DIVISION

OF

M.Sc. (SOFTWARE) ENGINEERING

Code	Subject Name	Credit
SE201	Theory of Programming Language (CA202)	4:0:0
SE202	Software Engineering	4:0:0
SE203	Software Project Management	4:0:0
SE204	Unified Software Development	4:0:0
SE205	Distributed Operating System	3:0:0
SE206	Introduction to Information Technology (CA102)	3:0:0
SE207	Artificial Intelligence and Knowledge Engineering	3:0:0
SE208	Object Oriented Software Engineering	4:0:0
SE209	Design Patterns	3:1:0

SE201 THEORY OF PROGRAMMING LANGUAGES (CA202)

Credit 4:0:0
Marks: (40+60)

Unit : I
Defining syntax, Character set, BNF, syntax graphs, Syntax and program readability. Programming Languages, Variables expression and statements, assignment statement.

Unit : II
Binding time & storage allocation, constant & initialisation, expression, conditional statements. Iterative statement, Go to statement & labels, Types data types and typing, Enumerated types, elementary, pointers.

Unit : III
Structured data type, type coersion, type equivalence, Scope and extent, basic, run time implementation, binding scope & extent. Procedure, features.

Unit : IV
Parameter evaluation & parsing, Call by name, Specification of objects in a procedure, aliasing, overloading generic functions, coroutines. Data abstraction, Introduction, abstract data types, examples.

Unit : V
Exception handling, need & issues in exception handling.
Concurrency, Concepts, semaphores, monitors, message passing, input & output. Concepts & Difference between functional program, Data flow programs & Object oriented programs.

Text book
1. Ellis Horowirz, “Fundamentals of Programming Language”, Galgotia Publications, 1995.

Reference books:

1. Terrance W Pratt, "Theory of Programming Languages", Addition Wesley, 4th Edition, 1999
2. Ravi Sethi, "Computer in Programming Language", Addition Wesley, Inter edition, 1991

SE202 SOFTWARE ENGINEERING

Credit: 4:0:0
Marks: (40+60)

Unit I : The Product And Process

The product: the evolving role of software, software crisis, software myths. The process: Software engineering – layered technology, software process, software process models, linear sequential model, prototyping model, the RAD model, evolutionary process model, formal methods model, fourth generation techniques, process technology, product and process. Managing software projects

Project management concepts: management spectrum, people, problem, process, project. Software process and project metrics: measures metrics and indicators, metrics in the process and project domains, software measurement, reconciling different metrics approaches, metrics for software quality, integrating metrics within the software process.Software project planning: observations on estimating, project planning objectives, software scope, resources, software project estimation, decomposition techniques, empirical estimation models, make-buy decision, automated estimation tools.(Chapters 1 to 5)

Unit II : Risk Management

Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk mitigation, monitoring, and management, safety risks and hazards, RMMM plan. Project Scheduling and Tracking: basic concepts, relationship between people and effort, defining a task set for the software project, selecting software engineering tasks, refinement of major tasks, defining a task network, scheduling, project plan. Software Quality Assurance: quality concepts, quality movement, software quality assurance, software reviews, formal technical reviews, formal approaches to SQA, statistical quality assurance, software reliability, SQA plan, ISO 9000 quality standards. Software configuration management: SCM process, identification of objects in the software configuration version control, change control, configuration audit, status reporting, SCM standards.

Conventional methods for Software Engineering

System Engineering: computer based system, system engineering hierarchy, information engineering, information strategy planning, business area analysis, product engineering, modeling the system architecture, system modeling and simulation, system specification. (Chapters 6 to 10)

Unit : III

Analysis concepts and principles: requirement analysis, communication techniques, analysis principles, software prototyping, specification, specification review. Analysis modeling: elements of analysis model, data modeling, functional modeling and information flow, behavioral modeling, mechanics of structured analysis, data dictionary, overview of other classical analysis methods. Design concepts and principles: software design and software engineering, design process, design principles, design concepts, effective modular designing, design heuristics for effective modularity, design model, design documentation. (Chapters 11 to 13)

Unit : IV

Design methods: data design, architectural design, architectural design process, transform mapping, transaction mapping, design post processing, architectural design optimization, interface design, human computer interface design, interface design guidelines, procedural design. - Design for real time systems: system consideration, real time systems, analysis and simulation of real-time systems, real time design. (Chapters 14 &15)

Unit : V

Software testing methods: software testing fundamentals, test case design, white box testing, basis path testing, control structure testing, black box testing, testing for specialized environments. Software testing Strategies: strategic approach to software testing, strategic issues, Unit, integration, validation, system testing, art of debugging. Technical metrics of Software: software quality, framework for technical software metrics, metrics for the analysis model, metrics for the design, source code, testing and maintenance. (Chapters 16 to 18)

Text Book

1. Roger S Pressman**,** *Software Engineering, A practitioner's approach*, Fourth Edition, McGraw Hill International, 2000.

