

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

**SCHEME OF INSTRUCTION & EXAMINATION**

**B.E. IIIrd YEAR  
(COMPUTER SCIENCE & ENGINEERING)**

**SEMESTER - II**

Sl. No.	Syllabus Ref. No.	SUBJECT	Scheme of Instruction		Scheme of Examination		
			Periods per week		Duration In Hours	Maximum Marks	
			L	D/P		Univ. Exam	Sessional s
		<b>THEORY</b>					
1.	CS 351	Web Programming & Services	4	-	3	75	25
2.	CS 352	Compiler Construction	4	-	3	75	25
3.	CS 356	Design and Analysis Of Algorithms	4	-	3	75	25
4.	CS 354	Object Oriented System Development	4	-	3	75	25
5.	CS 355	Computer Networks	4	-	3	75	25
		<b>PRACTICALS</b>					
1.	CS 381	WPS & CN Lab	-	3	3	50	25
2.	CS 382	OOSD Lab	-	3	3	50	25
3.	CS 383	Compiler Construction Lab	-	3	3	50	25
4.	CS 384	Mini Project	-	3	-	-	25
		<b>Total</b>	<b>20</b>	<b>12</b>	<b>24</b>	<b>525</b>	<b>225</b>

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

**CS 351**

**WEB PROGRAMMING & SERVICES**

Instruction	4 Periods per week
Duration of University Examination	3 Hours
University Examination	75 Marks
Sessional	25 Marks

**UNIT-I**

**Web basics and overview:** introduction to Internet, World Wide Web, Web Browsers, URL, MIME, HTTP, Web programmer's toolbox Introduction to XHTML. Basics of Java script.

Introduction to XML, XML Document structure, DTD, namespaces, Schemas, XSLT style sheets, XML Processors

**UNIT-II**

**The J2EE Platform:** Enterprise Architecture styles, J2EE Architecture

– Containers, J2EE Technologies, Deploying J2EE applications Introduction to Web containers

**Servlet Programming:** Overview of Java Servlet API, Servlet implementation, Servlet Configuration, Servlet Exceptions, Servlet Life cycle, Request and Responses.

**Servlet Sessions, Context, and collaboration:** Approaches to Session tracking, Session Tracking with Java Servlet API, Servlet Context, Servlet Collaboration

**UNIT-III**

**Filters for Web applications:** What is a Filter, Sample Filter, Filter API, Deployment Descriptor for Filters, Chat Application with Filters.

**Web Deployment, Authentication, and Packaging:** Web application structure, Mapping requests to applications and servlets, Securing web applications, Deploying configuration.

**JSP Basics and architecture:** Introduction to JSP, Jsp Directives, Scripting Elements, Standard Objects, JSP Design strategies.

**JSP Tag extensions:** Tag extensions, A simple Tag Anatomy of a Tag extension, Writing Tag extensions, Application life cycle events

**UNIT-IV**

**Java Mail:** Mail protocols, Java Mail Overview, Installation and Configuration, Java mail API, working with Mail, Java mail resources

Database Programming with JDBC: Database Drivers

**java.sql package:** JDBC Process, Different types of statements, Retrieving meta information from Database and ResultSet

**javax.sql package:** JDBC Data sources, Connection pooling, Distributed transactions, RowSet objects

**UNIT-V**

**.NET Platform:** Introduction to .NET Framework, Common type systems, Common Language Runtime.

Introduction to C#, Types and Objects, Program structure.

**Introduction to ASP.NET:** The basics, ASP.NET documents, Code behind files

ASP.NET controls- HTML controls, Life cycle, page level events, control events, web controls, creating controls with in code, Response output for controls, validation controls

.NET Remoting, Database Connectivity with ADO.NET

**Suggested Reading:**

1. Subramnyam Allamraju, *Professional java server programming J2EE 1.3 Edition*, Cedit Buest. Apress Publications
2. Robert W Sebesta, *Programming the World Wide Web*, Pearson Education
3. Uttam Kumar Roy, *Web Technologies*, first edition, Oxford University Press, 2010.
4. Joe Duffy, *Professional .NET Framework 2.0*, Wiley India 2007.

