|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHEME OF EXAMINATION FOR MASTER OF SCIENCE (COMPUTER SCIENCE (SOFTWARE))**  **w. e. f. Academic Session 2016-17(CHOICE BASED CREDIT SYSTEM ( CBCS))** | | | | | | | | | |
| **Paper Code** | **Nomenclature of Paper** | **Credits** | **Exam Time**  **(hrs.)** | **External Marks** | | **Internal Marks** | | **Total Marks** |
| **Max** | **Pass** | **Max** | **Pass** |
| **FIRST SEMESTER** | | | | | | | | | |
| MS-16-11 | WEB ENGINEERING | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-12 | DATA STRUCTURES AND ALGORITHMS | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-13 | SOFTWARE ENGINEERING | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-14 | DISCRETE MATHEMATICAL STRUCTURES | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-15 | S/W LAB – I BASED ON MS-16-11 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-16 | S/W LAB – II BASED ON MS-16-12 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-17 | SEMINAR | 1 | 1/2 |  |  | 50 | 20 | 50 |
|  | **TOTAL** | **22** |  | **450** |  | **200** |  | **650** |
| **SECOND SEMESTER** | | | | | | | | | |
| MS-16-21 | JAVA PROGRAMMING | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-22 | LINUX AND SHELL PROGRAMMING | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-23 | THEORY OF COMPUTATION | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-24 | COMPILER DESIGN | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-25 | S/W LAB – III BASED ON MS-16-21 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-26 | S/W LAB – IV BASED ON MS-16-22 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MS-16-27 | SEMINAR | 1 | 1/2 |  |  | 50 | 20 | 50 |
|  | **TOTAL** | **22** |  | **450** |  | **200** |  | **650** |

**Note 1:** Every student has to earn 2 credits by selecting an open elective paper from other department(s) of

Faculty of Sciences (Physical Sciences) of KUK during second semester (OE-201 to OE-209).

**Note 2:** Total Credits for the First Year will be 46 (22+22+2).

**Note 3:** Sessional Marks in each theory paper will be awarded by the concerned teacher on the basis of

marks obtained in one class test (of 15 Marks and 90 minutes’ duration) and evaluation of assignments (of 10 Marks).

**Note 4:** Sessional Marks in each practical paper will be awarded by the concerned teacher on the basis of

marks obtained in one practical exam (of 15 Marks and 90 minutes’ duration) and viva-voce (of 10 Marks).

**Note 5**: Size of Groups for all practicals should not be more than thirty students.

**MS-16-11 WEB ENGINEERING**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide fundamentals concepts of Web Services, JavaScript and lays foundations for the advanced studies in the area of web services.

**Learning Outcomes:**

At the end of this course students should be able to:

* Develop customized web based applications.
* Manage and optimize the web applications.

**UNIT – I**

Introduction to Web Engineering: Categories and Characteristics of Web Applications, Web Applications Vs Conventional Software, Need for an Engineering Approach.

Web Essentials: The Internet, Basic Internet Protocols, WWW, HTTP (Structure of Request and Response Messages), Web Browser and its functions, URL, Web Servers and their features, Defining Virtual Hosts, Secure Servers.

**UNIT – II**

MarkUp Languages: Introduction to HTML, Characteristics, XHTML Syntax and Semantics, Fundamental HTML Elements, Lists, Tables, Frames, Forms, XHTML Abstract Syntax, Creating HTML Pages.

Cascading Style Sheets: Features, Core Syntax, Types, Style Sheets and HTML, Style Rule Cascading and Inheritance, Text Properties, CSS Box Model, Normal Flow Box Layout, Positioning and other useful Style Properties.

**UNIT – III**

Client–Side Programming (JavaScript): Introduction, obtaining user inputs, memory concepts, Operators, Control Structures, looping constructs, break, continue statements, Programmer defined functions, Scoping rules, Recursion and iteration, Array declaration and allocation, passing arrays to function, Objects: String, Date, Boolean, Window, document; using cookies, Handling Events Using JavaScript, data validation using regular expressions.