Reference Books

1. Ian Somerville**,** *Software Engineering*, Fifth Edition, Addison Wesley publications, 1996
2. Shoomaqn, 'Software Engineering' McGraw Hill, 1997

SE203 SOFTWARE PROJECT MANAGEMENT

Credit 4:0:0
Marks: (40+60)

Unit : I

Conventional Software Management: The Waterfall Model, Conventional Software Management Performance. Evolution of Software Economics: Software Economics, Pragmatic Software Cost Estimation. Improving Software Economics: Reducing Software Product Size, Improving Software Processes, Improving Team Effectiveness, Improving Automation through Software Environments, Achieving Required Quality, Peer Inspections-A Pragmatic View. The Old Way and the New: The Principles of Conventional Software Engineering, The Principles of Modern Software Management, and Transitioning to an Iterative Process. Life-Cycle Phases: Engineering and Production Stages, Inception Phase, Elaboration Phase, Construction Phase, Transition Phase
(Chapters 1 to 5)

Unit : II

Artifacts of the Process: The Artifact Sets, Management Artifacts, Engineering Artifacts, Pragmatic Artifacts. Model-Based Software Architectures: A Management Perspective, A Technical Perspective. Workflows of the Process: Software Process Workflows, Iteration Workflows. Checkpoints of the Process: Major Milestones, Minor Milestones, Periodic Status Assessments. Iterative Process Planning: Work Breakdown Structures, Planning Guidelines, The Cost and Schedule Estimating Process, The Iteration Planning Process, Pragmatic Planning. Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, Evolution of Organizations. (Chapters 6 to 11)

Unit : III

Process Automation: Tools- Automation Building Blocks, The Project Environment. Project Control and Process Instrumentation: The Seven Core Metrics, Management Indicators, Quality Indicators, Life-Cycle Expectations, Pragmatic Software Metrics, Metrics Automation. Tailoring the Process: Process Discriminants, Example- Small-Scale Project versus Large-Scale Project. Modern Project Profiles: Continuous Integration, Early Risk Resolution, Evolutionary Requirements, Teamwork among Stakeholders, Top10 Software Management Best Practices. Next-Generation Software Economics: Next-Generation Cost Models, Modern Software Economics. Modern Process Transitions: Culture Shifts, Denouement. (Chapters 12 to 17)

Unit : IV

The State of the Practice in Software Management, The COCOMO Cost Estimation Model: COCOMO, Ada COCOMO, COCOMO II. Change Metrics: Metrics Deviation, Pragmatic Change Metrics. Process Improvement and Mapping to the CMM: CMM Overview, Pragmatic Process Improvement, Maturity Questionnaire, Questions Not Asked by the Maturity Questionnaire, Overall Process Assessment. (Appendices A, B, C, E)

Unit : V

CCPDS-R Case Study: Context for the Case Study, Common Subsystem Overview, Project Organization, Common Subsystem, Product Overview, Process Overview, Demonstration-Based Assessment, Core Metrics, Other Metrics, People Factors, Conclusions. (Appendix D)

Text Book

1. Walker Royce, *Software Project Management – A unified Framework*, Addison-Wesley, 1998

Reference Books

1. Mike Cotterell, Bob Hughes, 'Software Project Management'. Inclination, Thomas Computer Press, 1995.
2. Darrel Ince, H. Sharp and M. Woodman, 'Introduction to Software Project Management and Quality Assurance', Tata MacGraw Hill 1995.

SE204 UNIFIED SOFTWARE DEVELOPMENT

Credit 4:0:0
Marks: (40+60)

Unit I : The Unified process
Unified process in a nutshell, unified process is use-case driven, Architecture-centric, life of the unified process, integrated process. Four Ps: People are crucial, projects make the product, product is more than code, process directs projects, tools are integral to process. Use-Case-Driven process: Use-case-driven development in brief, Capturing the use cases, Analysis, design, and implementation to realize use cases testing. An Architecture-centric process: Architecture in brief, use cases and architecture, steps to an architecture, architecture description, three interesting concepts.

