# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Web Technologies

## 3290401

**COURSE OBJECTIVES:**

- To learn about the communication model and web architecture.
- To impart the design, development and implementation of Web Pages.
- To develop programs for Web using Scripting Languages.

---

**Unit – I:**

**Introduction to the Internet:** The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, Domains, Protocols, Hypertext Transfer Protocol, Internet Protocol, IP address, MAC Address, 3-Tier web architecture, Brief Overview of OSI models, web browser, Static and Dynamic web pages.

**Unit – II:**

**HTML**: Basic Syntax of HTML, Elements, Attributes, heading, paragraph, styles, formatting, comments, colors, links, images, tables, lists, forms, media.

**Unit – III:**

**Cascading Style Sheets:** Introduction to CSS, Syntax, Selectors, Box Model, Inline, internal and External CSS, colors, borders, margin, padding z-index.

**Unit – IV:**

**The Basics of JavaScript:** Introduction to javascript, syntax, comments, variables, constants, operators, data types, objects, strings, arrays, if else, switch, loops, function.

**Unit – V:**

**PHP&MYSQL:** Introduction to php, syntax, comments, variables, echo, datatypes, strings, operators, if else, switch, loops, function, arrays. Introduction to MySQL, clauses and simple query using select, where, order by, min max, count, avg, sum, like, alias, in., How to create website using

menu icon, tabs, navigations, search bar, Introduction to Content Management Systems (CMS)such as wordpress, joomla.

---

**RECOMMENDED BOOKS:**

1. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India
   Fficient
2. Web Technologies, Black Book, dreamtech Press
3. Web Technologies, Achyut Godbole, Atul Kahate, Tata McGraw-Hill
4. Principles of Web Design, Joel Sklar, Cengage Learning

---

**COURSE OUTCOMES:** After completing the course, the student will be able to:

**CO1: Distinguish** among various web designing technologies for website development

**CO2: Construct** webpages using HTML and CSS

**CO3: Model** website using JavaScript and PHP

**CO4: Design** Static and Dynamic website

**CO5: Explain** the working of web pages and data retrieval

---

**Course Articulation Matrix**

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | - |
| CO2 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 |
| CO3 | 2 | 2 | 2 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 |
| CO4 | 2 | 2 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 |
| CO5 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | - |

1 - Slightly; 2 - Moderately; 3 – Substantially

# Web Technologies

## 3290401

### List of Experiment

1. Write HTML program to learn various tags.
2. Write HTML program to change color and style of the content.
3. Write HTML program to open the MITS college web page using links.
4. Write HTML program to demonstrate list.
5. Write HTML program to demonstrate table.
6. Write HTML program to demonstrate forms.
7. Write HTML program to insert image.
8. Create a webpage with HTML describing your department use paragraph , lists tags and images.
9. Write programs to learn inline, internal and external CSS.
10. Write CSS programs to understand various selectors.
11. Write CSS programs to demonstrate padding properties
12. Write CSS programs to insert image using box Model.
13. Write javascript program to print “Hello World.”
14. Exploring if-else in javascript.
15. Exploring switch-case in javascript.
16. Create javascript programs with for loop, while and do-while loops
17. Arrays and Strings in Javascript.
18. Functions in Javascript
19. Write javascript program that changes HTML content.
20. Write javascript program that changes HTML attributes.
21. Write javascipt program to change the style of HTML elements.
22. Write javascript program to hide the HTML elements.
23. Write javascript program to show hidden HTML elements
24. Write program to demonstrate data types operators and print statement using PHP.
25. Exploring if-else, switch-case in PHP.
26. Exploring switch-case in PHP.
27. Create PHP programs with for loop, while and do-while loops
28. Array and String in PHP.
29. Functions in PHP.
30. Learn simple query using select and where clause.
31. Learn simple query using order by, min, max, count, avg, sum, like, alias, in keywords.

