

CO's-PO's & PSO's MAPPING

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	3	3	3	3	-	-	-	-	3	1	3	2	2	3	2
2	2	2	3	2	2	-	-	-	1	2	3	3	2	1	2
3	3	3	2	1	1	-	-	-	1	1	1	3	2	3	3
4	1	3	3	3	1	-	-	-	1	1	3	2	3	1	3
5	3	2	1	1	1	-	-	-	2	2	3	1	3	1	2
Avg.	2	3	2	2	1	-	-	-	2	1	3	2	2	2	2

1 - low, 2 - medium, 3 - high, ‘-’ - no correlation

CS3591

COMPUTER NETWORKS

L T P C
3 0 2 4

COURSE OBJECTIVES:

- To understand the concept of layering in networks.
- To know the functions of protocols of each layer of TCP/IP protocol suite.
- To visualize the end-to-end flow of information.
- To learn the functions of network layer and the various routing protocols
- To familiarize the functions and protocols of the Transport layer

UNIT I INTRODUCTION AND APPLICATION LAYER 10

Data Communication - Networks – Network Types – Protocol Layering – TCP/IP Protocol suite – OSI Model – Introduction to Sockets - Application Layer protocols: HTTP – FTP – Email protocols (SMTP - POP3 - IMAP - MIME) – DNS – SNMP

UNIT II TRANSPORT LAYER 9

Introduction - Transport-Layer Protocols: UDP – TCP: Connection Management – Flow control - Congestion Control - Congestion avoidance (DECbit, RED) – SCTP – Quality of Service

UNIT III NETWORK LAYER 7

Switching : Packet Switching - Internet protocol - IPV4 – IP Addressing – Subnetting - IPV6, ARP, RARP, ICMP, DHCP

UNIT IV ROUTING 7

Routing and protocols: Unicast routing - Distance Vector Routing - RIP - Link State Routing – OSPF – Path-vector routing - BGP - Multicast Routing: DVMRP – PIM.

UNIT V DATA LINK AND PHYSICAL LAYERS 12

Data Link Layer – Framing – Flow control – Error control – Data-Link Layer Protocols – HDLC – PPP - Media Access Control – Ethernet Basics – CSMA/CD – Virtual LAN – Wireless LAN (802.11) - Physical Layer: Data and Signals - Performance – Transmission media- Switching – Circuit Switching.

45 PERIODS

PRACTICAL EXERCISES:**30 PERIODS**

1. Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and trace route PDUs using a network protocol analyzer and examine.
2. Write a HTTP web client program to download a web page using TCP sockets.
3. Applications using TCP sockets like: a) Echo client and echo server b) Chat
4. Simulation of DNS using UDP sockets.
5. Use a tool like Wireshark to capture packets and examine the packets
6. Write a code simulating ARP /RARP protocols.
7. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.
8. Study of TCP/UDP performance using Simulation tool.
9. Simulation of Distance Vector/ Link State Routing algorithm.
10. Simulation of an error correction code (like CRC)

COURSE OUTCOMES:**At the end of this course, the students will be able to:****CO 1:** Explain the basic layers and its functions in computer networks.**CO 2:** Understand the basics of how data flows from one node to another.**CO 3:** Analyze routing algorithms.**CO 4:** Describe protocols for various functions in the network.**CO 5:** Analyze the working of various application layer protocols.**TOTAL:75 PERIODS****TEXT BOOKS**

1. James F. Kurose, Keith W. Ross, Computer Networking, A Top-Down Approach Featuring the Internet, Eighth Edition, Pearson Education, 2021.
2. Behrouz A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, Sixth Edition TMH, 2022

REFERENCES

1. Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, Fifth Edition, Morgan Kaufmann Publishers Inc., 2012.
2. William Stallings, Data and Computer Communications, Tenth Edition, Pearson Education, 2013.
3. Nader F. Mir, Computer and Communication Networks, Second Edition, Prentice Hall, 2014.
4. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", McGraw Hill, 2012.