Unit II : An interactive and incremental process
Iterative and incremental in brief, iterative approach is risk-drive, generic iteration, result of iteration is an increment models evolve from iterations, iterations challenge the organization. Core Workflows: From vision to requirements, purpose of the requirement workflow, overview of requirements capture, understanding the system context using a domain model, business model.

Unit III : Analysis
Introduction, analysis in brief, role of analysis, artifacts, workers, workflow of analysis Design: Introduction, role of design, artifacts, workers, workflow, implementation: Introduction, role of implementation, artifacts, workers, workflow Test : Introduction, role of testing, artifacts, workers, workflow

Unit IV : The generic iteration workflow
Need for balance, phases are the first division of work, generic iteration, planning precedes doing, risks affect project planning, use-case prioritization, resource needed, assess the iterations and phases, Inception launches the project: Inception phase in brief, archetypal inception, execute the core workflows, requirements to test Elaboration phase makes the architectural baseline: Elaboration phase in brief, early in the elaboration phase, archetypal elaboration, execute the core Workflows, requirements to test, asses the iterations, planning the construction phase.

Unit V : Construction leads to initial operational capability
The construction phase in brief, early in the construction phase archetypal construction iteration workflow execute the core workflow, requirements to test, Controlling the business case, assess the iterations, planning, key deliverables Transition completes product release: The transition phase in brief, early in the transition phase , core workflows play a small role in this phase, completing the business case, assess the transition phase, planning, key deliverables Making the unified process work: The unified process helps you deal with complexity, major themes, management leads conversion unified process, specializing relate to the boarder community, get the benefits of the unified process.

Text Book

1. Ivar Jacobson, Grady booch, James Rumbaugh, *The Unified Software Development Process*, Mc Graw Hill, 1999.

Reference Book

1. Ali Bahrami, *Object Oriented Systems Development*, International edition, Mc-Graw Hill, 1999.
2. WOEF Gang Pree, 'Design Patterns for Object Oriented Software Development' First Indian Reprint, 2000

SE205 DISTRIBUTED OPERATING SYSTEM

Credit 3:0:0
Marks: (40+60)

Unit I : Operating System Concepts

Process, files, system calls, shell, OS structure – Monolithic, layered Virtual Machines, Client server model; Process – Introduction to process, Inter process communication, process scheduling.

Unit : II

File Systems – File And Directories; File system implementation – security – Introduction to distributed system – Goals – Hardware concepts – Software concepts – design Issues.

Unit : III

Communication in distributed systems – layered protocols, client server model, Remote procedure call, Group communication; Synchronization in distributed systems – clock synchronization – manual exclusion – Election algorithm – Atomic transition – Dead lock in distributed systems.

Unit : IV

Process & Processor in distributed systems – Threads, system models, Processor allocation – Distributed file system design – Implementation and trends in distributed file systems.

Unit V : Case Studies

AMOEBA; MACH

Text Book:

1. Andrew S. Tanenbaum, **Modern Operating System**, Prentice Hall of India, 1996

Reference Books:

1. William stalling, **Operating Systems**, Prentice Hall of India, 1995.
2. Harvey M Dietel, **Operating Systems**, 2nd Edition, Addison Wesley, 1990

SE206 INTRODUCTION TO INFORMATION TECHNOLOGY (CA102)

Credit 3:0:0
Marks : (40 + 60)

Unit I : Software
Application software system software basics of operating system and networks operating objectives – operating system as a user/computer interface, operating system a a resource manager - evolution of operating systems -serial processing, simple batch systems, multiprogrammed batch systems, timesharing systems, Parallel systems, distributed Systems, rea1 time systems DOS, UNIX, windows

Unit : II
Network Technologies and Concepts – Types of networks, communication media, communication processor -communication methods, standards and protocols -internet - client/server - c1ient/ server as a business model - types of client / server. implementa1ion, distributed data in client / server environment, advantages and disadvantages of client / server.

Unit : III
Fundamentals of information Technology. Information technology -information in the new business, a new key business resource, information in an organisation, transaction processing and customer integrated Sytems, Management information systems, Work group support systems, decision support systems and artificial intelligence, executive information systems, interorganisationa1 systems.

