understand standard principles to check the occurrence of defects and its removal.
- To learn the functionality of automated testing tools
- To understand the models of software reliability.

2. Course pre-requisites : Software Engineering

Module – I TESTING ENVIRONMENT AND TEST PROCESSES 9

World-Class Software Testing Model – Building a Software Testing Environment - Overview of Software Testing Process – Organizing for Testing – Developing the Test Plan – Verification Testing – Analyzing and Reporting Test Results – Acceptance Testing – Operational Testing – Post Implementation Analysis

Module – II TESTING TECHNIQUES AND LEVELS OF TESTING 9

Using White Box Approach to Test design - Static Testing Vs. Structural Testing – Code Functional Testing – Coverage and Control Flow Graphs –Using Black Box Approaches to Test Case Design – Random Testing – Requirements based testing –Decision tables –State-based testing – Cause-effect graphing – Error guessing – Compatibility testing – Levels of Testing - Module Testing - Integration Testing - Defect Bash Elimination. System Testing - Usability and Accessibility Testing – Configuration Testing - Compatibility Testing - Case study for White box testing and Black box testing techniques.

Module – III INCORPORATING SPECIALIZED TESTING RESPONSIBILITIES 9

Testing Client/Server Systems – Rapid Application Development Testing – Testing in a Multiplatform Environment – Testing Software System Security - Testing Object-Oriented Software – Object Oriented Testing – Testing Web based systems – Web based system – Web Technology Evolution – Traditional Software and Web based Software – Challenges in Testing for Web-based Software –Testing a Data Warehouse - Case Study for Web Application Testing.

Module – IV TEST AUTOMATION 9

Selecting and Installing Software Testing Tools - Software Test Automation – Skills needed for Automation – Scope of Automation – Design and Architecture for Automation – Requirements for a Test Tool – Challenges in Automation – Tracking the Bug – Debugging – Case study using Bug Tracking Tool.

Module – V TESTING STANDARDS AND DOCUMENTATION 9

ISO ,CMMI and PCMMI, Six Sigma, Types of software documentation, The importance of documentation testing, Factors for reviewing documentation, Case study the realities of documentation Testing

STATE OF ART (Not for Exam)

“Big Data” for Software Testing: Test optimization Test case aging

Total Hours: 45+30

3. *Course outcomes:*

- *Test the software by applying testing techniques to deliver a product free from bugs*
- *Evaluate the web applications using bug tracking tools.*
- *Investigate the scenario and the able to select the proper testing technique*
- *Explore the test automation concepts and tools*
- *Deliver quality product to the clients by way of applying standards*

REFERENCES:

1. William Perry, “Effective Methods of Software Testing”, Third Edition, Wiley Publishing 2007
2. Srinivasan Desikan and Gopaldaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson Education, 2007.
3. NareshChauhan, “Software Testing Principles and Practices” Oxford University Press, New Delhi, 2010.
4. Dale H. Besterfield et al., “Total Quality Management”, Pearson Education Asia, Third Edition, Indian Reprint (2006).
5. Stephen Kan, “Metrics and Models in Software Quality”, Addison – Wesley, Second Edition, 2004.
6. LleneBurnstein, “Practical Software Testing”, Springer International Edition, Chennai, 2003
7. RenuRajani,Pradeep Oak, “Software Testing – Effective Methods, Tools and Techniques”, Tata McGraw Hill,2004.
8. Edward Kit, “Software Testing in the Real World – Improving the Process”, Pearson Education, 1995.
9. Boris Beizer, “Software Testing Techniques” – 2nd Edition, Van Nostrand Reinhold, NewYork, 1990
10. Adithya P. Mathur, “Foundations of Software Testing – Fundamentals algorithms and techniques”, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN403	SOFTWARE PROJECT PLANNING AND MANAGEMENT	3	2	0	4

Distribution of marks in Question Paper – Theory (70%) and Problem (30%)

1. Course objectives:

- To provide basic project management skills with a strong emphasis on issues and problems associated with delivering successful IT projects.
- To understand the issues encountered in handling IT projects and to offer methods, techniques and 'hands-on' experience in dealing with them
- To be able to undertake and be aware of aspects of project management.

2. Course pre-requisites : Software Engineering

Module - I INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT 9

Project Definition – Contract Management – Activities covered By Software Project Management – Overview of Project Planning – Stepwise Project Plan development.

Module - II PROJECT EVALUATION 9

Strategic Assessment – Technical Assessment – Cost Benefit Analysis – Cash Flow Forecasting – Cost Benefit Evaluation Techniques – Risk Evaluation – Selection of appropriate project planning.

Module - III ACTIVITY PLANNING 9

Objectives – Project Schedule – Sequencing and Scheduling Activities –Network Planning Models – Using dummy activities – Identifying critical path and Identifying critical activities. Forward Pass – Backward Pass – Activity Float – Shortening Project Duration – Activity on Arrow Networks – Risk Management – Nature Of Risk – Types Of Risk – Managing Risk – Hazard Identification – Hazard Analysis – Risk Planning And Control.

Module - IV SOFTWARE EFFORT ESTIMATION, MONITORING AND CONTROL 9

Problems with over and under estimate - The basis for software estimation - Software estimation Techniques. Creating Framework – Collecting The Data – Visualizing Progress – Cost Monitoring – Earned Value – Prioritizing Monitoring – Getting Project Back To Target – Change Control – Managing Contracts – Introduction – Types Of Contract – Stages In Contract Placement – Typical Terms Of A Contract – Contract Management – Acceptance.

Module - V MANAGING PEOPLE AND ORGANIZING TEAMS 9

Introduction – Understanding Behavior – Organizational Behaviour: A Background – Selecting The Right Person For The Job – Instruction In The Best Methods – Motivation – The Oldham – Hackman Job Characteristics Model – Stress –Health And Safety – Working In Groups – Becoming A Team – Decision Making – Leadership – Organizational Structures – Communication Plans – Case Studies.

STATE OF ART (Not for Exam)

Project Closure and Agile Project Management.

Total Hours: 45+30

3. *Course outcomes:*

- *Understand and practice the process of project management and its application in delivering successful IT projects;*
- *Evaluate a project to develop the scope of work, provide accurate cost estimates and to plan the various activities;*
- *Understand and use risk management analysis techniques that identify the factors that put a project at risk and to quantify the likely effect of risk on project timescales;*
- *Identify the resources required for a project and to produce a work plan and resource schedule;*

REFERENCES:

1. Bob Hughes, Mike Cotterell, "Software Project Management", Fifth Edition, Tata McGraw Hill, 2011.
2. E. Larson and C. Gray, Project Management: The Managerial Process, Fourth Edition, Tata McGraw Hill, 2010.
3. M.B. Chrissis, M.D. Konrad, and S. Shrum, CMMI for Development: Guidelines for Process Integration and Product Improvement, Addison-Wesley Professional, Third Edition, 2011.
4. Ramesh, Gopaldaswamy, "Managing Global Projects", Tata McGraw Hill, 2006.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN452	DATABASE & SOFTWARE TESTING LABORATORY	0	0	4	2

DATABASE

1. Course learning objectives :

- To understand the concept of designing a database with the necessary attributes.
- To know the methodology of Accessing, Modifying and Updating data & information from the relational databases.
- To understand the purpose of views and triggers
- To develop procedures, functions in SQL.
- To develop an application

2. Course pre-requisites : NIL

3. Expected Level of Output : Practical

4. Department Offered : Computer Science and Engineering

5. Nature of the Course : Group 5 - Practical

(Please choose any one)

- | | |
|---|------------------------------|
| A. Group 1 – 100 % Descriptive | B. Group 2 – 100% Analytical |
| C. Group 3 – __% Descriptive & __% Analytical | D. Group 4 – Programming |
| <u>E. Group 5 – Practical</u> | F. Group 6 – Project |
| G. Group 7 – 20 % Descriptive & 80% Programming | H. Group 8 – |
| I. Group 9 – | |