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

**CS 352**

**COMPILER CONSTRUCTION**

Instruction	4 Periods per week
Duration of University Examination	3 Hours
University Examination	75 Marks
Sessional	25 Marks

**UNIT-I**

**Introduction** – Programs related to compilers. Translation process. Major data structures. Other issues in compiler structure. Boot strapping and porting.

**Lexical analysis** – The role of Lexical Analyzer. Input Buffering. Specification of Tokens. Recognition of Tokens. The Lexical-Analyzer Generator Lex.

**UNIT-II**

**Syntax Analysis** – Introduction. Top-Down parsing, Brute Forcing, Recursive Descent, Predicative LL(1), Bottom-Up parsing : Introduction to LR Parsing, Powerful LR parsers SLR, CALR, LALR, Using Ambiguous Grammars, Parser Generators - Yacc.

**UNIT-III**

**Syntax Directed Translation** – Syntax Directed Definitions. Evaluation Orders for SDDs. Applications of Syntax Directed Translation.

**Symbol Table Organization** - Structure of Symbol table, Symbol Table organization for Block Structured and non block Structure languages, Data Structures of symbol Table.

**UNIT-IV**

**Intermediate code generation** : Variants of syntax trees. Three-Address Code, Types and Declarations. Translation of Expressions. Type Checking. Control Flow.

**Storage Organization.** Stack Allocation of Space. Access to Non local Data on the Stack. Heap Management. Introduction to Garbage Collection.

#### **UNIT-V**

**Code Generation** – Issues in the Design of a Code Generator. The Target Language. Addresses in the Target Code Basic Blocks and Flow Graphs. Optimization of Basic Blocks. Peephole Optimization. Register Allocation and Assignment. Machine Independent Optimizations – The Principal Sources of Optimizations, Introduction to data flow analysis, Foundation of data flow analysis.

**Error Recovery** : Introduction, Error detecting and Reporting in various Phases, Lexical Errors, Syntax Errors handling, and error Recovery in various Phases.

#### **Suggested Reading:**

1. Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D Ullman – *Compilers: Principles, Techniques & Tools*, Pearson Education 2<sup>nd</sup> Edition 2007.
2. Keith D Cooper & Linda Tarezon, *Engineering a Compiler*, Morgan Kaufman, Second edition.
3. K.Muneeswaran, *Compiler Design*, first edition, Oxford University Press, 2012.
4. Lex & Yacc, John R Levine, Tony Mason, Doug Brown, Shroff Publishers.
5. Kenneth C Loudon, *Compiler Construction: Principles and Practice*, Cengage Learning.
6. Lex & Yacc, John R Levine, Oreilly Publishers.

### **WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

#### **CS 356**

#### **DESIGN AND ANALYSIS OF ALGORITHMS**

Instruction	4 Periods per week
Duration of University Examination	3 Hours
University Examination	75 Marks
Sessional	25 Marks

#### **UNIT-I**

Introduction, Algorithm Specification, Performance analysis, Space Complexity, Time Complexity, Asymptotic Notation(O, Omega, Theta), Practical Complexities, Performance Measurement, Review of elementary data structure- Heap and Heap Sort, Hashing, Set representation. UNION, FIND.

#### **UNIT-II**

**Divide-and Conquer:** The general method, finding maximum minimum. Merge sort quick sort and selection.

**Greedy Method:** Knapsack problem, Optimal Storage on tapes, Job sequencing with deadlines, Optimal merge patterns, Minimum Spanning Trees.

#### **UNIT-III**

**Dynamic Programming And Traversal Technique:** Multistage graph, All Pair Shortest Path, Optimal Binary Search trees, 0/1 Knapsack, Reliability Design, Traveling Salesman Problem, Bi

connected Components and Depth First Search.

#### **UNIT-IV**

**Backtracking and Branch and Bounds:** 8-Queens Problem, Graph Coloring Hamilton cycle, Knapsack Problem, 0/1 Knapsack Problem, Traveling salesperson problem, Lower-Bound Theory.

#### **UNIT-V**

NP-Hard and NP-Completeness: Basic concepts, cook's theorem, NP-hard graph problems and scheduling problem, NP-hard code generation

Logic Programming Concepts, Prolog, Theoretical Foundations, Logic Programming in Perspective

problems, Clique Decision problem, Node covering decision, Scheduling problem, NP hard code generation problem.