Unit : IV
System Analysis : Definitions of systems analysis - starting a project - generating a broad alternative Solution - Economic feasibility – tools and methods of gathering information. System deisgn: Data flow diagrams - user interface design -program design .Change over procedures -performance evaluation quality assurance : Inceptions and walk throughs testing.

Unit : V
Introduction to database and database management systems: Relational database model – entity classes and primary keys, defining relationship among entities, Structure of relational database – relational commercial language – SQL – SQL Integrity constraints , referential integrity – functional dependencies – assertions – triggers – Relational database design: Normalization

Text Books
1. Haag. Cummings, "Management Information Systems for the Information Age „ Irwin McGraw Hill, 1998.

Reference Rooks
1. V. Rajaraman, Radhakrishanan, 'An Introduction to Digital Computers Design', PHI, 1997
2. Andrews S Tannenbaum, 'Modern Operating Systems', second edition, 1996
3. Bibim C Desai, 'An Introduction to Database Systems', West publishing company
4. Steven Alter, 'Information Systems', Third Edition 2000.

SE207 ARTIFICIAL INTELLIGENCE AND KNOWLEDGE ENGINEERING

Credit 3:0:0
Marks: (40+60)

Unit : I
AI –Introduction, reactive machines – Stimulus response agent: Neural Network, machine Evaluation, state machines, Robot vision.

Unit : II
Search in state spaces – agents that plan, uniformed search, heuristic search, Planning, acting, learning, alternative search formulation & application

Unit : III
Search in state spaces-search knowledge representation of reasoning- propositional calculus, predicate calculus, resolution in predicate calculus.

Unit : IV
Knowledge representation & reasoning –Knowledge based systems, Representing commonsense knowledge, reasoning with uncertain information, Learning & acting with Bayer Nets.

Unit : V
Planning methods Based on logic – situation calculus, planning communication & integration – multiple agents, Communication among agents, Agents architectures.

Text Book
1. Nils . J. Nillson , Artificial Intelligence – A New Synthesis , Magam Kaufmenn, 1998.

Reference Books
1. Nils . J. Nillson , Problem Solving Methods in Artificial Intelligence
2. Nils . J. Nillson , Principles of Artificial Intelligence, Narosa Publication, 1990
3. P.H. Winston , Artificial intelligence ,Third edition.
4. E. Charniak & Medemih, Introduction to Artificial Intelligence

SE208 OBJECT ORIENTED SOFTWARE ENGINEERING

Credit: 4:0:0
Marks: 40 + 60

Unit : I
The software process: client, developer and user – requirement phase – specification phase – design phase – implementation phase – integration phase – maintenance phase – retirement –

problems with software production: essence and accidents – improving the software process – capability maturity models – ISO 9000 – SPICE – costs and benefits of software process improvement- Software Life-cycle models: Build- and- fix model –waterfall model – rapid prototyping model – incremental model – synchronize–and-stabilize model –spiral model – object-oriented life-cycle models –comparison of life-cycle models-Teams and Tools of their Trade : team organization – democratic team approach – classical chief programmer – beyond chief programmer and democratic teams – synchronize-and-stabilize teams- stepwise refinement – cost –benefit analysis –software metrics – CASE – taxonomy of CASE – scope of CASE – software versions – configuration control – build tools – productivity gains with CASE technology.

Unit : II
Testing: Quality Issues – Non execution-based Testing – execution-based testing – what should be tested? – Testing versus correctness proves – who should perform execution-based testing? –Testing distributed software – testing real-time software- when testing stops? – Introduction to objects: what is a module? – Cohesion – coupling – data encapsulation – abstract data types – information hiding – objects –inheritance, polymorphism, dynamic binding –cohesion and coupling of objects-Reusability, portability and interoperability: reuse concepts – impediments to reuse – reuse case studies – objects and productivity – reuse during the design and implementation phases – reuse and maintenance – portability – why portability? – Techniques for achieving portability –interoperability – future trends in interoperability.