**UNIT – IV**

Server-Side Programming (PHP): PHP, Installing and Configuring MySQL and PHP, Basic Security Guidelines, Variables, Data Types, Operators and Expressions, Constants, Flow Control Functions; Switching Flow, Loops, Code Blocks and Browser Output, Objects, Strings Processing, Form processing, Connecting to database, cookies and session management, dynamic contents.

**Text Books:**

1. Deitel H.M., Deitel P.J., “Internet & World wide Web: How to program”, 4th Ed., Pearson Education.
2. Kogent Learning, “Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book”, Wiley India Pvt. Ltd.

**References Books:**

1. Jeffrey C. Jackson, “Web Technologies”, 1st Ed., Pearson Education, India.

2. Thomas Powell, “The Complete Reference HTML”, 5th Ed., Tata McGraw Hill, India.

3. William Pardi, “XML in Action”, 1st Ed., IT Professional, New York, USA.

**MS-16-12 DATA STRUCTURES AND ALGORITHMS**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide in-depth coverage of advanced data structures and algorithm design techniques. It focuses on learning about analyzing and designing algorithms to solve a problem and learn to find the asymptotic efficiency of an algorithm.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the basic and advanced data structures and to implement them.
* Analyze worst-case running times of algorithms using asymptotic analysis.
* Classify problems into different complexity classes corresponding to both deterministic and randomized algorithms

**UNIT-I**

Introduction to Data Structures: Classification of Data Structures, Arrays, Stacks & Queues: Representation of Stacks, Stack Operations, Applications, Queues, Operations on Queues, Circular Queues, Dequeue, Priority Queues, Applications.

Introduction to Algorithms: Role of algorithms in computing, Complexity of algorithms, analyzing algorithms, designing algorithms, asymptotic notations.

**UNIT-II**

Linked Lists: Introduction, Types, Operations (Insertion, Deletion, Traversal, Searching), Applications, Dynamic Memory Management, Implementation of Linked Representations.

Trees: Definition, Representation of Trees, Types of Tree, Representation of Binary Trees, Binary Tree Traversals, Threaded Binary Trees, Binary Search Trees and Operations, Minimum Spanning Tree, AVL Trees, Heap, m-way Search Trees, B-Trees, B+ Trees, Applications

**UNIT-III**

Divide and Conquer: Complexity of iterative programs and recursive programs, solving recurrence equations: back substitution method, recursion tree method, master’s theorem.

Analysis of heap sort and quick sort; Counting sort, Radix sort, Bucket sort, Lower bounds for sorting.

Dynamic Programming (DP): Elements of DP, Matrix chain multiplication, Longest common subsequence, optimal binary search trees.

**UNIT-IV**

Greedy Techniques (GT): Elements of GT, Activity selection problem, Huffman codes, Knapsack Problem.

Graph Algorithms: Single source shortest path: Analysis of Dijkstra’s Algorithm, Limitations of Dijkstra’s Algorithm, Negative weight cycle, Bellman-Ford algorithm. All Pairs Shortest Path: Relation of Shortest path and matrix multiplication, Analysis of Floyd Warshall algorithm. Maximum Flow: Flow network, Ford-Fulkerson method.

**Text Books:**

1. G.A.V Pai, “Data Structures and Algorithms”, 2nd Ed., McGraw-Hill.
2. Cormen, Leiserson, Rivest, “Introduction to Algorithms”, 3rd Ed., PHI India.

**Reference Books:**

1. Neapolitan R., “Foundations of Algorithms”, 4th Ed., Jones and Bartlett Learning.

2. Trembley, J.P. And Sorenson P.G., “An Introduction to Data Structures with Applications”, 2nd Ed.,

McGraw- Hill.

3. Cooper A., “Computability Theory”, Chapman and Hall/ CRC Press.

4. Robert Sedgewick, “Algorithms in C”, 3rd Ed., Pearson Education India.

5. Seymour Lipschutz, “Data Structures”, Revised 1st Ed., McGraw-Hill, Schaum’s Outlines, New Delhi.

**MS-16-13 SOFTWARE ENGINEERING**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives:**

The objective of this course is to educate the students about (i) the different models of software development, (ii) metrics used in software engineering and (iii) different quality standards.