Continuous Internal Assessment (CIA) : 40 Marks

Semester End Examination (SEE) : 60 Marks

6. List of Experiments:

1. Study of SQL commands-DDL, DML, Nested Queries & Join Queries, and Views.
2. Working with Triggers, Procedures and Functions
3. Database Design using ER diagram and Normalization
4. Database implementation (Mini Project)
5. Working with NoSQL Databases (PIG/HIVE)

7. Expected outcome of the course:

- Construct an Entity Relationship (E-R) diagram for an application.
- Create a normalized relational database model and write queries to retrieve information.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN452	DATABASE & SOFTWARE TESTING LABORATORY	0	0	4	2

SOFTWARE TESTING

1. **Course learning objectives** :
- To know the input and deliverables of the testing process.
 - To understand appropriate technological tools to analyze, test and automate the elements of software

2. **Course pre-requisites** : NIL

3. **Expected Level of Output** : **Practical**

4. **Department Offered** : Computer Science and Engineering

5. **Nature of the Course** : Group 5 - Practical

(Please choose any one)

A. Group 1 – 100 % Descriptive

B. Group 2 – 100% Analytical

C. Group 3 – __% Descriptive & __% Analytical

D. Group 4 – Programming

E. Group 5 – Practical

F. Group 6 – Project

G. Group 7 – 20 % Descriptive & 80% Programming

H. Group 8 –

I. Group 9 –

Continuous Internal Assessment (CIA) : 40 Marks

Semester End Examination (SEE) : 60 Marks

6. List of Experiments:

1. Cause Effect Graph Testing for a Triangle Program
2. Boundary Value Analysis for a Software Unit
3. Cyclomatic Complexity for Binary Search
4. Data Flow Testing for Gregorian Calendar
5. State based Testing for an Assembler
6. Stress Testing of a Map-Aided Vehicle Tracking and Scheduling System
7. Model Based Testing
8. Web Application Testing for Student Grade System
9. Do the following testing for a chosen software implementation using any commercial or freeware tools: Path-testing
10. Do the following measurements on a chosen software implementation: JUnit Test

7. Expected outcome of the course:

- *Ability to find the practical solution to the problem.*
- *Ability to write test cases for various applications.*
- *Apply the principles of software engineering and mathematics for the design and implementation of software systems*
- *Ability to design processes and quality system components to meet the specific needs.*

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN501	SOFTWARE REUSE	3	0	0	3

Distribution of marks in Question Paper – Theory (90%) and Problem (10%)

1. Course objectives:

- *To identify benefits of software reuse*
- *To know software reuse management*
- *To design reusable software components*
- *To determine different phases of software reuse*
- *To learn clean room software Engineering*

2. Course pre-requisites : Software Engineering

Module - I INTRODUCTION 9

Organizing Reuse – Introduction – Motivation for Reuse – Framework for Reuse- Evolution of Reuse - Reuse in industry – Managing a reuse project – Software Reuse Products- Software Reuse Processes and paradigms – Reuse tools.

Module - II REUSE MANAGEMENT 9

Managing a repository – The REBOOT component model – Classification – Configuration management of the repository – Managerial aspects of software Reuse– Software Reuse Metrics – Software Reuse Cost estimation – Forming a reuse Strategy – Assessing reuse maturity.

Module - III REUSABLE COMPONENTS 9

Practicing reuse – Reuse Techniques- Generic reuse development processes – Develop for reuse – Testing reusable components – Object oriented components – Object oriented development for reuse – Reuse Techniques- Reuse Technologies- Detailed design for reuse – Implementation for reuse – Verification, test and validation.

Module - IV REUSE PHASES 9

Development with reuse – with reuse specific activities – Common reuse processes – Phases of development with reuse – Impact of reuse on development cycle- Reuse Technologies.

Module - V CLEANROOM SOFTWARE ENGINEERING 9

Re-engineering for reuse – Methodology – Retrieving objects in non-object oriented code– Measurements – Tools support for re-engineering – Overview of clean room software engineering – Phases in clean room method – Box structures algorithms – Adapting the box structures.

Total Hours: 45

3. Course outcomes:

- *Understand Software Reuse*
- *Establish and Manage Software Reuse*
- *To identify and use software reuse tools*
- *Analyze software reuse tools and practice clean room software engineering*

REFERENCES:

1. Wayne C.Lim, "Managing Software Reuse", Prentice Hall.
2. Hafedh Mili , Ali Mili, Sherif Yacoub, "Reuse based Software Engineering: Techniques, Organizations and Controls", John Wiley and Sons.
3. Karma McClure, "Software Reuse Techniques – Additional Reuse To The Systems Development Process ", Prentice Hall.
4. Even-Andre Karisson, "Software Reuse – A Holistic Approach", John Wiley And Sons.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN502	SOFTWARE AGENTS	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To help students understand various Software Components.
- To know the use of various java agents
- To understand the purpose of intelligent software agents

2. Course pre-requisites : Software Engineering

Module - I AGENTS – OVERVIEW 9
 Agent Definition – Agent Programming Paradigms – Agent Vs Object – Aglet – Mobile Agents – Agent Frameworks – Agent Reasoning.

Module - II JAVA AGENTS 9
 Processes – Threads – Daemons – Components – Java Beans – ActiveX – Sockets – RPCs – Distributed Computing – Aglets Programming – Jini Architecture – Actors and Agents – Typed and proactive messages.

Module - III MULTIAGENT SYSTEMS 9
 Interaction between agents – Reactive Agents – Cognitive Agents – Interaction protocols – Agent coordination – Agent negotiation – Agent Cooperation – Agent Organization – Self-Interested agents in Electronic Commerce Applications.

Module - IV INTELLIGENT SOFTWARE AGENTS 9
 Interface Agents – Agent Communication Languages – Agent Knowledge Representation – Agent Adaptability – Belief Desire Intension – Mobile Agent Applications.

Module - V AGENTS AND SECURITY 9
 Agent Security Issues – Mobile Agents Security – Protecting Agents against Malicious Hosts – Untrusted Agent – Black Box Security – Authentication for agents – Security issues for Aglets.

Total Hours: 45

3. Course outcomes:

- To know the use of various software agents
- Able to use java agents for software developments
- Able to use Intelligent software agents.

REFERENCES

1. Bigus & Bigus, " Constructing Intelligent agents with Java ", Wiley,.
2. Bradshaw, " Software Agents ", MIT Press,.
3. Russel, Norvig, "Artificial Intelligence: A Modern Approach", Second Edition, Pearson Education.
4. Richard Murch, Tony Johnson, "Intelligent Software Agents", Prentice Hall.
5. Gerhard Weiss, "Multi Agent Systems – A Modern Approach to Distributed Artificial Intelligence", MIT Press.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN503	FORMAL METHODS IN SOFTWARE ENGINEERING	3	0	0	3

Distribution of marks in Question Paper – Theory (60%) and Problem (40%)

1. Course Objectives:

- *To understand the basic elements of Z.*
- *To understand relations, functions, and logical structures in Z.*
- *To understand Z schemas and schema calculus.*
- *To learn selected Z case studies.*
- *To understand Z schema refinement.*

2. Course Pre-requisites : Software Engineering

Module - I FOUNDATIONS OF Z 9

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.

Module - II STRUCTURES IN Z 9

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 – modeling systems and change.

Module - III Z SCHEMAS AND SCHEMA CALCULUS 9

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.

Module - IV Z CASE STUDIES 9

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.

Module - V Z REFINEMENT 9

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.