Unit : III
Planning and Estimating: planning and the software process – estimating duration and cost – components of a software project management plan – software project management plan framework – IEEE software project management plan –planning of testing –planning of object-oriented projects –training requirements – documentation standards – case tools for planning and estimating- testing the software project management plan - The Phases of the software life cycle: Requirements phase: requirements analysis techniques – rapid prototyping – human factors – rapid prototyping as a specification technique – reusing the rapid prototype – other uses of prototyping –management implications of the rapid prototyping model-experiences with rapid prototyping - Joint Application Design (JAD) –Comparison of requirements analysis techniques – testing during the requirements phase – case tools for the requirements phase – metrics for the requirements phase –osbert oglesby case study : requirement phase - osbert oglesby case study:rapid prototype – object oriented requirements? – Specification phase: the specification document – informal specifications – structured system analysis –other semiformal techniques – entity-relationship modeling- -finite state machines – petri nets- Z 357 – other formal techniques – comparison of specification techniques – testing during the specification phase – CASE tools for the specification phase – metrics for the specification phase - osbert oglesby case study: structured system analysis - osbert oglesby case study: software project management plan.

Unit : IV
Object oriented analysis phase: object oriented versus structured paradigm - Object oriented analysis –elevator problem: Object oriented analysis- use case modeling – class modeling –

dynamic modeling – testing during the Object oriented analysis phase – CASE tools for the Object oriented analysis phase - osbert oglesby case study: Object oriented analysis - osbert oglesby case study: software project management plan –Design phase: design and abstraction-action-oriented design data flow analysis-transaction analysis –data oriented design – object-oriented design – elevator problem:object oriented design – formal techniques for detailed design – real-time design techniques – testing during the design phase – CASE tools for the design phase – metrics for the design phase - osbert oglesby case study: object-oriented design – Implementation phase : choice of programming languages – fourth generation languages-good programming practice-coding standards-module reuse-module test case selection-black box module testing techniques-glass box module testing techniques-code walkthroughs and inspections-comparison of module testing techniques-cleanroom-potential problems when testing objects-management aspects of module testing-when to rewrite rather than debug a module-CASE tools for the implementation phase- osbert oglesby case study:black-box test cases.

Unit : V
Implementation and Integration phase: implementation and integration –testing during the implementation and integration phase-integration testing of graphical user interfaces-product testing-acceptance testing-CASE tools for the implementation and integration phase-CASE tools for the complete software process-integrated environments – environments for business applications-public tool infrastructures-potential problems with environments-metrics for the implementation and integration phase- osbert ogles-by case study :implementation and integration phase-Maintenance phase: why maintenance is necessary-what is required of maintenance programmers- maintenance CASE study-management of maintenance – maintenance of object-oriented software- maintenance skills versus development skills-reverse engineering-testing during the maintenance phase-CASE tools for the maintenance phase – metrics for the maintenance phase- osbert oglesby case study: maintenance.

Text Book
1. **Stephen R.Schach**, Object-Oriented Software Engineering, McGraw Hill publications, fourth Edition.

Reference Books
1. Dr. Bernd Bruegge, Allen Dutoit , "Object-Oriented Software Engineering: Conquering Complex and Changing Systems, Prentice-Hall, Inc,
2. Subhash Mehta, Suresh K. Basandra, Object Oriented Software Engineering, Galgotia Publications Pvt. Ltd. 1995

SE209 DESIGN PATTERNS

Credit: 3:1:0
Marks: 40 + 60

Unit : I
Introduction: What is a design pattern? - Design patterns in Smalltalk MVC – describing design patterns – the catalog of design patterns – organizing the catalog – how design patterns solve design problems – how to select a design pattern – how to use a design pattern.

Unit : II
A Case Study: Designing a document editor: design problems – document structure – formatting –embellishing the user interface – supporting multiple look-and-feel standards – supporting multiple window systems – user operations – spelling checking and hyphenation.

Unit : III
Design Pattern Catalog: creational patterns: abstract factory – builder – factory method – prototype – singleton – discussion of creational patterns.

Unit : IV
Structural Patterns: Adapter –bridge – composite –decorator –façade – flyweight-proxy – discussion of structural patterns.

Unit : V
Behavioral Patterns: Chain of responsibility – command – interpreter – iterator – mediator – memento – observer – state – strategy – template method – visitor-discussion of behavioral patterns- Class diagram –object diagram – interaction diagram.

Text Book
1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 'Design Patterns', International Student Edition, 1998.

