

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY**  
**M.TECHNOLOGY (COMPUTER SCIENCE & ENGINEERING)**  
**COURSE STRUCTURE AND SYLLABUS**

**I YEAR I SEMESTER**

<b>Code</b>	<b>Group</b>	<b>Subject</b>	<b>L</b>	<b>P</b>	<b>Credit</b>
		Advanced Problem Solving	3	0	3
		Computer System Design	3	0	3
		Embedded Systems	3	0	3
		Java and Web Technologies	3	0	3
	Elective -I	Object Oriented Modeling Software Quality Assurance and Testing Software Architecture and Design Patterns	3	0	3
	Elective -II	Software Design and Engineering Advanced Compiler Design Image Processing and Pattern Recognition	3	0	3
	Lab	Java and Web Technologies Lab	0	3	2
		Seminar	-	-	2
		Total Credits (6 Theory + 1 Lab.)			22

**ADVANCED PROBLEM SOLVING**

**Unit I**

OOP Using Java - Class and Objects, Variables, Operators, Expressions, Methods, Decision statements, Loops, Arrays, OOP concepts- Encapsulation, Inheritance, Polymorphism, Abstraction, Modularity, Exception handling, Input and Output, Java and Pointers, Interfaces, Packages, Abstract classes, Casting in Inheritance hierarchy, Casting with Interfaces, Vectors in java.util, Data Structures and OOP, Writing a java program- Design, coding, testing and debugging.

Basic concepts (Review)- Abstract Data Types, Data structures, Algorithms- Characteristics of Algorithms, Performance analysis- Time complexity and Space complexity, Asymptotic Analysis- Big O, Omega and Theta notations.

**Unit II**

Linear data structures- The List ADT, Array and Linked Implementations, Singly Linked Lists- Operations- Insertion, Deletion, Traversals, Doubly Linked Lists- Operations- Insertion, Deletion, Skip Lists- implementation, Stack ADT, definitions, operations, Array and Linked implementations, applications- infix to postfix conversion, recursion implementation, tail recursion, nontail recursion, indirect recursion, Queue ADT, definitions and operations, Array and Linked Implementations, Priority Queue ADT, Deque ADT, Implementation using doubly linked lists, Stacks and Queues in java.util.

**Unit III**

Non Linear data structures- Trees- Basic Terminology, Binary tree ADT, array and linked representations, iterative traversals, threaded binary trees, Applications- Disjoint-Sets, Union and Find algorithms, Huffman coding, General tree to binary tree conversion, Realizing a Priority Queue using Heap.

Search Trees- Binary Search Tree ADT, Implementation, Operations- Searching, Insertion and Deletion, Balanced Search trees- AVL Trees, Operations – Insertion and Searching, B-Trees, B-Tree of order m, Operations- Insertion, Deletion and Searching, Introduction to Red-Black Trees, Splay Trees, B\*-Trees, B+-Trees (Elementary treatment), Comparison of Search Trees, Trees in java.util.

**Unit IV**

Searching- Linear Search, Binary Search, Hashing- Hash functions, Collision-Handling schemes, Hashing in java.util, Dictionary ADT, Linear list representation, Skip list representation, Hash table representation, Comparison of Searching methods.

Sorting- Bubble Sort, Insertion Sort, Shell sort, Heap Sort, Radix Sort, Quick sort, Merge sort, Comparison of Sorting methods, Sorting in java.util.

**Unit V**

Graphs- Basic Terminology, Graph Representations- Adjacency matrix, Adjacency lists, Adjacency multilists, Graph traversals- DFS and BFS, Spanning trees- Minimum cost spanning trees, Kruskal's Algorithm for Minimum cost Spanning trees, Shortest paths- Single Source Shortest Path Problem, All Pairs Shortest Path Problem.

Text Processing - Pattern matching algorithms- The Knuth-Morris-Pratt algorithm, The Boyer-Moore algorithm, Tries- Standard Tries, Compressed Tries, Suffix tries.

**TEXT BOOKS :**

1. Data structures and Algorithms in Java, Adam Drozdek, Cengage Learning.
2. Data structures and Algorithms in Java, Michael T. Goodrich and R. Tomassia, Wiley India edition.
3. Data structures, Algorithms and Applications in Java, S. Sahani, Universities Press.