Total Hours: 45

3. Course Outcomes:

- *Apply the basic elements of Z*
- *Develop relational, functional, and logical Z structures*
- *Develop Z schema as models of software systems*
- *Perform verifications and conduct proofs using Z models*
- *Refine Z models towards implementing software systems*

REFERENCES:

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.
3. Jim Woodcock and Jim Davies, “Using Z – Specification, Refinement, and Proof”, Prentice Hall, 1996.
4. J. M. Spivey, “The Z notation: A reference manual”, Second Edition, Prentice Hall, 1992.
5. M. Ben-Ari, “Mathematical logic for computer science”, Second Edition, Springer, 2003.
6. M. Huth and M. Ryan, “Logic in Computer Science – Modeling and Reasoning about systems”, Second Edition, Cambridge University Press, 2004.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN504	SOFTWARE RELIABILITY	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand different definitions of software quality and how you might measure it
- To understand different notions of defects and be able to classify them
- To understand the basic techniques of data collection and how to apply them
- To learn software metrics that defines relevant metrics in a rigorous way.

2. Course pre-requisites : Software Testing, Software Quality Management

Module - I INTRODUCTION OF SOFTWARE RELIABILITY 9

Basic Concepts –Failure and Faults –Environment –Availability –Modeling –uses –requirements reliability metrics –design & code reliability metrics –testing reliability metrics.

Module – II SOFTWARE RELIABILITY MODELING 9

Concepts –General Model Characteristic –Historical Development of models –Model Classification scheme –Markovian models –General concepts –General Poisson Type Models –Binomial Type Models –Poisson Type models –Fault reduction factor for Poisson Type models.

Module – III COMPARISON OF SOFTWARE RELIABILITY MODELS 9

Comparison Criteria –Failure Data –Comparison of Predictive Validity of Model Groups –Recommended Models –Comparison of Time Domains –Calendar Time Modeling –Limiting Resource Concept –Resource Usage model –Resource Utilization –Calendar Time Estimation and confidence Intervals

Module – IV FUNDAMENTALS OF MEASUREMENTS 9

Measurements in Software Engineering –Scope of Software metrics –Measurements theory –Goal based Framework –Software Measurement Validation.

Module – V PRODUCT METRICS 9

Measurement of Internet Product Attributes –Size and Structure –External Product Attributes –Measurement of Quality –Reliability Growth Model –Model Evaluation.

STATE OF ART (Not For Exam)

Software Testing – Types, White and Black Box, Operational Profiles – Difficulties, Estimating Reliability.

Total Hours: 45

3. Course outcomes:

- To be able to perform some simple statistical analysis relevant to software measurement data.
- To understand from practical examples both the benefits and limitations of software metrics for quality control and assurance

REFERENCES:

1. John D. Musa, Anthony Iannino, Kazuhira Okumoto, —Software Reliability –Measurement, Prediction, Application, Series in Software Engineering and Technology, McGraw Hill, 1987.
2. John D. Musa, —Software Reliability Engineering, Tata McGraw Hill, 1999.
3. Norman E . Fenton, Shari Lawrence Pfleeger, "Software metrics", Second Edition, International Student Edition, 2003.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN505	SOFTWARE METRICS & QUALITY ASSURANCE	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand software metrics and measurement.
- To emphasize the use of product and quality metrics.
- To explain quality assurance and various tools used in quality management.
- To learn in detail about various quality assurance models.
- To understand the audit and assessment procedures to achieve quality.

2. Course pre-requisites : Software Quality Assurance

Module – I INTRODUCTION TO SOFTWARE METRICS 9
 Fundamentals of measurement-Scope of software metrics-Measurement theory-Software measurement validation software metrics ,data collection – Analysis methods.

Module – II PRODUCT AND QUALITY METRICS 9
 Measurement of internet product attributes-size and structure-external product attributes, measurement of quality- Software quality metrics-product quality-process quality- metrics for software maintenance.

Module – III FUNDAMENTALS OF SOFTWARE QUALITY ASSURANCE 9
 SQA basics-Software quality in business context – Planning for software quality assurance –Product quality and process quality – Software process models -Total Quality Management- QC Tools and Modern Tools.

Module -IV QUALITY ASSURANCE MODELS 9
 Models for Quality Assurance-ISO-9000 – Series- CMM- CMMI-Test Maturity Models, SPICE,Malcolm Baldrige Model- P-CMM.

Module - V SOFTWARE QUALITY ASSURANCE TRENDS 9
 Software Process- PSP and TSP - OO Methodology, Clean-room software engineering, Defect injection and prevention -Internal Auditing and Assessments-Inspections & Walkthroughs.

STATE OF ART (Not for Exam)

Software Testing Metrics, Social Networks Metrics in Object-Oriented Software, Agile Software Quality Assurance

Total Hours: 45

3. Course outcomes:

- Knowledge on how to choose which metrics to collect and use them to make predictions.
- Ken on product and quality metrics.
- Understand how to detect, classify, prevent and remove defects.
- Choose appropriate quality assurance models and develop quality.
- Ability to conduct formal inspections, record and evaluate results of inspections

REFERENCES:

1. Norman E-Fentor and Share Lawrence Pflieger." Software Metrics". International Thomson Computer Press, 1997.
2. Stephen H.Kan,"Metric and Models in software Quality Engineering", Addison QWesley 1995.
3. S.A.Kelkar,"Software quality and Testing, PHI Learning, Pvt, Ltd., New Delhi 2012.
4. Watts S Humphrey, "Managing the Software Process", Pearson Education Inc, 2008.
5. Mary Beth Chrissis, Mike Konrad and Sandy Shrum, "CMMI", Pearson
a. Education(Singapore) Pte Ltd, 2003
6. Philip B Crosby, " Quality is Free: The Art of Making Quality Certain ", Mass Market,1992

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN601	PERSONAL AND TEAM SOFTWARE PROCESS	3	0	0	3

Distribution of marks in Question Paper – Theory (90%) and Problem (10%)

1. Course objectives:

- *To help students understand and improve their performance, by using a disciplined, data driven procedure.*
- *To improve their estimating and planning skills.*
- *To understand the structure of the TSP.*

2. Course pre-requisites : Software Engineering

Module – I INTRODUCTION 9
 Software Engineering – Time management – Tracking Time – Period and product planning – Product planning – Product Size – Managing your time – Managing commitments – Managing Schedules.

Module – II PLANNING 9
 The project plan – The Software Development process – Defects – Finding Defects –The Code review checklist – Design defects – Product quality – Process quality.

Module – III TSP STRATEGY 9
 Team Software process overview – The logic of the team software process – Launching a Team project - The development strategy – The Development Plan – Defining the requirement.

Module - IV PRODUCT IMPLEMENTATION 9
 Designing with teams – Product implementation – integration & system testing – The Postmortem.

Module – V TEAM MANAGEMENT 9
 The team leader role – Development manager role – The planning manager role – The quality – Process manager role – The support manager role.

Total Hours: 45

3. Course outcomes:

- *Understand the PSP process-based approach for developing software*
- *Measure and analyze their personal software processes*
- *Manage and reduce defects through their software processes*
- *Understand and Analyze Team Software Process*

REFERENCE BOOKS

1. Watt S Humphery “Introduction to Personal Software Process, Addison Wesley.
2. Watt S Humphery “Introduction to Personal Software Process, Addison Wesley ”.
3. Mukesh Jain “Delivering Successful Projects with TSP(SM) and Six Sigma: A Practical Guide to Implementing Team Software Process”.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN602	SOFTWARE DESIGN PATTERN	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- *To apply a fundamental set of design patterns utilizing object oriented principles to solve real world software design problems.*
- *Understand and be able to apply incremental/iterative development*
- *Be able to identify appropriate patterns for design problems*
- *Be able to refactor badly designed program by properly using design patterns*

2. Course pre-requisites : Software Engineering

Module – I SOFTWARE DESIGN FUNDAMENTALS AND MODELS 9

Design process- Software Design Process – Design in Software development Process – Design Qualities – Design strategies- Top down and bottom up strategies for design, Design by Template and Design Reuse - Object-Oriented and Object-based design –Hierarchical Object Oriented Design – Architectural design and mapping - Round trip Engineering.