Reference Books
1. Martin Fowler, Analysis Patterns: Reusable Object Models,Addition Wesley, 1997.
2. Karen Gardner, Alex Rush, Bob Konitzer, Mike Crist and Bobbin Teegarden, Cognitive Patterns: Problem Solving Frameworks for Object Technology, Cambridge University Press (1998).
3. Doug Lea, Concurrent Programming in Java: Design Principles and Patterns, Addison-Wesley, October 1996.

ADDITIONAL SUBJECTS

Code	Subject Name	Credit
SE210	Linux Administration	4:0:0
SE211	Web Technology	4:0:0
SE212	Linux Administration Lab	0:0:2
SE213	Web Technology Lab	0:0:2
SE214	PHP Programming	4:0:0
SE215	Software Reuse	4:0:0
SE216	Multimedia Lab	0:0:2

SE210 LINUX ADMINISTRATION

Credit 4:0:0 **Marks (40 + 60)**

Unit: I

Introduction: Linux's relationship to UNIX – Linux and UNIX history – Linux distributors – Notation and typographical conventions – Where to go for information – How to find and install Software - Essential tasks of the system administrator – System administration under duress.

Booting and Shutting Down: Bootstrapping - Booting PCs – Boot loaders: LILO and GRUB – Booting Single user mode – Startup scripts – Rebooting and shutting down.

Rootly Powers: Ownership of files and processes – The Super user - Choosing a root password - becoming root – other pseudo-users. (Chapters: 1,2,3)

Unit: II

Controlling Processes: Components of a process – The life cycle of a process – signals – process states – Runaway processes.

The FileSystem: Pathnames – Mounting and unmounting file systems – the organization of the file tree – File Types – File attributes.

Adding New Users: The /etc/passwd file - The /etc/shadow file – The /etc/group file – Adding users – Removing Users. (Chapters: 4,5,6)

Unit: III

Serial Devices: Serial standards – Alternative connectors – Hard and soft carrier – Hardware flow control – Cable length – Serial device files – setserial – Software configuration for serial devices – Configuration of hardwired terminals – Special characters and the terminal driver – How to unwedge a terminal – Modems – Debugging a serial line - other common I/O ports.

Adding a Disk: Disk interfaces - Disk geometry – An overview of the disk installation procedure – The ext2 and ext3 file systems – fsck:check and repair file systems – Adding a disk to Linux.

Periodic Processes:cron:schedule commands – The format of crontab files – Crontab management – Some common users for cron. (Chapters:7,8,9)

Unit: IV

Backups: Motherhood and apple pie – Backup devices and media – Setting up an incremental backu regime with dump – Restoring from dumps with restore – Dumping and restoring for upgrades – Using other archiving programs – Using multiple files on a single tape - Amanda – Commercial backup products

Syslog and log files: Logging policies – Linux log files – logrotate – Syslog – Condensing log files to useful information

Drivers and the Kernel: Kernel adaption – Why configure the kernel ? – Configuration methods - Tuning a Linux Kernel – Adding a device drivers – Adding a Linux device driver – Device files – Loadable kernel modules – Building a Linux kernel (Chapters:10,11,12)

Unit:V

The Network File System: General information about NFS – Server side NFS – Client –side NFS – Automating mounting – automount - amd .

Sharing System Files-Copying Files around NIS, NIS+, LDAP.

Network Management and Debugging –Trouble Shooting a Network –Ping ,TraceRoute,Net Stat,Packets Sniffers,Network Management Protocol,SNMP,The NET-SMNP,Network Management Applications.

Coperating with Windows: File and Print Sharing - Secure terminal emultion with SSH – X windows emulators – PC mail clients - PC backups – Dual booting – Running Windowa application under Linux – PC hardware. (Chapters: 17,18,20,27)

Text Book

1. Evi Nemeth, Garth Snyder and Trent R.Hein, *Linux Administration Handbook*, Prentice – Hall of India ,2002.

Reference Book

1. Christopher Negus, *Red Hat Linux 9 Bible*, Wiley Publishing, 2003.

SE211 WEB TECHNOLOGY

Credits 4:0:0 **Marks (40 + 60)**

Unit: I

VBScript Language Elements: Constants - Variables and Data Types - Mathematical Operations - Logical Operators - Looping and Decision Structures.

VBScript Functions and Objects: Data Conversion Functions - Mathematical Functions - Data Formatting Functions - Text Manipulation Functions - Data and Time Functions - Built-in Objects. (Chapters: 2,3)

Unit: II

ASP fundamentals: Using Server – Side Includes- Learning the SSI Directives – Creating Modular ASP Code.