**REFERENCE BOOKS :**

1. Data structures and algorithms in Java,Robert Lafore,Pearson Education.
2. Data structures with Java,W.H.Ford and W.R.Topp,Pearson Education.
3. Classic Data structures in Java,T.Budd,Pearson Education.
4. Data Structures using Java,D.S. Malik and P.S.Nair, Cengage Learning,
- 5.An Introduction to Data structures and Algorithms,J.A.Storer,Springer.
- 6.Data structures and Java Collections Frame Work,W.J.Collins,Mc Graw Hill.
- 7.Data structures with Java,J.R.Hubbard and A.Huray,PHI.
- 8.Data Structures using Java,Y.Langsam,M.Augenstein,A.Tanenbaum,Pearson Education.
- 9.Data structures with Java,J.R.Hubbard,Schaum's Outlines, TMH.

**COMPUTER SYSTEM DESIGN**

**UNIT I**

**Computer structure** – hardware, software, system software, Von-neumann architecture – case study. IA -32 Pentium: registers and addressing, instructions, assembly language, program flow control, logic and shift/rotate instructions, multiply, divide MMX,SIMD instructions, I/O operations, subroutines.

Input/Output organization, interrupts, DMA, Buses, Interface circuits, I/O interfaces, device drivers in windows, interrupt handlers

**UNIT II**

**Processing Unit:** Execution of a complete instruction, multiple bus organization, hardwired control, micro programmed control.

**Pipelining:** data hazards, instruction hazards, influence on instruction sets, data path & control consideration,RISC architecture introduction.

**UNIT – III**

**Memory:** types and hierarchy, model level organization, cache memory, performance considerations, mapping, virtual memory, swapping, paging, segmentation, replacement policies.

**UNIT – IV**

**Processes and Threads:** processes, threads, inter process communication, classical IPC problems, Deadlocks.

**UNIT – V**

**File system:** Files, directories, Implementation, Unix file system

**Security:** Threats, intruders, accident data loss, basics of cryptography, user authentication.

**TEXT BOOKS:**

1. Computer Organization – Car Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition, McGraw Hill.
2. Modern Operating Systems, Andrew S Tanenbaum 2<sup>nd</sup> edition Pearson/PHI

**REFERENCE BOOKS:**

1. Computer Organization and Architecture – William Stallings Sixth Edition, pearson/PHI
2. Morris Mano -Computer System Architecture –3<sup>rd</sup> Edition-Pearson Education .
3. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7<sup>th</sup> Edition, John Wiley
4. Operating Systems – Internals and Design Principles Stallings, Fifth Edition–2005, Pearson Education/PHI

**EMBEDDED SYSTEMS**

**UNIT I**

**Introduction to Embedded Systems:** Embedded Systems, Processor Embedded into a System, Embedded Hardware Units and Devices in a System, Embedded Software, Complex System Design, Design Process in Embedded System, Formalization of System Design, Classification of Embedded Systems

**UNIT II**

**8051 and Advanced Processor Architecture:** 8051 Architecture, 8051 Micro controller Hardware, Input/Output Ports and Circuits, External Memory, Counter and Timers, Serial data Input/Output, Interrupts, Introduction to Advanced Architectures, Real World Interfacing, Processor and Memory organization - **Devices and Communication Buses for Devices Network:** Serial and parallel Devices & ports, Wireless Devices, Timer and Counting Devices, Watchdog Timer, Real Time Clock, Networked Embedded Systems, Internet Enabled Systems, Wireless and Mobile System protocols

**UNIT III**

**Embedded Programming Concepts:** Software programming in Assembly language and High Level Language, Data types, Structures, Modifiers, Loops and Pointers, Macros and Functions, object oriented Programming, Embedded Programming in C++ & JAVA

**UNIT IV**

**Real – Time Operating Systems:** OS Services, Process and Memory Management, Real – Time Operating Systems, Basic Design Using an RTOS, Task Scheduling Models, Interrupt Latency, Response of Task as Performance Metrics - **RTOS Programming:** Basic functions and Types of RTOSes, RTOS VxWorks, Windows CE

**UNIT V**

**Embedded Software Development Process and Tools:** Introduction to Embedded Software Development Process and Tools, Host and Target Machines, Linking and Locating Software, Getting Embedded Software into the Target System, Issues in Hardware-Software Design and Co-Design - **Testing, Simulation and Debugging Techniques and Tools:** Testing on Host Machine, Simulators, Laboratory Tools

**TEXT BOOKS:**

1. Embedded Systems, Raj Kamal, Second Edition TMH.

**REFERENCE BOOKS :**

1. Embedded/Real-Time Systems, Dr.K.V.K.K.Prasad, dreamTech press
2. The 8051 Microcontroller and Embedded Systems, Muhammad Ali Mazidi, Pearson.
3. The 8051 Microcontroller, Third Edition, Kenneth J.Ayala, Thomson.
4. An Embedded Software Primer, David E. Simon, Pearson Education.
5. Micro Controllers, Ajay V Deshmukhi, TMH.
6. Microcontrollers, Raj kamal, Pearson Education.
7. Introduction to Embedded Systems,Shibu K.V,TMH.