CO's-PO's & PSO's MAPPING

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	-	2	-	-		-	-	-	-	-	-	-	3	-	-
2	-	1	-	-	2	-	-	-	-	-	-	2	-	2	-
3	-	2	-	-	3	-	-	-	-	-	-	-	-	3	-
4	-		-	1	2	-	-	-	-	3	-	-	-	-	-
5	-	3	2	-	-	-	-	-	-	-	-	-	-	-	3
AVg.	-	1	-	-	1	-	-	-	-	1	-	-	-	1	1

1 - low, 2 - medium, 3 - high, ‘-’ - no correlation

COURSE OBJECTIVES:

- To learn the various phases of compiler.
- To learn the various parsing techniques.
- To understand intermediate code generation and run-time environment.
- To learn to implement the front-end of the compiler.
- To learn to implement code generator.
- To learn to implement code optimization.

UNIT I INTRODUCTION TO COMPILERS & LEXICAL ANALYSIS

8

Introduction- Translators- Compilation and Interpretation- Language processors -The Phases of Compiler – Lexical Analysis – Role of Lexical Analyzer – Input Buffering – Specification of Tokens – Recognition of Tokens – Finite Automata – Regular Expressions to Automata NFA, DFA – Minimizing DFA - Language for Specifying Lexical Analyzers – Lex tool.

UNIT II SYNTAX ANALYSIS

11

Role of Parser – Grammars – Context-free grammars – Writing a grammar Top Down Parsing - General Strategies - Recursive Descent Parser Predictive Parser-LL(1) - Parser-Shift Reduce Parser-LR Parser- LR (0)Item Construction of SLR Parsing Table - Introduction to LALR Parser - Error Handling and Recovery in Syntax Analyzer-YACC tool - Design of a syntax Analyzer for a Sample Language

UNIT III SYNTAX DIRECTED TRANSLATION & INTERMEDIATE CODE GENERATION 9

Syntax directed Definitions-Construction of Syntax Tree-Bottom-up Evaluation of S-Attribute Definitions- Design of predictive translator - Type Systems-Specification of a simple type Checker-Equivalence of Type Expressions-Type Conversions. Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Back patching.

UNIT IV RUN-TIME ENVIRONMENT AND CODE GENERATION

9

Runtime Environments – source language issues – Storage organization – Storage Allocation Strategies: Static, Stack and Heap allocation - Parameter Passing-Symbol Tables - Dynamic Storage Allocation - Issues in the Design of a code generator – Basic Blocks and Flow graphs - Design of a simple Code Generator - Optimal Code Generation for Expressions– Dynamic Programming Code Generation.

UNIT V CODE OPTIMIZATION

8

Principal Sources of Optimization – Peep-hole optimization - DAG- Optimization of Basic Blocks - Global Data Flow Analysis - Efficient Data Flow Algorithm – Recent trends in Compiler Design.

45 PERIODS

LIST OF EXPERIMENTS:

1. Using the LEX tool, Develop a lexical analyzer to recognize a few patterns in C. (Ex. identifiers, constants, comments, operators etc.). Create a symbol table, while recognizing identifiers.
2. Implement a Lexical Analyzer using LEX Tool
3. Generate YACC specification for a few syntactic categories.
 - a. Program to recognize a valid arithmetic expression that uses operator +, -, * and /.
 - b. Program to recognize a valid variable which starts with a letter followed by any number of letters or digits.
 - c. Program to recognize a valid control structures syntax of C language (For loop, while loop, if-else, if-else-if, switch-case, etc.).
 - d. Implementation of calculator using LEX and YACC
4. Generate three address code for a simple program using LEX and YACC.
5. Implement type checking using Lex and Yacc.
6. Implement simple code optimization techniques (Constant folding, Strength reduction and Algebraic transformation)
7. Implement back-end of the compiler for which the three address code is given as input and the 8086 assembly language code is produced as output.

**30 PERIODS
TOTAL: 75 PERIODS**

COURSE OUTCOMES:

On Completion of the course, the students should be able to:

CO1:Understand the techniques in different phases of a compiler.