Module – II INTRODUCTION TO SOFTWARE DESIGN PATTERNS 6

History and origin of patterns - Pattern envy and ethics - Prototyping – Testing, Types of pattern - Quality and elements - patterns and rules - Creativity and patterns - How to select and use Patterns.

Module – III DESIGN PATTERNS (CREATIONAL & STRUCTURAL) 10

Abstract Factory - Factory Method – Prototype - Singleton – Builder Adapter Pattern –Decorator – Facade – Proxy - Bridge

Module – IV DESIGN PATTERNS (BEHAVIOURAL) 10

Chain of Responsibility – Mediator – Observer – Strategy– Memento Idioms – Pattern Systems

Module – V FRAMEWORKS AND CASE STUDIES 10

Algorithms and frameworks for patterns - Anti-patterns - Case studies in UML and CORBA.

STATE OF ART (Not for Exam)

Pattern Level Evolutions: Independent, packaged, class group, Correlated classes, Correlated attributes/operations

Total Hours: 45

3. Course outcomes:

- *Students demonstrate a thorough understanding of patterns and their underlying principles and to understand how to evolve patterns into a design*
- *To understand what design pattern to apply to a specific problem*
- *To demonstrate what tradeoffs need to be made when implementing a design pattern*
- *Students will be able to use design patterns when developing software.*

REFERENCES:

1. Eric Gamma, Richard Helm, Ralph Johnson, John Vissides, Grady Booch, "Design patterns: Elements of Reusable object-oriented software", Pearson Education, 2009.
2. David Budgen, "Software Design ", Second Edition, Pearson Education, 2003.
3. William J Brown et al., "Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN603	SOFTWARE RISK MANAGEMENT AND MAINTENANCE	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand the various risk levels in software development
- To gain expertise in discovering risk and usage of risk assessment tools
- To understand the risk plan , implementation and tracking risks
- To realize the software maintenance process, measurement and benchmarking
- To expertise in the SQA maintenance tools

2. Course pre-requisites : Software Engineering

Module – I RISK CULTURE AND MANAGEMENT PROCESS 9

Risk- Basic Terms- Risk Vocabulary – Risk- Driven Project Management- Controlling the Process, Environment and Risk- Maturity in Risk Culture – Risk Scale – Preparing for Risk – Risk Management Paradigms- Five Models of Risk Management – Thinking about Less Risky alternatives – Risk Management at Different Levels – Risk Escalation – Risk Models- Risk Intelligence - Software Risk Management steps.

Module – II DISCOVERING RISK AND ASSESSMENT 9

Identifying software risk- Classification of Risks – Risk Taxonomy – Risk Mapping – Statements– Risk Reviews – Risk Ownership and stakeholder management – Risk Assessment Approach– Risk Assessment tools and techniques – Risk Probability, impact, exposure, matrix and Application Problem- Self-assessment checklist.

Module – III RESPONDING TO RISKS AND TRACKING 9

Special Treatment for Catastrophic risks- Constraint Risks – Risk Mitigation Plan Case Study Contingency Plans- Implementing Risk Response- Tracking Risk Response and Hazards –Trigger Levels- Tracking Project Risks and Operational Risks- Learning by Tracking and Risk Tracker Tool.

Module – IV MAINTENANCE PROCESS 9

Software Maintenance- Customer’s Viewpoint- Economics of Maintenance- Issues in Maintenance- Software Maintenance Standard, Process, Activities and Categories –Maintenance Measurement –Service Measurement and Benchmarking – Problem Resolution- Reporting – Fix Distribution.

Module – V ACTIVITIES FOR MAINTENANCE 9

Role of SQA for Support and Maintenance – SQA tools for Maintenance- Configuration Management and Maintenance – Maintenance of Mission Critical Systems – Global Maintenance Teams – Exemplary Practices.

3. Course outcomes:

- *To students will be able to learn about various risk levels in software development*
- *Students are trained to discover risk and how to use risk assessment tools*
- *Students will be able to prepare risk plan, implement and track risks*
- *They learn about measurement, benchmarking and SQA maintenance tools*

REFERENCES:

1. C. RavindranathPandian, “Applied Software Risk Management: A guide for Software Project Managers”, Auerbach Publications, 2007.
2. John Mcmanus, “Risk Management in Software Development Projects”, Elsevier Butterworth-Heinemann, First Edition, 2004.
3. Alian April and Alain Abran, “Software Maintenance Management: Evaluation and Continuous Improvement”, John Wiley & Sons Inc, 2008.
4. Gopalaswamy Ramesh and Ramesh Bhattiprolu, “Software Maintenance: Effective Practices for Geographically Distributed Environments”, Second Reprint, Tata McGraw-Hill, 2009.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN604	PATTERN ORIENTED SOFTWARE ARCHITECTURE	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- *Students will understand the fundamentals of pattern oriented software architecture.*
- *To know about various architecture patterns*

2. Course pre-requisites : Software Engineering

Module - I INTRODUCTION 9

Architecture business cycle – architectural patterns – reference models – architectural structures, views - Basic Concepts of Software Architecture

Module -II SOFTWARE ARCHITECTURAL PATTERNS 9

Architectural Patterns – Introduction to Styles – Simple Styles - Distributed and Networked Architectures- Architecture for network based applications – Decentralized Architectures

Module - III DESIGNING FOR NON FUNCTIONAL PROPERTIES 9

Understanding Quality Attributes – Functionality and Architecture – Architecture and Quality Attributes - System Quality Attributes – Quality attribute Scenarios in Practice – Introducing Tactics – Availability Tactics – Modifiability Tactics – Performance Tactics - Security Tactics –Testability Tactics – Usability Tactics – Relationship of Tactics to Architectural Patterns –Architectural Patterns and Styles

Module - IV ARCHITECTURE DESCRIPTION DOCUMENTATION AND EVALUATION 9

Early Architecture Description Languages – Domain and Style Specific ADLs – Extensible ADLs - Documenting Software architecture - Architecture Evaluation - ATAM

Module - V ARCHITECTURE ADAPTATION AND CASE STUDY 9

A Conceptual Framework for Architectural Adaptation – Techniques for supporting architecture centric change- The World Wide Web – A Case Study in Interoperability.

STATE OF ART (Not for Exam)

Ecosystem of Edge-Dominant systems, implications of metropolis model

Total Hours: 45

3. Course outcomes:

- *Gives the fundamentals about software architecture*
- *It provides various patterns used for architecture designs*
- *Students will know about functional and non functional properties while designing architecture*
- *Gives the details about how to evaluate the document*

REFERENCES

1. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Third Edition, Addison-Wesley, 2003.
2. Richard N.Taylor, Nenad Medvidovic and Eric M.Dashofy, "Software Architecture, Foundations, Theory and Practice", Wiley 2010.
3. Frank Buschmann, Regine Meunier, Hans Rohnert, Michael Stal, "Pattern Oriented Software Architecture" ,Volume 1, 1996.
4. Mary shaw and David Garlan, "Software Architecture – Perspectives on an emerging discipline", Pearson education, 2008.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN605	SOFTWARE ARCHITECTURE AND DESIGN	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- *To learn the fundamentals of software architecture and design*
- *To learn the software architecture process*
- *To have an in-depth knowledge about architecture models and description*
- *To have an in-depth knowledge about design quality, design approaches and design styles.*

2. Course pre-requisites : Nil

Module - I FUNDAMENTALS OF ARCHITECTURE AND DESIGN 9

Software architecture – Elements, Stakeholders, Architectural descriptions, Architectural views, Viewpoints, Viewpoint pitfalls, Architectural perspectives, Role of software architect.
Design concepts – Design characteristics, Design elements, Design factors.