**Learning Outcomes:**

At the end of this course students should be able to:

* Apply the different tools and techniques to the software development.
* Quantitatively evaluate the process and product.

**UNIT – I**

Introduction: Software Crisis-problem and causes, Software Processes, Software life cycle models: Waterfall, Prototype, Evolutionary and Spiral models, Overview of Quality Standards like ISO 9001, SEI-CMM, CMMI.

Software Metrics: Size Metrics like LOC, Token Count, Function Count, Design Metrics, Data Structure Metrics, Information Flow Metrics, cyclomatic complexity, Halstead Complexity measures.

**UNIT – II**

Software Project Planning: Cost estimation, static, Single and multivariate models, COCOMO model, Putnam Resource Allocation Model, Risk management, project scheduling, personnel planning, team structure, Software configuration management, quality assurance, project monitoring.

Software Requirement Analysis and Specifications: Structured Analysis, Data Flow Diagrams, Data Dictionaries, Entity-Relationship diagrams, Software Requirement and Specifications, Behavioral and non-behavioral requirements.

**UNIT – III**

Software Design: Design fundamentals, problem partitioning and abstraction, design methodology, Cohesion & Coupling, Function Oriented Design and User Interface Design.

Coding: Programming style, structured programming.

Software reliability: Metric and specification, Musa and JM reliability model, fault avoidance and tolerance, exception handling, defensive programming.

**UNIT – IV**

Software Testing: Functional testing: Boundary Value Analysis, Equivalence class testing, Cause effect graphing, Structural testing: Control flow based and data flow based testing, loop testing, mutation testing, load, stress and performance testing, software testing strategies: unit testing, integration testing, System testing, Alpha and Beta testing, debugging.

Static Testing: Formal Technical Reviews, Walk Through, Code Inspection.

Software Maintenance: Types of Maintenance, Maintenance Process, Maintenance characteristics, Reverse Engineering, Software Re-engineering.

**Text Books**

1. Pressman R. S., “Software Engineering – A practitioner’s approach”, 7th Ed., Tata McGraw Hill.

2. Sommerville, “Software Engineering”, 6th Ed., Pearson Education.

**Reference Books:**

1. Pfleeger, “Software Engineering: Theory and Practice”, 4th Ed., Pearson Education.

2. P. Jalote, “An Integrated approach to Software Engineering”, 2nd Ed., Narosa Publications.

3. R. Fairley, “Software Engineering Concepts”, Tata McGraw Hill Ed.

4. James Peter, W. Pedrycz, “Software Engineering”, Wiley India Pvt. Ltd, 2007.

**MS-16-14 Discrete Mathematical Structures**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide in-depth coverage of discrete mathematical structures. It focuses on learning about sets, logics, analysis techniques, and graphs and their use in the field of computer science.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the basic concepts of sets, function and relations.
* Understand logics and counting principles.
* Understand the graphs and their use and implementation in computer science.

**UNIT – I**

Set Theory: Basic Set Theory, Operations on Sets, Algebra of sets, Venn Diagrams.

Relations: Binary Relations, Complement of relations, Inverse of relations, Composite relations, Properties, Equivalence, Partial Order and Total order relations.

Functions: Functions on Set, Domain, Co-domain, Representation of Functions, Types, Identity and Inverse Functions, Composition of Functions.

**UNIT –II**

Propositional Calculus: Propositional logic, Equivalences, Predicates, Quantifiers, Nested Quantifiers, Rules of Inference, Normal Forms, Proofs: Methods, Strategy.

Counting: Pigeonhole Principle, Inclusion-Exclusion Principle, Permutations and Combinations, Binomial Coefficients, Counting Principles.

**UNIT –III**

Advanced Counting Techniques: Recurrence Relations, Solving Recurrence Relations, Divide and Conquer Algorithms and Recurrence Relations, Solution of Recurrence Relations by the method of Generating Function.

Latices and Boolean algebra: Lattices, Hasse Diagram, Principle of Duality, Types of Lattices, Special Lattices, Boolean Expression, Equivalent circuits, Dual, Normal Forms.