CO2:Design a lexical analyser for a sample language and learn to use the LEX tool.

CO3:Apply different parsing algorithms to develop a parser and learn to use YACC tool

CO4:Understand semantics rules (SDT), intermediate code generation and run-time environment.

CO5:Implement code generation and apply code optimization techniques.

TEXT BOOK:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques and Tools", Second Edition, Pearson Education, 2009.

REFERENCES

1. Randy Allen, Ken Kennedy, Optimizing Compilers for Modern Architectures: A Dependence based Approach, Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper and Linda Torczon, Engineering a Compiler, Morgan Kaufmann Publishers Elsevier Science, 2004.
4. V. Raghavan, Principles of Compiler Design, Tata McGraw Hill Education Publishers, 2010.
5. Allen I. Holub, Compiler Design in C, Prentice-Hall Software Series, 1993.

CO's-PO's & PSO's MAPPING

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	3	3	3	3	-	-	-	-	3	3	1	3	2	3	2
2	3	3	3	3	3	-	-	-	3	2	3	2	2	1	2
3	3	3	2	2	3	-	-	-	3	1	1	1	2	2	3
4	3	2	2	1	1	-	-	-	2	3	2	3	1	2	1
5	3	3	3	2	1	-	-	-	2	1	1	3	2	1	2
AVg.	3.00	2.80	2.60	2.20	2.00	-	-	-	2.60	2.00	1.60	2.40	1.80	1.80	2.00

1 - low, 2 - medium, 3 - high, ‘-’ - no correlation

CB3491

CRYPTOGRAPHY AND CYBER SECURITY

L T P C
3 0 0 3

COURSE OBJECTIVES:

- Learn to analyze the security of in-built cryptosystems.
- Know the fundamental mathematical concepts related to security.
- Develop cryptographic algorithms for information security.
- Comprehend the various types of data integrity and authentication schemes
- Understand cyber crimes and cyber security.

UNIT I INTRODUCTION TO SECURITY 9

Computer Security Concepts – The OSI Security Architecture – Security Attacks – Security Services and Mechanisms – A Model for Network Security – Classical encryption techniques: Substitution techniques, Transposition techniques, Steganography – Foundations of modern cryptography: Perfect security – Information Theory – Product Cryptosystem – Cryptanalysis.

UNIT II SYMMETRIC CIPHERS 9

Number theory – Algebraic Structures – Modular Arithmetic - Euclid's algorithm – Congruence and matrices – Group, Rings, Fields, Finite Fields

SYMMETRIC KEY CIPHERS: SDES – Block Ciphers – DES, Strength of DES – Differential and linear cryptanalysis – Block cipher design principles – Block cipher mode of operation – Evaluation criteria for AES – Pseudorandom Number Generators – RC4 – Key distribution.

UNIT III ASYMMETRIC CRYPTOGRAPHY 9

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY: Primes – Primality Testing – Factorization – Euler's totient function, Fermat's and Euler's Theorem – Chinese Remainder Theorem – Exponentiation and logarithm

ASYMMETRIC KEY CIPHERS: RSA cryptosystem – Key distribution – Key management – Diffie Hellman key exchange -- Elliptic curve arithmetic – Elliptic curve cryptography.

UNIT IV INTEGRITY AND AUTHENTICATION ALGORITHMS 9

Authentication requirement – Authentication function – MAC – Hash function – Security of hash function: HMAC, CMAC – SHA – Digital signature and authentication protocols – DSS – Schnorr Digital Signature Scheme – ElGamal cryptosystem – Entity Authentication: Biometrics, Passwords, Challenge Response protocols – Authentication applications – Kerberos

MUTUAL TRUST: Key management and distribution – Symmetric key distribution using symmetric and asymmetric encryption – Distribution of public keys – X.509 Certificates.