Module - II PROCESS OF SOFTWARE ARCHITECTURE 9

Introduction to architecture – Architecture definition process – Guiding principles - Process outcomes – Process context – Supporting activities – Architecture definition activities – Process exit criteria.
Scope – Concerns - Principles & Constraints – Selection & classes of stakeholders.
Identifying & using scenarios – Types of scenarios, Uses for scenarios, Identifying & prioritizing scenarios, Capturing scenarios, Applying & use of scenarios.
Software patterns, Styles, Patterns & Idioms.

Module – III ARCHITECTURE MODELS & DESCRIPTION 9

Producing architectural models – Need for models, Types of models, Modeling languages, Guidelines for creating effective models, Agile modeling techniques.
Creating architectural description: Properties of effective architectural description, Glossaries, IEEE standard, Contents of architectural description, Validating the architecture, Validation techniques.
Scenario based evaluation methods, Introduction to view point catalog and functional view point.

Module - IV DESIGN QUALITY 9

Software quality models, Effect of design on software quality and Quality attributes.
Design principles: Design roles, Design processes, Notion of software architecture, Software architecture style.
Description of software architectures: Visual notation and Client-server pair.

Module – V APPROACHES, ARCHITECTURAL & DESIGN STYLES 9

Typical architecture styles – Data flow, Independent components, Call & return, Data centred and Virtual machine.
Design styles – Choices of styles, Combinations of styles and case study on keyword frequency vector.
Approaches: Structured Systems Analysis and Structured Design (SSA/SD), Jackson Structured Programming (JSP) and Jackson System Development (JSD).

Total Hours: 45

3. Course outcomes:

- *The student will have knowledge about the software architecture process and architecture models.*
- *The student will be able to develop software systems with suitable design.*
- *The student will have knowledge about the design styles and design approaches.*
- *The student will be able to develop models and documents confirming to standards.*

REFERENCES:

1. Nick Rozanski, Eoin Woods, “Software Systems Architecture – Working with Stakeholders Using Viewpoints and Perspectives”, Pearson Education, 2009.
2. Hong Zhu, “Software Design Methodology – From Principles to Architectural Styles”, An imprint of Elsevier, 2005.
3. David Budgen, “Software Design”, Pearson Education, 2003.
4. Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice”, Second Edition, Pearson Education, 2004.
5. Eric J. Braude, “Software Design: From Programming to Architecture”, John Wiley & Sons, 2004.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PN606	AGILE SOFTWARE PROCESS	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand the basic concepts of Agile Software Process.
- To gain knowledge in the area of various Agile Methodologies.
- To develop Agile Software Process
- To know the principles of Agile Testing

2. Course pre-requisites : Software Engineering

Module – I INTRODUCTION

9

Software is new product development – Iterative development – Risk (Driven and Client(Driven iterative planning – Time boxed iterative development – During the iteration, No changes from external stakeholders – Evolutionary and adaptive development (Evolutionary requirements analysis – Early “Top Ten” high(level requirements and skilful analysis – Evolutionary and adaptive planning – Incremental delivery – Evolutionary delivery – The most common mistake – Specific iterative and Evolutionary methods

Module – II AGILE AND ITS SIGNIFICANCE

9

Agile development – Classification of methods – The agile manifesto and principles – Agile project management – Embrace communication and feedback – Simple practices and project tools – Empirical Vs defined and prescriptive process – Principle(based versus Rule(Based – Sustainable discipline: The human touch – Team as a complex adaptive system – Agile hype – Specific agile methods. The facts of change on software projects –Key motivations for iterative development – Meeting the requirements challenge iteratively – Problems with the waterfall. Research evidence – Early historical project evidence – Standards(Body evidence – Expert and thought leader evidence – A Business case for iterative development – The historical accident of waterfall validity.

Module – III AGILE METHODOLOGY

9

Method overview – Lifecycle – Work products, Roles and Practices values – Common mistakes and misunderstandings – Sample projects – Process mixtures - Adoption strategies – Fact versus fantasy – Strengths versus “Other” history.

Module – IV CASE STUDY

9

Agile – Motivation – Evidence – Scrum – Extreme Programming – Unified Process - Evo – Practice Tips

Module – V AGILE PRACTICING AND TESTING

9

Project management – Environment – Requirements – Test – The agile alliances - The manifesto – Supporting the values – Agile testing – Nine principles and six concrete practices for testing on agile teams.

STATE OF ART (Not for Exam)

Agile software planning, estimation

Total Hours: 45

3. Course outcomes:

- *Ability to understand basic concepts of Agile Software Process.*
- *To gain knowledge in the area of various Agile Methodologies.*
- *Ability To develop Agile Software Process*
- *To understand the principles of Agile Testing*

REFERENCES:

1. Elisabeth Hendrickson, “Agile Testing” Quality Tree Software Inc 2008.
2. Craig Larman “Agile and Iterative Development – A Manager’s Guide” Pearson Education – 2004.
3. Alistair “Agile Software Development series” Cockburn 2001.
4. www.agileintro.wordpress.com/2008
5. www.serena.com/docs/repository/solutions/introtoagiledevel.pdf
6. www.qualitytree.com

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ607 / 15PN607	DISTRIBUTED COMPUTING	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To list the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions.
- To recognize how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems.
- To design a distributed system that fulfills requirements with regards to key distributed computing properties (such as scalability, transparency, etc.), be able to recognize when this is not possible, and explain why.
- To build distributed system software using basic OS mechanisms as well as higher-level middleware and languages.

2. Course pre-requisites : Operating Systems, Computer Networks

Module – I INTRODUCTION TO DISTRIBUTED SYSTEMS 8

Characterization of Distributed Systems- Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models- Introduction, Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication. Distributed objects and Remote Invocation- Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

Module – II OPERATING SYSTEM SUPPORT AND DISTRIBUTED FILE SYSTEMS 8

Operating System Support- Introduction, OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture, Distributed File Systems-Introduction, File Service architecture, case study- SUN network file systems. Name Services-Introduction; Name Services and the Domain Name System, Case study of the Global Name Service, Case study of the X.500 Directory Service.

Module – III TIME AND GLOBAL STATES 10

Peer to Peer Systems-Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies-Squirrel, OceanStore. Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging. Coordination and Agreement - Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

Module – IV DISTRIBUTED TRANSACTIONS 10

Transactions and Concurrency control - Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency controls. Distributed Transactions - Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery, Replication-Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data.

Module – V SECURITY TECHNIQUES

9

Security - Introduction, Overview of Security techniques, Cryptographic algorithms, Digital signatures, Case studies-Kerberos, TLS, 802.11 WiFi. Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy case study, Release consistency and Munin case study, other consistency models, CORBA case study- Introduction, CORBA RMI, CORBA Services.