**UNIT –IV**

Graphs: Introduction, Terminology, Types of Graphs, Representation of Graphs, Paths and Circuits, Cut-set and Cut - Vertices, Graph Isomorphism, Homomorphism, Connectivity, Bipartite Graphs, Subgraphs, Operations on Graphs, Euler and Hamiltonian Graphs, Shortest Path Problem, Planar & Dual Graphs, Coloring.

Tree: Tree Notations, Properties of tree, Types of Tree, Operations, Minimum Spanning Tree (MST).

**Text Books**:

1. Kenneth G. Rosen, “Discrete Mathematics and Its Applications”, 7th Ed., Tata McGraw Hill.
2. Koshy T., “Discrete Mathematics with Applications”, 1st Ed., Elsevier India.

**Reference Books**:

1. Eric Gosett, “Discrete Mathematics with proof”, 2nd Ed., Wiley India Pvt. Ltd.
2. Seymour Lipshutz, “Schaum Outlines of Discrete Mathematics”, 2nd Ed., Tata McGraw-Hill.
3. Kenneth Ross, “Discrete Mathematics”, 5th Ed., Pearson Educations India.

**MS-16–17 SEMInar**

**Maximum marks: 50 Time: 1/2 hours CREDITS: 1**

**Seminar**

Each student shall individually prepare and submit a seminar report within stipulated time. A panel consisting of two teachers (internal) should evaluate the seminar report and the presentation. Marks should be distributed considering report writing, presentation, technical content, depth of knowledge, brevity and references and their participation in seminar. The time allotted for presentation is 30 minutes.

**MS-16-21 JAVA PROGRAMMING**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide in-depth knowledge of programing with Java. It focusses on the various basic concepts of core java so that the concepts can be utilized to create projects in Java.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the basic features of programming with Java.
* Use the knowledge of Java to produce software for real life problems.

**UNIT – I**

Introduction to Java: Importance and features of Java, Java virtual machine, Byte code, JDK, Keywords, constants, variables and Data Types, Operators and Expressions, Decision Making, Branching and Looping, jump statements: break, continue, return. Introducing classes, objects and methods: defining a class, adding variables and methods, creating objects, constructors, class inheritance. Arrays and String: Creating an array, one and two dimensional arrays, string array and methods.

**UNIT – II**

Packages and interfaces, Exception Handling: Fundamentals exception types, uncaught exceptions, throw exception, built in exception, creating your own exceptions, Multithreaded Programming: Fundamentals, Java thread model: synchronization, messaging, thread classes, Runnable interface, inter thread Communication, suspending, resuming and stopping threads.

**UNIT – III**

I/O Streams: String and String Buffer classes, Wrapper classes: Basics types, using super, Multilevel hierarchy abstract and final classes. Input/output Programming: Basics, Streams, Byte and Character Stream, predefined streams, Reading and writing from console and files.

**UNIT –IV**

Event Handling: Different Mechanism, the Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and Inner Classes, working with windows, Graphics and Text, using AWT controls, Layout managers and menus, handling Image, animation, sound and video, Java Applet.

Beans: Introduction to Java Beans and Swings.

**Text Books:**

1. Patrick Naughton and Herbertz Schildt, “Java-2 The complete Reference”, 10th Ed., McGraw Hill.

**Reference Books:**

1. E Balaguruswamy, **“**Programming with Java”, 4th Ed., Tata McGraw-Hill.

2. Horstmann, “Computing Concepts with Java 2 Essentials”, 2nd Ed., Wiley India Pvt. Ltd.

3. Decker & Hirshfield, “Programming Java”, 3rd ed., Vikas Publication House.

**MS-16-22 LINUX AND SHELL PROGRAMMING**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide in-depth coverage of administering Linux. It focuses on learning about shell programming, c-environment in Linux, and security issues regarding Linux.

**Learning Outcomes:**

At the end of this course students should be able to:

* Write and implement shell programming
* Use C-Environment under Linux
* Handle security issues in Linux.

**UNIT – I**

Introduction: Basic features, architecture, distributions, Installation requirements; Kernel, Shell.