#### ***Suggested Reading:***

- 1.Horowitz E. Sahani S: "*Fundamentals of Computer Algorithm*", Galgotia Publications.
- 2.Anany Levitin, "*Introduction to the Design & Analysis, of Algorithms*", Pearson Education, 2000.
3. S. Sridhar, *Design and Analysis of Algorithms*, first edition Oxford University Press, 2014
4. Aho, Hopcroft, Ulman, "*The Design and Analysis of Computer Algorithm*", Pearson Education, 2000.
5. Parag H. Dave, Himanshu B. Dave "*Design and Analysis of Algorithms*" Pearson Education, 2008.

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

### **CS 354**

#### **OBJECT ORIENTED SYSTEM DEVELOPMENT**

Instruction	4 Periods per week
Duration of University Examination	3 Hours
University Examination	75 Marks
Sessional	25 Marks

#### **UNIT-I**

**UML Introduction** : Necessity of a Model, Introducing the UML, Hello World.

**Basic Structural Modeling:** Classes, Relationships, Common Mechanisms, Diagrams, Class Diagrams.

**Advanced Structural Modeling:** Advanced Classes, Advanced Relationships, Interfaces, Types and Roles, Packages, Instances, Object Diagrams, Components.

## **UNIT-II**

**Basic Behavioral Modeling:** Interactions, Use Cases, Use Case Diagrams, Interaction diagrams, Activity diagrams.

**Advanced Behavioral Modeling:** Events and Signals, State Machines, Processes and Threads, Time and space, State Chart Diagrams.

## **UNIT-III**

**Architectural Modeling:** Artifacts, Deployment Collaborations, Patterns and Frame-works, Artifact diagrams, Deployment diagrams, Systems and models.

## **UNIT-IV**

**Unified Software Development Process:** The Unified Process, The Four Ps, A Use-Case-Driven Process, An Architecture-Centric Processes, An Iterative and Incremental Process.

## **UNIT-V**

**Core Workflows:** Requirements Capture, Capturing Requirements as Use Case, Analysis, Design, Implementation, Test.

### ***Suggested Reading:***

1. Grady Booch, James Rumbaugh, Ivor Jacobson, "*The Unified Modeling Language-User Guide*" (Covering UML 2.0), 2<sup>nd</sup> Edition, Pearson Education, India,2007.
2. Ivor Jacobson, Grady Booch, James Rumbaugh: "*The Unified Software Development, Process*" Pearson Education, India, 2008.

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

**CS 355**

**COMPUTER NETWORKS**

Instruction

4 Periods per week

Duration of University Examination	3 Hours
University Examination	75 Marks
Sessional	25 Marks
Credits	4

### **Objectives:**

- To understand the state-of-the-art technology in network protocols, network architecture and networked systems
- To learn the design principles of network infrastructure
- To gain proficiency in network programming

### **Outcomes:**

This course enables the student to develop and demonstrate the following

- Ability to compare different network architectures
- Ability to understand the design principles of networking
- Ability to develop applications using network programming

### **UNIT-I**

Review of ISO OSI Reference Model and TCP/IP Architectures.

Network Layer: Design issues, Services, Internal organization, Comparison of Virtual circuits and Datagram subnets. Routing Algorithms: The Optimality principle, Shortest path routing, Flooding, Flow-based algorithms, Distance vector, Link state, Hierarchical algorithms, Broadcast and Multicast routings. Congestion control algorithms: General principles, Traffic shaping, Congestion control in virtual circuit subnets, Choke packets, Load shedding, Jitter control and Congestion control for multicasting, Quality of Service (QoS)

### **UNIT-II**

Internet working: How networks differ, Concatenated virtual circuits, Connectionless internet working, Tunneling, Internetwork routing, Fragmentation and Firewalls.

The Network Layer of the Internet: The IP protocol, IP addresses, Subnets, Internet control protocols, Gateway routing protocols, Multicasting, CIDR.

### **UNIT-III**

Transport Layer: Service primitives, Addressing, Establishing a connection, Releasing a connection, Flow control, Buffering, Multiplexing and Crash recovery.

Internet Transport Protocols (TCP and UDP): The TCP service model, The TCP protocol, The TCP Segment Header, TCP connection management, Transmission policy: Congestion control, Timer management and UDP, Performance issues.