STATE OF ART (Not for Exam)

Designing distributed systems: Google case study

Total Hours: 45

3. *Course outcomes:*

- *Identify models of distributed computing.*
- *Analyze algorithms for coordination, communication, security and synchronization in distributed systems.*
- *Classify distributed shared memory models.*
- *Design and implement distributed file systems.*
- *Design distributed algorithms for deadlocks.*

REFERENCES:

1. G Coulouris, J Dollimore and T Kindberg, “Distributed Systems Concepts and Design”, Fourth Edition, Pearson Education.
2. S.Ghosh, Chapman & Hall/CRC, “Distributed Systems”, Taylor & Francis Group, 2010.
3. S.Mahajan and S.Shah, “Distributed Computing”, Oxford University Press.
4. Pradeep K.Sinha, “Distributed Operating Systems Concepts and Design”, PHI.
5. M Singhal, N G Shivarathri, “Advanced Concepts in Operating Systems”, Tata McGraw-Hill Edition.
6. K.P.Birman, ”Reliable Distributed Systems”, Springer.
7. A.S. Tanenbaum and M.V. Steen, “Distributed Systems – Principles and Paradigms”, Pearson Education.
8. R.Chow, T.Johnson, “Distributed Operating Systems and Algorithm Analysis”, Pearson.
9. A.S.Tanenbaum, “Distributed Operating Systems”, Pearson education.
10. Ajay D. Kshemakalyani & Mukesh Singhal, “Distributed Computing, Principles, Algorithms and Systems”, Cambridge, 2010

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ608 / 15PN608	SERVICE ORIENTED ARCHITECTURE	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To gain understanding of the basic principles of service orientation
- To learn service oriented analysis techniques
- To learn technology underlying the service design
- To learn advanced concepts such as service composition, orchestration and Choreography
- To know about various WS-* specification standards

2. Course pre-requisites : Internet Programming, OOPS

MODULE I SOA Evolution – Architecture 9

Roots of SOA – Characteristics of SOA - Comparing SOA to client-server and distributed internet architectures – Anatomy of SOA- How components in an SOA interrelate - Principles of service orientation

MODULE II Web services and Primitive SOA 9

Web services – Service descriptions – Messaging with SOAP –Message exchange Patterns – Coordination –Atomic Transactions – Business activities – Orchestration – Choreography - Service layer abstraction – Application Service Layer – Business Service Layer – Orchestration Service Layer.

MODULE III Service Oriented Analysis 9

Service oriented analysis – Business-centric SOA – Deriving business services- service modeling - Service Oriented Design – WSDL basics – SOAP basics – SOA composition guidelines – Entity-centric business service design – Application service design – Taskcentric business service design

MODULE IV SOA platforms 9

SOA platform basics – SOA support in J2EE – Java API for XML-based web services (JAX-WS) - Java architecture for XML binding (JAXB) – Java API for XML Registries (JAXR) - Java API for XML based RPC (JAX-RPC)- Web Services Interoperability Technologies (WSIT) - SOA support in .NET – Common Language Runtime - ASP.NET web forms – ASP.NET web services – Web Services Enhancements (WSE).

MODULE V WS Extensions 9

WS-BPEL basics – WS-Coordination overview - WS-Choreography, WS-Policy, WS Security

STATE OF ART (Not for Exam)

Study of Web application that use SOA – redbus,expedia,tripadvisor

Total Hours: 45

3. Course outcomes:

- Know the SOA concepts and architecture
- Understand principles of Web services and SOA.
- Steps in Service oriented Analysis
- Study the concepts SOA platforms and WS standard extensions.

REFERENCES:

1. Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2005.
2. Thomas Erl, "SOA Principles of Service Design "(The Prentice Hall Service-Oriented Computing Series from Thomas Erl), 2005.
3. Newcomer, Lomow, "Understanding SOA with Web Services", Pearson Education, 2005.
4. Sandeep Chatterjee, James Webber, "Developing Enterprise Web Services, An Architect's Guide", Pearson Education, 2005.
5. Dan Woods and Thomas Mattern, " Enterprise SOA Designing IT for Business Innovation" O'REILLY, First Edition, 2006.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ609 / 15PN609	XML AND WEB SERVICES	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To provide an in-depth knowledge of XML and Web Services.
- To gain knowledge about SOAP, UDDI and XML to create Web Services.
- To design Web service Architecture.
- To Study Building Blocks of Web services.
- To understand XML security issues.
- To develop and deploy Web Service Applications.

2. Course pre-requisites : Computer Networks

Module – I INTRODUCTION TO XML TECHNOLOGY 9

XML – benefits – Advantages of XML over HTML – EDL –Databases – XML based standards – DTD – XML Schemas – X- Files – XML processing – DOM –SAX.

Module – II PRESENTATION TECHNOLOGIES 9

Presentation technologies – XSL – XFORMS – XHTML – voice XML – Transformation – XSLT – XLINK – XPATH –XQ.

Module – III WEB SERVICES BUILDING BLOCK 9

Transport protocols for web services – messaging with web services – protocols – SOAP – describing web services – WSDL – Anatomy of WSDL – manipulating WSDL – web service policy – Discovering web services – UDDI – Anatomy of UDDI- Web service inspection – Ad-Hoc Discovery – Securing web services.

Module – IV IMPLEMENTING XML IN E-BUSINESS 9

B2B - B2C Applications – Different types of B2B interaction – Components of e-business XML systems – ebXML – Rosetta Net Applied XML in vertical industry – Web services for mobile devices.

Module – V DEPLOY WEBSERVICES USING .NET/JAVA 9

Develop and Deploy Web Services – Global Weather Forecast - Current weather and weather conditions for major cities around the world. Translate Service - Convert text from one language to another language. ISBN Information Retrieval - Book Information web services by ISBN. Country Details - Get Currency, Currency code, International Dialing code, ISO country code for all countries.

STATE OF ART (Not for Exam)

Responsive Web Design - Twitter Bootstrap and Zurb Foundation.

Total Hours: 45

3. Course outcomes:

- Understand the fundamental elements in Web Technology and XML services.
- Understand Web Services and its Infrastructure.
- Building a Web Service.
- Deploying and Publishing Web Services.

REFERENCES:

1. Ron schmelzer et al, "XML and Web Services", Pearson Education, 2002.
2. Sandeep Chatterjee and James Webber, "Developing Enterprise Web Services: An Architect's Guide", Prentice Hall, 2004.
3. Frank P. Coyle, "XML, Web Services and the Data Revolution", Pearson Education, 2002.
4. Keith Ballinger, ".NET Web Services Architecture and Implementation", Pearson Education,2003.
5. Henry Bequet and Meeraj Kunnumpurath, "Beginning Java Web Services", Apress, 2004.
6. Russ Basiura and Mike Batongbacal, "Professional ASP.NET Web Services", Apress,2.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ610 / 15PN610	MOBILE APPLICATION DEVELOPEMENT	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- Understand system requirements for mobile applications
- Generate suitable design using specific mobile development frameworks
- Generate mobile application design
- Implement the design using specific mobile development frameworks
- Deploy the mobile applications in marketplace for distribution

2. Course pre-requisites : NIL

UNIT I INTRODUCTION 5

Introduction-Need for mobile applications – Cost of Development – Importance of Mobile strategies in the Business world-Market and business drivers for mobile application- Requirements gathering and validation for mobile applications- –Mobile Myths, Third party framework – Publishing and delivery of Mobile Applications-Marketing.

UNIT II BASIC DESIGN 8

Introduction –Basics of embedded system design - Embedded OS - Design constrains for mobile applications, both hardware and software related-Architecting mobile applications- Understanding Mobile Application Users - Effective Use of Screen Real Estate – User interface for mobile applications-touch events and gestures-Using the Tools of Mobile Interface Design

UNIT III ADVANCED DESIGN 8

Designing applications with multimedia and web access capabilities – Integration with GPS and Social media networking applications-Accessing applications hosted in a cloud computing environment – Design pattern for mobile applications.

UNIT IV TECHNOLOGY I -A NDROID 12

Introduction– Establishing the development environment –Android architecture-Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment- Interaction with server side applications- Using Google Maps, GPS and Wifi –Integration with social media applications.

UNIT V TECHNOLOGY II - IOS 12

Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using Wifi -iPhone marketplace.

STATE OF ART (NOT FOR EXAM)
Testing Methodologies for Mobile Application.