File System: boot block, super block, inode table, data blocks, accessing files, storage of files, standard directories, system calls for files, file and disk related commands, hard disk partitions; System startup and shut down processes, init and run levels, rc and init files.

**UNIT – II**

C language compiler, layout of C program in memory, process environment, kernel support, process images, managing zombie and orphan processes, use of makefiles, dependency calculations, dynamic loader, debugging with gdb.

**UNIT – III**

User Management: Adding new users and groups, super users, creating and mounting file systems. User management commands.

Security and Connections: viewing and changing the permissions and ownerships of files and directories, creating networks, Signal generation and handling, Environment variables, Commands: man, ping, ifconfig, raise, alarm, pause, abort etc.

**UNIT – IV**

Shell: meaning, types; connecting processes with pipes, tee, redirect input and output, background processes, managing multiple processes, changing priority, scheduling of processes, at, batch and cron commands, process related commands, filters.

Shell Programming: Introduction, shell programming in various shells, file name substitution, read command, operators, conditional statements, looping and case statements, expr statement, command line arguments, parameter passing and arguments, associative arrays, string and mathematical functions, arrays and functions, libraries, shell variables, shell programs to automate system tasks, interrupt processing, shell scripts for administrators, debugging shell scripts.

**Text Books:**

1. Matthew Neil, Stones Richard, “Beginning Linux Programming”, 4th Ed., Wiley India Pvt. Ltd.

2. John Goerzen, “Linux Programming Bible”, IDG Books, New Delhi.

**Reference Books:**

1. Negus Christopher, “Linux Bible”, 8th Ed., Wiley India Pvt. Ltd.

2. Petersen Richard, “Linux: The Complete Reference”, 6th Ed., Tata Mcgraw Hill.

3. Venkateshmurthy M.G., “Introduction to Unix & Shell Programming”, Pearson Education.

**MS-16-23 THEORY OF COMPUTATION**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide in-depth coverage of design of machines. It focuses on learning about various languages and their respective machines.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the computable and non-computable problems
* Understand the Grammars for various machines
* Design the machines for various problems.
* Use the reducibility and computability to solve various problems.

**Unit – I**

Computability and Non-computability and examples of non-computable problems, Russel's paradox, Finite State System, Extended Transition Function, Designing of DFA and NDFA, Finite Automata with E-Transitions, Equivalence of DFA and NFA with proof, Regular Expression, Laws of Regular Expressions, Kleene’s Theorem 1 and 2, Properties and Limitations of FSM, FSM with Output: Moore and Mealy Machines, Arden’s Theorem with proof, Closure Properties of Regular Sets, Application of Pumping Lemma, Myhill-Nerode Theorem, Minimization of FA.

**Unit – II**

Grammar: Definition, Chomsky Classification of Grammars, Construction of Context Free Grammar, Derivation, Parse Trees, Ambiguity, Removal of Ambiguity, Simplification of Context Free Grammar, CNF and GNF, Closure properties of CFL, Pumping Lemma for CFL.

Pushdown Automaton: Introduction, Types of PDA, Designing of PDA’s, Conversion from PDA to CFG and vice-versa, Applications, Parsing: Early's, Cook-Kasami-Young, Tomito's, top-down and bottom-up methods

**Unit – III**

Linear Bounded Automata (LBA), Turing machines, variants of TMs, Restricted TMs, TMs and Computers. Recursive and recursively- enumerable languages and Properties.

Decidability: Post's correspondence problem, Rice's theorem, decidability of membership, emptiness and equivalence problems of languages. Random Access Machines, Decidable languages, decidable problems, The halting problem, Diagonalization method, Undecidable problems for Regular expressions, Turing machines and other undecidable problems.

**Unit – IV**

Reducibility: The Set NP and Polynomial Verifiability, Polynomial-Time Reductions and NP-Completeness, The Cook-Levin Theorem, Some Other NP-Complete Problems, Reduction, mapping reducibility.

Computational Complexity: Primitive recursive functions, computable functions, Recursion theorem. Tractable and Intractable problems.