Using the Request Object: Using Form Information - Using Query String Information – Using Server Variables.

Using the Response Object: Create Output – Managing Output – Managing the Connection. (Chapters: 4,5,6)

Unit: III

Using Cookies: Introduction to Cookies – Cookies and Your Browser – Creating a Cookie – Modifying and removing Cookies – Tracking Preferences with Cookies

Using the Application, Session, and Server Objects: The application Object - The Session Object – The Server Object – Using the global .asa file

Active Data Objects Essentials: Microsoft's Universal Data Access Strategy – The Connection Object – The Record set and Field Objects – The Command and Parameter Objects – Using the Errors Collection. (Chapters: 7,8,10)

Unit: IV

Introducing XML: What is XML - Why are Developers Excited About XML? – The Life of an XML documents - Related technologies

XML applications: XML Applications–XML for XML–Behind the Scene Uses of XML

Your First XML Document: Hello XML – Exploring the Simple XML Document – Assigning Meaning to XML Tags – Writing a Style Sheet for an XML Document – Attaching a Style Sheet to an XML Document.

Structuring Data: Examining the Data – XML izing the Data – The advantages of the XML Format - Preparing a Style Sheet for Document Display. (Chapters: 1,2,3,4)

Unit: V

Attributes, Empty tags & XSL: Attributes – Attributes versus Elements – Empty Elements and Empty Element Tags – XSL

DTDs and Validity: Document Type Definitions - Element Declarations – DTD Files – Document Type Declarations – Validating Against a DTD.

Element Declaration - Entity Declarations: What Is an Entity – Internal General Entities – External General Entities – Internal Parameter Entities – External Parameter Entities – Building a Document from Places.

Attribute Declaration: What is an Attribute – Declaring Attributes in DTDs - Declaring Multiple Attributes – Specifying Default Values for Attributes – Attribute Types – Predefined Attributes – A DTD for Attribute- Based Baseball Statistics.
(Chapters: 5,8,9,10,11)

Text Books

1. Eric A. Smith, *ASP 3 Programming Bible*,Wiley-dreamlech India (P) Ltd ,2002.
2. Elliotte Rusty Harold, *XML Bible*, IDG Books India(P) Ltd, 2001.

Reference Books

1. Dave Mercer,*ASP 3.0 Beginners Guide* , Tata MCGRAw-Hill Edition,fifth reprint,2001.
2. Kenneth L. Spencer, Kenneth C. Miller & Lauran Lassesen , *Introducing VBSCript & ActiveX*, Comdex Computer Publication, 1997

SE212 LINUX ADMINISTRATION LAB

Credit 0:0:2 **Marks (50 +50)**

12 experiments will be notified by the HOD from time to time

SE213 WEB TECHNOLOGY LAB

Credit 0:0:2 **Marks (50 + 50)**

12 experiments will be notified by the HOD from time to time

SE214 PHP PROGRAMMING

Credit (4:0:0) **Marks (40 + 60)**

Unit: I

PHP Introduction: PHP Basics - Evolution of PHP - Introduction to PHP Programming
Variables, Operators, and Constants: Introduction to Variables – Operators - Type Juggling - Type Casting - Variable Variables - function for Determining and Setting Variable- Types - Constants.
Control Structures: Conditional Expressions- Arrays: Introduction to Arrays- Initializing Arrays - Working with Arrays.
Functions: Introduction to Functions - Passing Arguments to Functions - Returning Values from Functions - Understanding the Scope of a Variable within a Function, Variable Functions and Variable Argument Functions (Chapters:1,3,4,5,6)

Unit: II

Understanding Classes: Classes – Constructors - Extending a class.
Form Parsing in PHP: Parsing HTML Posted Values in PHP - Form Validation- Printing the Confirmation Page - Printing Hidden Fields - Putting Theory into Practice.
Handling Files: Working with Files - Putting Theory into Practice.(Chapters:7,10,11)

Unit: III

Handling Data Storage: An Introduction to Database Concepts - Database Management System - Relational Database Management System - Database Normalization - PHP Support to various Databases - Web Database Architecture - MySQL Database Programming.
Using PHP with SQL Databases: Working with MySQL - Using Multiple Databases Simultaneously - Important PHP-MySQL Functions - Error handling in MySQL - Error Types in PHP - Creating Customized Error Handlers.(Chapters:12,13)

Unit: IV

Session Tracking: An Overview of Sessions - Tracking Sessions.
Input Validators in PHP: Validation Basics - Performing Validations in an HTML Web Page - Performing Validations in PHP - Functions for Validating User Input - Validating E-Mail Addresses.**Cookies:** What Is a Cookie? Implementing Cookies in PHP - Are Cookies Harmful?(Chapters:18,20,24)

Unit: V

Graphics in PHP: Graphics on the web - Creating Images in PHP.