Total Hours: 45

3. Course outcomes:

- Describe the requirements for mobile applications
- Explain the challenges in mobile application design and development
- Develop design for mobile applications for specific requirements
- Implement the design using Android SDK
- Implement the design using Objective C and iOS
- Deploy mobile applications in Android and iPhone marketplace for distribution

REFERENCES:

1. <http://developer.android.com/develop/index.html>
2. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012
3. Charlie Collins, Michael Galpin and Matthias Kappler, "Android in Practice", DreamTech, 2012
4. Wei-Meng Lee "Beginning iPhone SDK Programming with Objective-C", Wrox, 2010

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ611 / 15PN611	USER INTERFACE DESIGN	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand the basics of User Interface Design.
- To design the user interface, design, menu creation and windows creation
- To understand the concept of menus, windows, interfaces, business functions, various problems in windows design with colour, text, Non-anthropomorphic Design.
- To study the design process and evaluations.

2. Course pre-requisites : Software Engineering

Module - I INTERACTIVE SOFTWARE AND INTERACTION DEVICE 9
 Human-Computer Interface – Characteristics Of Graphics Interface –Direct Manipulation
 Graphical System – Web User Interface –Popularity –Characteristic & Principles.

Module - II HUMAN COMPUTER INTERACTION 9
 User Interface Design Process – Obstacles –Usability –Human Characteristics In Design –Human
 Interaction Speed –Business Functions –Requirement Analysis – Direct – Indirect Methods – Basic
 Business Functions – Design Standards – General Design Principles –Conceptual Model Design –
 Conceptual Model Mock-Ups

Module - III WINDOWS 9
 Characteristics– Components– Presentation Styles– Types– Managements– Organizations– Operations–
 Web Systems– System Timings - Device– Based Controls Characteristics– Screen – Based Controls —
 Human Consideration In Screen Design – Structures Of Menus –Functions Of Menus– Contents Of
 Menu– Formatting – Phrasing The Menu – Selecting Menu Choice– Navigating Menus– Graphical Menus.
 Operate Control – Text Boxes– Selection Control– Combination Control– Custom Control– Presentation
 Control.

Module - IV MULTIMEDIA 9
 Text For Web Pages – Effective Feedback– Guidance & Assistance– Internationalization– Accessibility–
 Icons– Image– Multimedia – Coloring- Case Study: Addressing usability in Ecommerce sites

Module - V DESIGN PROCESS AND EVALUATION 9
 User Interface Design Process - Usability Testing - Usability Requirements and Specification procedures
 and techniques- User Interface Design Evaluation

STATE OF ART (Not for Exam)

GUI hardware options- Smart systems design- Expert system design,

Total Hours: 45

3. *Course outcomes:*

- *Knowledge on development methodologies, evaluation techniques and user interface building tools.*
- *Explore a representative range of design guidelines.*
- *Gain experience in applying design guidelines to user interface design tasks.*
- *Ability to design their own Human Computer.*

REFERENCES

1. Wilbent. O. Galitz ,“The Essential Guide To User Interface Design”, John Wiley& Sons, 2001.
2. Deborah Mayhew, The Usability Engineering Lifecycle, Morgan Kaufmann, 1999Ben.
3. Shneiderman, “Design The User Interface”, Pearson Education, 1998.
4. Alan Cooper, “The Essential Of User Interface Design”, Wiley – Dream Tech Ltd., 2002.
5. Sharp, Rogers, Preece, ‘Interaction Design’, Wiley India Edition, 2007
6. Alan Dix et al, " Human - Computer Interaction ", Prentice Hall, 1993.
7. Ben Schneiderman, " Designing the User Interface ", Addison Wesley, 2000.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ612 / 15PN612	SOCIAL NETWORK ANALYSIS	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand the components of the social network.
- To model and visualize the social network
- To mine the users in the social network
- To understand the evolution of the social network.
- To mine the interest of the user

2. Course pre-requisites : NIL

Module – I INTRODUCTION 9

Introduction to Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web – Statistical Properties of Social Networks -Network analysis - Development of Social Network Analysis - Key concepts and measures in network analysis - Discussion networks - Blogs and online communities - Web-based networks

Module – II MODELING AND VISUALIZATION 9

Visualizing Online Social Networks - A Taxonomy of Visualizations - Graph Representation - Centrality-Clustering - Node-Edge Diagrams - Visualizing Social Networks with Matrix-Based Representations-Node-Link Diagrams - Hybrid Representations - Modelling and aggregating social network data – Random Walks and their Applications –Use of Hadoop and Map Reduce - Ontological representation of social individuals and relationships.

Module – III MINING COMMUNITIES 9

Aggregating and reasoning with social network data, Advanced Representations - Extracting evolution of Web Community from a Series of Web Archive - Detecting Communities in Social Networks - Evaluating Communities – Core Methods for Community Detection & Mining - Applications of Community Mining Algorithms - Node Classification in Social Networks.

Module – IV EVOLUTION 9

Evolution in Social Networks – Framework - Tracing Smoothly Evolving Communities - Models and Algorithms for Social Influence Analysis - Influence Related Statistics - Social Similarity and Influence - Influence Maximization in Viral Marketing - Algorithms and Systems for Expert Location in Social Networks - Expert Location without Graph Constraints - with Score Propagation – Expert Team Formation - Link Prediction in Social Networks - Feature based Link Prediction - Bayesian Probabilistic Models - Probabilistic Relational Models

Module – V TEXT AND OPINION MINING 9

Text Mining in Social Networks -Opinion extraction – Sentiment classification and clustering - Temporal sentiment analysis - Irony detection in opinion mining - Wish analysis - Product review mining – Review Classification – Tracking sentiments towards topics over time

STATE OF ART (Not for Exam) :
Social Roles of Social Network Analysis

Total Hours: 45

3. *Course outcomes:*

- *Work on the internal components of the social network Model and visualize the social network*
- *Mine the behaviour of the users in the social network Predict the possible next outcome of the social network Mine the opinion of the user*

REFERENCES:

1. Charu C. Aggarwal, "Social Network Data Analytics", Springer; 2011
2. Peter Mika, "Social Networks and the Semantic Web", Springer, 1st edition, 2007.
3. Borko Furht, "Handbook of Social Network Technologies and Applications", Springer, 1st edition, 2010.
4. Guandong Xu , Yanchun Zhang and Lin Li, "Web Mining and Social Networking – Techniques and applications", Springer, 1st edition, 2011.
5. Giles, Mark Smith, John Yen, "Advances in Social Network Mining and Analysis", Springer, 2010.
6. Ajith Abraham, Aboul Ella Hassanien, Václav Snášel, "Computational Social Network Analysis: Trends, Tools and Research Advances", Springer, 2009.
7. Toby Segaran, "Programming Collective Intelligence", O'Reilly, 2012

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ613 / 15PN613	DATA WAREHOUSING AND MINING	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand data warehousing and various data mining techniques
- To acquire knowledge in pattern mining.
- To perform data classification and clustering.
- To detect outliers based on classification and clustering.
- To gain knowledge about the emerging trends in data mining.

2. Course pre-requisites : NIL

Module – I DATA WAREHOUSING CONCEPTS 9

Basic Concepts – Architecture – Data warehouse modeling – Data cube and OLAP – Data warehouse design and usage – Framework for data warehouse design – Data warehouse design process - Data warehouse implementation – Efficient data cube computation – Indexing OLAP data – Efficient processing of OLAP queries – OLAP server architectures.

Module – II FUNDAMENTALS OF DATA MINING AND PATTERN MINING 9

Data Mining Functionalities – Data Preprocessing – Data Cleaning – Data Integration– Data Reduction – Data Transformation and Data Discretization - Mining frequent patterns, associations and correlations – Basic concepts – Frequent itemset mining – Pattern evaluation methods – Advanced pattern mining – Pattern mining in multilevel, multidimensional space – Constraint based frequent pattern mining.