**Text Books**

1. John C. Martin, “Introduction to Languages and Theory of Computation”, 3rd Ed., McGraw Hill.
2. Peter Linz, “An introduction to formal language & automata”, 4th Ed., Jones & Bartlett publications.

**Reference Books:**

1. Hopcroft, J.E., Ullman, J.D., “Formal languages and their relation to Automata”, 3rd Ed., Pearson

Education.

2. Lewis, H.R. & Papadimitrious, C.H., “Elements of the theory of computation”, 2nd Ed., PHI

3. Michael Sipser, “Introduction to the Theory of Computation”, 3rd Ed., Cengage Learning.

**MS-16-24 COMPILER DESIGN**

Maximum marks: 100 External: 75

Time: 3 hours CREDITS: 4 Internal: 25

**Note**: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide in-depth coverage of compiler construction for various languages.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the basic of various translators.
* Design the front-end and back-end of compiler.
* Optimize the compilers.

**Unit – I**

Compilers and Translators, Need of Translators, Tools used for compilation, Structure and Phases of Compiler, Single-Pass and Multi-Pass Compilers, Bootstrapping, Compiler Construction Tools. Bootstrap compilers, Phases of Compilation process.

Lexical Analysis: Design of Lexical Analyzer, Finite Automata and Regular Expressions, Lex package on UNIX systems. Process of Lexical Analysis, Recognition of Regular Expressions.

**Unit – II**

Syntax-Directed Translation: Translation Schemes, Implementation of Syntax-Directed Translators, Intermediate code and its need, Postfix Notation, Parse Trees and Syntax Trees, Three-address code and its representations, Boolean Expressions, Flow of Control.

Symbol Table: Contents of Symbol Table, Data Structures used for symbol table, Representing scope information.

Run Time Storage Administration: Types of Storage Allocation Schemes, Implementation of Stack Allocation Scheme, Implementation of Block Structured Languages, Storage Allocation in Block Structured Languages. Error Detection and Recovery: Errors, Lexical-Phase Errors, Syntactic Phase Errors, Semantic Errors.

**Unit – III**

Parsing Techniques: Top down & Bottom-up parsing, Shift Reduce parsing, Operator Precedence parsing, Predictive Parsers. Left Recursion and its removal, Recursive Descent parser, Automatic Construction of efficient Parsers: LR parsers, the Canonical Collection of LR(0) items, Constructing SLR parsing tables, Constructing Canonical LR parsing tables, Constructing LALR parsing tables, Using Ambiguous Grammars, an Automatic Parser Generator, Implementation of LR parsing tables, Constructing LALR sets of items. YACC package on UNIX systems.

**Unit – IV**

Intermediate Code Generation: Object programs, Issues in the design of a code generator, Intermediate languages, Quadruples, Generating intermediate code for declarative statement, Register Allocation and Assignment statement, Boolean expression, and case statement, peephole optimization.

Code Optimization: Principle sources of Optimization, optimization of basic blocks, Loop Optimizations, DAG Representation of Basic Blocks, Loop Invariant Computation, Reducible Flow Graphs, Global Data Flow Analysis, code improving transformation.

**Text Books**

1. Alfred V Aho, “Principles of Compiler Design, 3rd Ed., Narosa Publishing House.
2. Tremblay J.P. and Sorenson, “The Theory and Practice of Compiler Writing”, 2nd Ed., McGraw Hill.

**Reference Books:**

1. Dhamdhere D.M, System programming and operating system, 2nd Ed., McGraw Hill.

2. Aho, Sethi, & Ullman, Compilers Principles, Techniques and Tools, 3rd Ed., Pearson Education.

3. Fischer, “Crafting a compiler in C”, Pearson Education, 2005.

**MS-16–27 SEMInar**

**Maximum marks: 50 Time: 1/2 hours CREDITS: 1**

**Seminar**

Each student shall individually prepare and submit a seminar report within stipulated time. A panel consisting of two teachers (internal) should evaluate the seminar report and the presentation. Marks should be distributed considering report writing, presentation, technical content, depth of knowledge, brevity and references and their participation in seminar. The time allotted for presentation is 30 minutes.