Understanding CVS: CVS an Overview – The CVS Repository – Environment Variables in CVS - CVS Command Options .
PEAR: Introduction to PEAR – Coding Standards in PEAR – PEAR and CVS – Contributing Codes to PEAR – Requirements to Make Changes in PEAR – Help in PEAR.(Chapters:19,21,22)

Text Book

1. Ashish Wilfred,Meeta Gupta and Kartik Bhatnagar with NIIT,*PHP Professional Projects*,Prentice hall of India,2002.

Reference

1. Rasmus Lerdorf and Kevin Tatroe, "Programming in PHP", O'Reilly and Associates, 2002

SE215 SOFTWARE REUSE

Credit 4:0:0 **Marks (40+60)**

Unit: I
Software Reuse Success Factors: Software reuse is a simple idea – components a systematic approach- reuse requires changes in process- reuse requires changes in organization- adopt reuse systematically and incrementally- Input from other reuse programs-reuse Business has business characteristics- architect components and applications- Software engineering processes establishing and managing a reuse business.
Object-oriented Software Engineering: Software engineering transforms requirements into code- Software engineering is a team process- systematic model building- objects unify the modeling process- use case model- analysis model - design model-Implementation model-test model. (Chapters : 1,3)

Unit: II
Application and Component Systems: Application developers- Application systems-group components into component systems- Facades control access- facades and component systems- component systems- specialize some components- variability occurs at variation points- variability mechanism- Reuse variable components- package and document component systems.
Use Case Components: Structure the use case model- use case model- reusing components-design- Not all use cases should be reusable components-reusing concrete- expressing use case variability, packaging and documenting.
Object Components: Object models- reusing analysis and design components,-expressing variability- tracing use case variability- reusable analysis- design and implementation-packaging and documenting. (Chapters:4,5,6)

Unit: III
Layered Architecture: Architecture structure- good architecture- layered architecture-reduces software dependencies- middleware layer- business-specific layer- using multiple model- representing a layered system as super ordinate system - use cases in relation to a layered system-actors of the application and component systems use cases- wrapping legacy systems. **Object-Oriented Business Engineering:** Business process- well-defined process

for business process- business engineering delivers models- using business actors- using workers and work objects-grouping workers into competence units- Applying business engineering to define processes and organization. Processes and organization of the reuse Business- Software engineering processes- interplay between reuse business processes. (Chapters:7,8,9)

Unit: IV

Application Family Engineering: Developing an architecture- planning, AFE1, 2,3,4,5, managing architectural change- expressing application family engineering in terms of workers-leaner approach to application family - Component system engineering. Building flexible component systems, CSE1, 2,3,4,5,6, expressing component system engineering in terms of workers.

Application System Engineering: Building application systems, ASE1, ASE2, 3,4,5,6, expressing application system engineering in terms of workers

(Chapters 10, 11, 12)

Unit: V

Transition to a Reuse Business: A systematic- incremental approach is needed-incremental transition process- TRA1, TRA2, TRA3, TRA4, TRA5.

Managing the reuse business: Ongoing management is crucial, measurement is key- TRA6: Managing people and organization

Afterward: Making the reuse business work: Putting it all together - reuse improves the performance of your business processes - common misconceptions doing reuse is difficult - Without vision, the people perish - Reuse depends on architecture - management works through organization - reuse business must earn a return on its investment - Software engineering depends on process- object technology aids process-business engineering.

(Chapters : 13,14,15)

Text Book

1. Ivar Jacobson, Martin Griss and Patrik Jonsson, "*Software Reuse*"**,** Addison Wesley, 1997.

Reference

1. Jog Sodhi and Prince Sodhi, 'Software Reuse: Domain Analysis and Design Processes", McGraw Hill, 2000

SE216 MULTIMEDIA LAB

Credit 0:0:2 **Marks 50 + 50**

12 experiments will be notified by HOD from time to time