**JAVA AND WEB TECHNOLOGIES**

## Unit I:

HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;  
Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script, CSS

## Unit II:

XML: Document type definition, XML Schemas, Document Object model, Presenting XML,  
Using XML Processors: DOM and SAX Review of Applets, Class, Event Handling, AWT  
Programming.

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

## Unit III:

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

Web servers: Tomcat Server installation & Testing.

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

## Unit IV:

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

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

## Unit V:

JSP Application Development: Generating Dynamic Content, Using Scripting Elements

Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set  
an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data  
Between JSP pages, Requests, and Users Passing Control and Data between Pages – Sharing  
Session and Application Data – Memory Usage Considerations

Database Access Database Programming using JDBC Studying Javax.sql.\* package Accessing  
a Database from a JSP Page Application – Specific Database Actions Deploying JAVA Beans  
in a JSP Page

**TEXT BOOKS:**

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

## REFERENCE BOOKS:

1. Programming world wide web-Sebesta,Pearson
2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES , Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program , Dietel and Nieto PHI/Pearson.
4. Jakarta Struts Cookbook , Bill Siggelkow, S P D O'Reilly for chap 8.
5. Murach's beginning JAVA JDK 5, Murach, SPD
6. An Introduction to web Design and Programming –Wang-Thomson
7. Professional Java Server Programming,S.Allamaraju and othersApress(dreamtech).
8. Java Server Programming ,Ivan Bayross and others,The X Team,SPD
9. Web Warrior Guide to Web Programmming-Bai/Ekedaw-Thomas
10. Beginning Web Programming-Jon Duckett WROX.
11. Java Server Pages, Pekowsky, Pearson.
12. Java Script,D.Flanagan,O'Reilly,SPD.

## OBJECT ORIENTED MODELING ELECTIVE – I

### UNIT I

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

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

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

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

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

### UNIT II

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

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

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

### UNIT III

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

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

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

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

### UNIT IV

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

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

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

### UNIT V

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

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

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

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

### TEXT BOOKS :

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

### REFERENCE BOOKS :

1. Fundamentals of Object Oriented Design in UML By Meilir Page-Jones, Pearson Education
2. Object Oriented Analysis & Design By Atul Kahate, The McGraw-Hill.
3. Practical Object-Oriented Design with UML By Mark Priestley, TATA McGrawHill
4. Object Oriented Analysis & Design By Brett D McLaughlin, Gary Pollice and David West, O'REILY .
5. Object-Oriented Analysis and Design using UML By Simon Bennet, Steve McRobb and Ray Farmer, 2<sup>nd</sup> Edition, TATA McGrawHill.
6. Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, THOMSON Course Technology.
7. UML and C++, R.C.Lee, and W.M.Tepfenhart, PHI.



**SOFTWARE QUALITY ASSURANCE AND TESTING  
ELECTIVE – I****UNIT I**

**Software Quality Assurance Framework and Standards SQA Framework:** What is Quality? Software Quality Assurance, Components of Software Quality Assurance – **Software Quality Assurance Plan:** Steps to develop and implement a Software Quality Assurance Plan – **Quality Standards:** ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma

**UNIT II**

**Software Quality Assurance Metrics and Measurement Software Quality Metrics:** Product Quality metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metric Programs – **Software Quality metrics methodology:** Establish quality requirements, Identify Software quality metrics, Implement the software quality metrics, analyze software metrics results, validate the software quality metrics – **Software quality indicators – Fundamentals in Measurement theory**

**UNIT III**

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

**Software Testing Methodology**

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

**UNIT IV****Software Testing Techniques**

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

**Software Testing Tools**

Taxonomy of Testing tools, Methodology to evaluate automated testing tools, Load Runner, Win runner and Rational Testing Tools, Silk test, Java Testing Tools, JMetra, JUNIT and Cactus.

**UNIT V****Testing Process**

**Eleven Step Testing Process:** Assess Project Management Development Estimate and Status, Develop Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report test results, testing software installation, Test software changes, Evaluate Test Effectiveness.