### **UNIT-IV**

Application Layer:

Domain Name System: DNS name space, Resource records, Name services.

SMTP and MIME, HTTP, SNMP, Telnet, ftp, Multimedia.

### **UNIT-V**

Socket programming: Socket address, Elementary socket system calls, Advanced socket system calls, Reserved ports, Socket options, Asynchronous I/O, Input/Output Multiplexing, Out-of-

Band data, Sockets and Signals, Internet Super Server, DNS.

**Suggested Reading:**

1. Andrew S. Tanenbaum, David J. Wetherall, Computer Networks, 5th Edition, Pearson, 2012
2. Chwan-Hwa (John) Wu, J. David Irwin, Introduction to Computer Networks and Cyber Security, CRC Press, 2013.
3. Bhushan Trivedi, *Computer Networks*, first edition, Oxford University Press, 2011
4. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 5th Edition, Addison-Wesley, 2012
5. W. Richard Stevens, Unix Network Programming, Prentice Hall/Pearson Education, 2009
6. W. Richard Stevens, Andrew M Rudoff, Bill Fenner, Unix Network Programming: Networking APIs: Sockets and XTI (Volume 1) 3rd Edition, PHI

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

**CS 381**

**WEB PROGRAMMING AND NETWORKING LAB**

Instruction	3 Periods per week
Duration of University Examination	3 Hours
University Examination	50 Marks
Sessional	25 Marks

***Web Programming Experiments:***

1. Creation of static web site using XHTML.
2. Demonstration of XML, XSLT.
3. Validation of static web page using Java script.
4. Creation of dynamic content in web application using servlets.
5. Handling Sessions in web applications.
6. Usage of Filters in web applications.
7. Creation of dynamic content in web application using JSP.
8. Creation of dynamic content in web application using ASP.NET
9. Providing data store support for web site using JDBC

***Network Programming Experiments:***

10. Understanding and using the following commands. Ifconfig, netstat, ping, arp, telnet, tftp, ftp.



11. Implementation of concurrent and iterative Echo server using both connection oriented and connectionless Socket System Calls.
12. Implementation of time of the day service as Connection Oriented Concurrent Server using Socket System Calls.
13. Build a concurrent Multithreaded File Transfer Server. Use separate Threads to allow the server to handle multiple clients concurrently.
14. Implementation of Remote Program execution using Socket system calls. Programs to demonstrate the usage of Advanced Socket System calls Like Getsockopt(), Setsockopt(), Select(), Readv(), getpeername(), Getsockname()
15. Implement a Concurrent Chat Server that allows currently logged in users to communicate with one another. Use Socket System calls

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

### **CS 382**

#### **OOSD LAB**

Instruction	3 Periods per week
Duration of University Examination	3 Hours
University Examination	50 Marks
Sessional	25 Marks

*Select one large Information System/Approach and device the following using CASE TOOL.*

1. Data Flow diagram.
2. E-R diagram.
3. Dynamic Model and Using Finite State Automata.
4. Software Requirement Specification Document (SRS)
5. Functional Decomposition and Structure.
6. Data Dictionary.
7. Module Specifications.
8. Test Data Generation.
9. Cost and Resource Estimates.
10. Verification
11. User Manual
12. Study of Software Maintenance Tools (SCCS, Debug Tools).
13. A case study using Case Tool supporting UML.

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

### **CS 383**

#### **COMPILER CONSTRUCTION LAB**

Instruction	3 Periods per week
Duration of University Examination	3 Hours
University Examination	50 Marks
Sessional	25 Marks

1. Scanner programs using C
2. Scanner programs using LEX
3. Finding first set and follow set of productions
4. Top down parsers (Recursive decent parser, LL(1) parser, etc.)
5. Bottom up parsers (LR, SLR etc.)
6. Parser programs using YACC
7. Intermediate code generation
8. Code optimization

**WITH EFFECT FROM THE ACADEMIC YEAR 2016 - 2017**

**CS 384**

**MINI PROJECT**

Instruction	3 Periods per week
Sessional	25 Marks

The students are required to carry out mini projects in any of the areas such as Data Communications, Web Programming & Services, Computer Networks, Compiler Construction, and Object Oriented System Development.

*Students are required to submit a report on the mini project at the end of the semester.*

\* \* \*