Module – III CLASSIFICATION METHODS 9

Decision Tree Induction – Bayesian Classification – Bayesian belief networks - Rule Based Classification – Classification by Back propagation – Support Vector Machines – Classification using frequent patterns – Lazy Learners – Other Classification Methods – Model evaluation and selection – Techniques to improve classification accuracy.

Module IV CLUSTER ANALYSIS AND OUTLIER DETECTION 9

Cluster Analysis - Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Evaluation of clustering - Clustering High-Dimensional Data – Constraint- Based Cluster Analysis - Outliers and outlier analysis – Outlier detection methods – Proximity based approaches – Clustering based approaches – Classification based approaches

Module – V CURRENT TRENDS 9

Mining complex data types - Graph mining – Temporal data mining – Spatial data mining – Distributed data mining – Privacy, security and legal aspects of data mining – Data mining applications – Financial data analysis – Telecommunication industry – Retail industry – Health care and biomedical research.

STATE OF ART (Not for Exam)

Machine Learning - Big data

Total Hours: 45

3. *Course outcomes:*

- *Ability to design a data warehouse*
- *Ability to perform data preprocessing*
- *Ability to evaluate classification and clustering methods*
- *Ability to mine various complex data*

REFERENCES:

1. Jiawei Han, Micheline Kamber and Jian Pei, “Data Mining Concepts and Techniques” Third Edition, Elsevier, 2012.
2. M. Kantardzic, “Data Mining: Concepts, Models, Methods, and Algorithms”, 2nd edition, Wiley-IEEE Press, 2011.
3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar “Introduction to Data Mining”, Pearson Education, 2007.
4. Alex Berson and Stephen J. Smith “Data Warehousing, Data Mining & OLAP”, Tata McGraw – Hill Edition, Tenth Reprint 2007.
5. K.P. Soman, Shyam Diwakar and V. Ajay “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ614 / 15PN614	INTERNET OF THINGS	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To get acquainted with the building blocks of Internet of Things (IoTs), characteristics and taxonomy of IoT levels
- To learn a generic design methodology and programming aspects of IoT.
- To gain knowledge on the real world applications of IoT.
- To know about various packages, frameworks and cloud services
- To get acquainted with data analytics for IoT.

2. Course pre-requisites : Web Technology, Internet Programming

Module – I INTRODUCTION TO IoT 9

Definition and Characteristics – Physical Design Things – Protocols - Logical Design – Functional Blocks – Communication Models – Communication APIs – Introduction to measure the physical quantities – IoT Enabling Technologies – Wireless Sensor Networks - Cloud Computing – Big Data Analytics – Communication Protocols – Embedded Systems – IoT Levels and Deployment Templates

Module –II DEVELOPING INTERNET OF THINGS 9

Introduction to Smart Systems using IoT – IoT Design Methodology – Case Study: Weather Monitoring – Logical Design using Python – Data types & Data Structures – Control Flow – Functions – Modules – Packages – File Handling – Date/Time Operations – Classes – Python Packages of Interest for IoT

Module – III DOMAIN SPECIFIC IoTS 9

Home Automation – Cities – Environment – Energy – Retail – Logistics – Agriculture – Industry – Health and Lifestyle – IoT and M2M

Module – IV IoT PHYSICAL DEVICES, ENDPOINTS AND CLOUD OFFERINGS 9

IoT Device – Raspberry Pi – Interfaces – Programming Raspberry Pi with Python – Other IoT Devices – IoT Physical Servers and Cloud Offerings – Cloud Storage Models and communication APIs – WAMP – Xively Cloud – Django – Amazon Web Services for IoT – SkyNet IoT Messaging Platform – Basics of Secure IoT Programming – Case Study: Home Automation.

Module – V DATA ANALYTICS FOR IoT 9

Introduction – Apache Hadoop –Using Hadoop MapReduce for Batch Data Analysis –Apache Oozie – Apache Spark – Apache Storm – Using Apache Storm for Real-time Data Analysis – Case Study: Structural Health Monitoring.

STATE OF ART (Not for Exam)

The Role of the Internet of Things for Increased Autonomy and Agility in Collaborative Production Environments

Total Hours: 45

3. *Course outcomes:*

- *Identify and design the new models for market strategic interaction*
- *Design business intelligence and information security for WoB*
- *Analyze various protocols for IoT*
- *Design a middleware for IoT*

REFERENCES:

1. Arshdeep Bahga, Vijay Madiseti, “Internet of Things: A Hands-On Approach”, Published by Arshdeep Bahga & Vijay Madiseti, 2014
2. Smart Things: Ubiquitous Computing User Experience Design. Mike Kuniavsky. Morgan Kaufmann Publishers. 2010.
3. Meta Products: Building the Internet of Things. Sara Cordoba, Wimer Hazenberg, Menno Huisman. BIS Publishers. 2011.
4. Getting Started with Arduino (Make: Projects). Massimo Banzi. O'Reilly Media. 2008.
5. Emotional Design: Why We Love (or Hate) Everyday Things. Donald A. Norman. Basic Books, 2004.
6. Physical Computing: Sensing and Controlling the Physical World with Computers. Tom Igoe, Dan O'Sullivan. Premier Press. 2004.

Course Code	Course Name	Contact Hours			
		L	T	P	C
15PQ615 / 15PN615	BIG DATA ANALYTICS	3	0	0	3

Distribution of marks in Question Paper – Theory (80%) and Problem (20%)

1. Course objectives:

- To understand the fundamental concepts of big data analytics.
- To learn to use various techniques for mining data stream.
- To explore the technologies associated with big data analytics such as NoSQL, Hadoop and Map Reduce

2. Course pre-requisites : Database Management System & Data mining

Module – I INTRODUCTION 9

Big data overview – State of the Practice in Analytics - Key Roles – Data Analytics Lifecycle – Discovery - Data Preparation - Model Planning - Model Building - Communicate Results – Operationalize

Module – II MINING DATA STREAMS 9

Stream Data Model - Sampling Data in a Stream–Filtering Streams–Counting Distinct Elements in a Stream–Estimating Moments–Counting Ones in a Window–Decaying Window

Module – III LARGE –SCALE FILE SYSTEMS AND MAP-REDUCE 9

Distributed File Systems – MapReduce - Algorithms Using MapReduce - Extensions to MapReduce - Communication Cost Model - Complexity Theory for MapReduce

Module IV NoSQL DATA MANAGEMENT 9

Introduction to NoSQL – aggregate data models – aggregates – key-value and document data models – relationships – graph databases – schemaless databases – materialized views – distribution models – sharding – master-slave replication – peer-peer replication – sharding and replication – consistency – relaxing consistency

Module – V OVERVIEW OF HADOOP 9

Hadoop Introduction - Data format – Analyzing data with Hadoop – Scaling out – Hadoop streaming – Hadoop pipes – Hadoop Distributed File System (HDFS) – HDFS Design – HDFS Concepts – Hadoop I/O – Data integrity – compression – serialization – sequence file – map file

STATE OF ART (Not for Exam)

Real time data analytics

Total Hours: 45

3. *Course outcomes:*

- *Ability to understand the concepts of big data analytics.*
- *Ability to mine data streams*
- *Ability to design schemaless database*
- *Ability to perform map reduce using hadoop*

REFERENCES:

1. EMC Education Services, “Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data”, Wiley, 2015.
2. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, Cambridge University Press, 2012.
3. P.J.Sadalage and M.Flower, “NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence”, Addison – Wesley Professional, 2013.
4. Tom White, “Hadoop: The Definitive Guide”, Third Edition, O’Reilly, 2012.
5. Paul C. Zikopoulos,Chris Eaton,Dirk deRoos,Thomas Deutsch ,George Lapis, “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill, 2012.