**Testing Specialized Systems and Applications**

Testing Client/Server – Web applications, Testing off the Shelf Components, Testing Security, Testing a Data Warehouse

**TEXT BOOKS:**

1. Effective Methods for Software Testing, 2nd Edition, William E. Perry , Second Edition, Wiley India, 2006.
2. Software Quality, Mordechai Ben-Menachem/Garry S. Marliss, Thomson Learning publication, 1997.

**REFERENCE BOOKS:**

1. Testing and Quality Assurance for Component-based Software, by Gao, Tsao and Wu, Artech House Publishers
2. Software Testing Techniques, by Bories Beizer, Second Edition, Dreamtech Press
3. Managing the Testing Process, by Rex Black, Wiley
4. Handbook of Software Quality Assurance, by G. Gordon Schulmeyer, James I. McManus, Second Edition, International Thomson Computer Press
5. Software Testing and continuous Quality Improvement, by William E. Lewis, Gunasekaran Veerapillai, Second Edition, Auerbach Publications
6. Metrics and Models for Software Quality Engineering, by Stephen H. Kan, by Pearson Education Publication
7. Software Testing Tools, K.V.K.K. Prasad, Dream tech press, 2008.
8. Practical Software Testing, Ilene Burnstein, Springer, 2003.
9. Software Testing, Srinivasan Desikan & Gopaldaswamy Ramesh, Pearson Education, 2006.
10. Software testing techniques, Scott Loveland & Geoffrey Miller, Shroff Publishers, 2005.
11. Software Quality, Martin Wieczorek & Dirk Meyerhoff, Springer, 2001.

**SOFTWARE ARCHITECTURE AND DESIGN PATTERNS  
ELECTIVE-I**

**UNIT I****Envisioning Architecture**

The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views.

**Creating an Architecture**

Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

**UNIT II****Analyzing Architectures**

Architecture Evaluation, Architecture design decision making, ATAM, CBAM.

**UNIT III****Moving from one system to many**

Software Product Lines, Building systems from off the shelf components, Software architecture in future.

**UNIT IV****Patterns**

Pattern Description, Organizing catalogs, role in solving design problems ,Selection and usage.

**Creational and Structural patterns**

Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight, Proxy.

**UNIT V****Behavioral patterns**

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

**Case Studies**

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

**TEXT BOOKS:**

1. Software Architecture in Practice, second edition, Len Bass,Paul Clements&Rick Kazman, Pearson Education,2003.
2. Design Patterns, Erich Gamma, Pearson Education,1995.

**REFERENCE BOOKS:**

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

**SOFTWARE DESIGN AND ENGINEERING**  
ELECTIVE –II**UNIT I**

**Introduction to Software Engineering:** The evolving role of software, Changing Nature of Software, legacy software, Software myths. **A Generic view of process:** Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models. **Process models:** The waterfall model, Incremental process models, Evolutionary process models, Specialized process models, The Unified process.

**Software Requirements:** Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. **Requirements engineering process:** Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

**UNIT II****Role of Software Design**

The nature of the design process, transferring design knowledge, constraints upon the design process and product, recording design decisions, designing with others, context for design, economic factors, assessing design qualities, quality attributes of the design product, assessing the design process.

**Transferring Design Knowledge-**Representing abstract ideas, design viewpoints, the architecture concept, design methods, design patterns, Design representations, rationale for design methods.

**Design Processes and Strategies:** The role of strategy in design methods, describing the design process – The D – Matrix, design by top-down decomposition, design by composition, organizational influences upon design.

**UNIT III****Designing with objects and components**

**Designing with objects:** design practices for object-oriented paradigm, Object- oriented frame works, Hierarchical object oriented design process and heuristics, the fusion method, the unified process.

**Component – based design:** The component concept, designing with components, designing components, COTS. **Performing User interface design-**The Golden rules, Interface analysis and design models, user and task analysis, analysis of display content and work environment, applying interface design steps, user interface design issues, design evaluation.

**UNIT IV****Project Management and Metrics**

**Project Management :** The management spectrum: people, product, process and project, W5HH principle, critical practices.

**Metrics for Process and Projects:** Process metrics, project metrics, size-oriented metrics, function-oriented metrics, Object-oriented and use-case metrics, metrics for software quality, integrating metrics with in the software process.

**UNIT V****Project Scheduling and Risk Management**

**Project Scheduling:** Basic concepts, project scheduling, defining a task set and task network, timeline charts, tracking the schedule, tracking the progress for an OO project, Earned value analysis.

**Risk Management:** Reactive Vs. Proactive risk strategies, software risks, risk identification, risk projection, risk refinement, risk mitigation and monitoring, the RMMM plan.