UNIT V**CYBER CRIMES AND CYBER SECURITY**

9

Cyber Crime and Information Security – classifications of Cyber Crimes – Tools and Methods – Password Cracking, Keyloggers, Spywares, SQL Injection – Network Access Control – Cloud Security – Web Security – Wireless Security

TOTAL:45 PERIODS**COURSE OUTCOMES:**

CO1: Understand the fundamentals of networks security, security architecture, threats and vulnerabilities

CO2: Apply the different cryptographic operations of symmetric cryptographic algorithms

CO3: Apply the different cryptographic operations of public key cryptography

CO4: Apply the various Authentication schemes to simulate different applications.

CO5: Understand various cyber crimes and cyber security.

TEXT BOOKS

1. William Stallings, "Cryptography and Network Security - Principles and Practice", Seventh Edition, Pearson Education, 2017.
2. Nina Godbole, Sunit Belapure, "Cyber Security: Understanding Cyber crimes, Computer Forensics and Legal Perspectives", First Edition, Wiley India, 2011.

REFERENCES

1. Behrouz A. Ferouzan, Debdeep Mukhopadhyay, "Cryptography and Network Security", 3rd Edition, Tata Mc Graw Hill, 2015.
2. Charles Pfleeger, Shari Pfleeger, Jonathan Margulies, "Security in Computing", Fifth Edition, Prentice Hall, New Delhi, 2015.

CO's-PO's & PSO's MAPPING

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	3	2	1	2	2	-	-	-	1	-	-	1	2	3	3
2	3	3	3	3	3	-	-	-	2	-	-	1	3	3	3
3	3	3	3	3	3	-	-	-	2	-	-	1	3	3	3
4	3	3	3	3	3	-	-	-	2	-	-	1	3	3	3
5	3	2	3	2	3	-	-	-	3	-	-	2	3	2	3
AVg.	3	2.6	2.6	2.6	2.8	-	-	-	2	-	-	1.2	2.8	2.8	3

1 - low, 2 - medium, 3 - high, “-“ no correlation

CS3551**DISTRIBUTED COMPUTING**

L T P C

3 0 0 3

COURSE OBJECTIVES:

- To introduce the computation and communication models of distributed systems
- To illustrate the issues of synchronization and collection of information in distributed systems
- To describe distributed mutual exclusion and distributed deadlock detection techniques
- To elucidate agreement protocols and fault tolerance mechanisms in distributed systems
- To explain the cloud computing models and the underlying concepts

UNIT I	INTRODUCTION	8
Introduction: Definition-Relation to Computer System Components – Motivation – Message -Passing Systems versus Shared Memory Systems – Primitives for Distributed Communication – Synchronous versus Asynchronous Executions – Design Issues and Challenges; A Model of Distributed Computations: A Distributed Program – A Model of Distributed Executions – Models of Communication Networks – Global State of a Distributed System.		
UNIT II	LOGICAL TIME AND GLOBAL STATE	10
Logical Time: Physical Clock Synchronization: NTP – A Framework for a System of Logical Clocks – Scalar Time – Vector Time; Message Ordering and Group Communication: Message Ordering Paradigms – Asynchronous Execution with Synchronous Communication – Synchronous Program Order on Asynchronous System – Group Communication – Causal Order – Total Order; Global State and Snapshot Recording Algorithms: Introduction – System Model and Definitions – Snapshot Algorithms for FIFO Channels.		
UNIT III	DISTRIBUTED MUTEX AND DEADLOCK	10
Distributed Mutual exclusion Algorithms: Introduction – Preliminaries – Lamport's algorithm – Ricart-Agrawala's Algorithm — Token-Based Algorithms – Suzuki-Kasami's Broadcast Algorithm; Deadlock Detection in Distributed Systems: Introduction – System Model – Preliminaries – Models of Deadlocks – Chandy-Misra-Haas Algorithm for the AND model and OR Model.		
UNIT IV	CONSENSUS AND RECOVERY	10
Consensus and Agreement Algorithms: Problem Definition – Overview of Results – Agreement in a Failure-Free System(Synchronous and Asynchronous) – Agreement in Synchronous Systems with Failures; Checkpointing and Rollback Recovery: Introduction – Background and Definitions – Issues in Failure Recovery – Checkpoint-based Recovery – Coordinated Checkpointing Algorithm - - Algorithm for Asynchronous Checkpointing and Recovery		
UNIT V	CLOUD COMPUTING	7
Definition of Cloud Computing – Characteristics of Cloud – Cloud Deployment Models – Cloud Service Models – Driving Factors and Challenges of Cloud – Virtualization – Load Balancing – Scalability and Elasticity – Replication – Monitoring – Cloud Services and Platforms: Compute Services – Storage Services – Application Services		

COURSE OUTCOMES:

Upon the completion of this course, the student will be able to

CO1: Explain the foundations of distributed systems (K2)

CO2: Solve synchronization and state consistency problems (K3)

CO3 Use resource sharing techniques in distributed systems (K3)

CO4: Apply working model of consensus and reliability of distributed systems (K3)

CO5: Explain the fundamentals of cloud computing (K2)

TOTAL:45 PERIODS

TEXT BOOKS

1. Kshemkalyani Ajay D, Mukesh Singhal, "Distributed Computing: Principles, Algorithms and Systems", Cambridge Press, 2011.
2. Mukesh Singhal, Niranjan G Shivaratri, "Advanced Concepts in Operating systems", McGraw Hill Publishers, 1994.

REFERENCES

1. George Coulouris, Jean Dollimore, Timo Kindberg, "Distributed Systems Concepts and Design", Fifth Edition, Pearson Education, 2012.
2. Pradeep L Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.
3. Tanenbaum A S, Van Steen M, "Distributed Systems: Principles and Paradigms", Pearson Education, 2007.
4. Liu M L, "Distributed Computing: Principles and Applications", Pearson Education, 2004.
5. Nancy A Lynch, "Distributed Algorithms", Morgan Kaufman Publishers, 2003.
6. Arshdeep Bagga, Vijay Madisetti, "Cloud Computing: A Hands-On Approach", Universities Press, 2014.

CO's-PO's & PSO's MAPPING

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	2	2	3	3	1	-	-	-	2	1	3	3	2	1	1
2	1	3	2	1	2	-	-	-	2	2	2	2	1	3	2
3	2	2	1	3	3	-	-	-	3	2	1	1	1	2	1
4	1	2	2	3	1	-	-	-	3	3	2	1	3	1	1
5	3	3	1	2	3	-	-	-	3	3	3	1	3	2	3
AVg.	1.8	2.4	1.8	2.4	2	-	-	-	2.6	2.2	2.2	1.6	2	1.8	1.6

1 - low, 2 - medium, 3 - high, ‘-’ - no correlation

CCS356

OBJECT ORIENTED SOFTWARE ENGINEERING

L T P C
3 0 2 4

COURSE OBJECTIVES:

- To understand Software Engineering Lifecycle Models
- To Perform software requirements analysis
- To gain knowledge of the System Analysis and Design concepts using UML.
- To understand software testing and maintenance approaches
- To work on project management scheduling using DevOps

UNIT I SOFTWARE PROCESS AND AGILE DEVELOPMENT 9

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models –Introduction to Agility-Agile process-Extreme programming-XP Process-Case Study.

UNIT II REQUIREMENTS ANALYSIS AND SPECIFICATION 9

Requirement analysis and specification – Requirements gathering and analysis – Software Requirement Specification – Formal system specification – Finite State Machines – Petrinets – Object modelling using UML – Use case Model – Class diagrams – Interaction diagrams – Activity diagrams – State chart diagrams – Functional modelling – Data Flow Diagram- CASE TOOLS.

UNIT III SOFTWARE DESIGN 9

Software design – Design process – Design concepts – Coupling – Cohesion – Functional independence – Design patterns – Model-view-controller – Publish-subscribe – Adapter – Command – Strategy – Observer – Proxy – Facade – Architectural styles – Layered - Client Server - Tiered - Pipe and filter- User interface design-Case Study.