**TEXT BOOKS :**

1. Software design, David Budgen, second edition, Pearson education, 2003.
2. Software Engineering :A practitioner's Approach, Roger S Pressman, seventh edition. McGrawHill International Edition, 2009.

**REFERENCE BOOKS :**

1. Applying domain – driven design and patterns, Jimmy Nilsson, Pearson education,2006
2. Software Engineering, Ian Sommerville, seventh edition, Pearson education,2004.
3. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition,Tata Mc-Graw Hill,2006
4. The art of Project management, Scott Berkun, O'Reilly, 2005.
5. Software Engineering Project Management, Richard H. Thayer & Edward Yourdon, second edition, Wiley India, 2004.
6. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008
7. Applied Software Project Management, Andrew Stellman & Jennifer Greene,O'Reilly, 2006.
8. Software Design,Eric Braude,John Wiley&Sons.

**ADVANCED COMPILER DESIGN**  
**ELECTIVE-II**

**UNIT I : Overview of Compilation:** Phases of Compilation – Lexical Analysis, Regular Grammar and regular expression for common programming language features, pass and Phases of translation, interpretation, bootstrapping, data structures in compilation – LEX lexical analyzer generator.

**UNIT II : Parsing:** Context free grammars, Top down parsing – Backtracking, LL (1), recursive descent parsing, Predictive parsing, Preprocessing steps required for predictive parsing.

**Bottom up parsing:** - Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing , handling ambiguous grammar, YACC – automatic parser generator.

**UNIT III : Semantic analysis:** Intermediate forms of source Programs – abstract syntax tree, Attributed grammars, Syntax directed translation, Conversion of popular Programming languages language Constructs into Intermediate code forms, Type checker.

**Symbol Tables:** Symbol table format, organization for block structured languages, hashing, tree structures representation of scope information. Block structures and non block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records.

**UNIT IV : Code Generation-** Processing the intermediate Code- Interpretation, Code generation, Simple code generation, code generation for basic blocks, BURS Code generation and dynamic programming, Register allocation by graph coloring, Evaluation of code generation techniques Preprocessing the intermediate code, post processing the target code, machine code generation.

**Code optimization:** Consideration for Optimization, Machine dependent and machine independent code optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding, DAG representation.

**UNIT V : Data flow analysis:** Dataflow Analysis, Intermediate representation for flow analysis , Various dataflow analyses , Transformations using dataflow analysis Speeding up dataflow analysis , Alias analysis.

**Loop Optimizations** – Dominators, Loop-invariant computations, Induction variables, Array bounds checks, Loop unrolling

**TEXT BOOKS :**

1. Principles of compiler design -A.V. Aho . J.D.Ullman; Pearson Education
2. Modern Compiler Design- Dick Grune, Henry E. Bal, Cariel T. H. Jacobs, Wiley dreamtech.

**REFERENCE BOOKS :**

1. Advanced Compiler Design Implementation, S.S. Muchnick, Elsevier.
2. Compilers principles , techniques and tools A.V. Aho, Ravi Sethi & J.D. Ullman; Pearson ed.,
3. lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
4. Modern Compiler Implementation in C- Andrew N. Appel, Cambridge University Press.
5. Engineering a Compiler-Cooper & Linda, Elsevier.
6. Compiler Construction, Loudon, Thomson..

**IMAGE PROCESSING AND PATTERN RECOGNITION  
ELECTIVE –II**

**UNIT – I**

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

**UNIT – II**

Statistical and spatial operations, Intensity functions transformations, histogram processing, smoothing & sharpening – spatial filters Frequency domain filters, homomorphic filtering, image filtering & restoration. Inverse and weiner filtering, FIR weiner filter, Filtering using image transforms, smoothing splines and interpolation.

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

**UNIT- III**

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

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

**UNIT –IV**

Representation and Description

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

**UNIT V**

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

**Pattern classification:**

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

**TEXT BOOKS :**

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

**REFERENCE BOOKS :**

1. Image Processing, Analysis and Machine Vision, Second Edition, Milan Sonka, Vaclav Hlavac and Roger Boyle. Thomson learning.
2. Digital Image Processing – Williamk. Pratl –John wiley edition
3. Fundamentals of digital image processing – by A.K. Jain. PH
4. Pattern classification, Richard Duda, Hart and David strok John Weily publishers.
5. Digital Image Processing, S.Jayaraman, S.Esakkirajan, T.Veerakumar, TMH.
6. Pattern Recognition, R.Shinghal, Oxford University Press.