# Web Technologies

# 3290401

## List of Skill Based Mini Projects

1. Create a website for Conference
2. Create a website for Music Portfolio
3. Create a website for Photography Site
4. Create a website for Personal Portfolio
5. Create a website for Technical Documentation
6. Create a website containing all information for a person you admire.
7. Create a website for auction purpose
8. Create a e-commerce website
9. Create a website for books lover.
10. Create a website for online food delivery system

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Web Technologies

## 2290401(OLD)

**COURSE OBJECTIVES:**

- To learn about the communication model and web architecture.
- To impart the design, development and implementation of Web Pages.
- To develop programs for Web using Scripting Languages.

---

**Unit – I:**

**Introduction to the Internet:** The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, Domains, Protocols, Hypertext Transfer Protocol, Internet Protocol, IP address, MAC Address, 3-Tier web architecture, Brief Overview of OSI models, web browser, Static and Dynamic web pages.

**Unit – II:**

**HTML**: Basic Syntax of HTML, Elements, Attributes, heading, paragraph, styles, formatting, comments, colors, links, images, tables, lists, forms, media.

**Unit – III:**

**Cascading Style Sheets:** Introduction to CSS, Syntax, Selectors, Box Model, Inline, internal and External CSS, colors, borders, margin, padding z-index.

**Unit – IV:**

**The Basics of JavaScript:** Introduction to javascript, syntax, comments, variables, constants, operators, data types, objects, strings, arrays, if else, switch, loops, function.

**Unit – V:**

**PHP&MYSQL:** Introduction to php, syntax, comments, variables, echo, datatypes, strings, operators, if else, switch, loops, function, arrays. Introduction to MySQL, clauses and simple query using select, where, order by, min max, count, avg, sum, like, alias, in., How to create website using menu icon, tabs, navigations, search bar.

---

**RECOMMENDED BOOKS:**

1. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India
   Fficient
2. Web Technologies, Black Book, dreamtech Press
3. Web Technologies, Achyut Godbole, Atul Kahate, Tata McGraw-Hill
4. Principles of Web Design, Joel Sklar, Cengage Learning

---

**COURSE OUTCOMES:** After completing the course, the student will be able to:

**CO1: Distinguish** among various web designing technologies for website development

**CO2: Construct** webpages using HTML and CSS

**CO3: Model** website using JavaScript and PHP

**CO4: Design** Static and Dynamic website

**CO5: Explain** the working of web pages and data retrieval

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Microprocessor Design & Interfacing

## 3290402

**UNIT I:**

Technological trends, measuring performance: MIPS, CPI/IPC, Benchmark suite, Geometric and Arithmetic means, Speed up, Amdahl's law, Introduction to Microprocessor Systems, Historical development, Basic architecture and components, Overview of different microprocessor families, Microprocessor vs. Microcontroller.

**UNIT II:**

Introduction of 8085 Microprocessor: Architecture of 8085 processor. Register Architecture: Accumulator, Temporally Register and Flag Register. Program Counter, Stack pointer and Instruction register. Addressing Modes: Direct addressing mode and Register direct Addressing Mode. Register Indirect Addressing Mode, Immediate Addressing Mode and Implicit or Implied Addressing Mode.

**UNIT III:**

Introduction to Assembly Language Programming: Various Instructions Classifications: Instruction Format, Opcode, Operand and Hex code. Instruction Operation Status, Various Instruction Sets: Data Transfer Group Instructions, Arithmetic Group Instructions, Logical Group Instruction, Branch Group Instructions: Conditional and Unconditional and Machine control Instructions.

**UNIT IV:**

Parallel Computing: Flynn's Classification of Computer Architecture, Types of Parallelism, Parallel programming models. Multi processors and multi computers, Shared memory organization-Interleaved memory organization, CISC and RISC scalar processors- Super scalar processors-VLIW architecture- Multivector and SIMD computers.

**UNIT V:**

Intel's 14th Gen Core Processors, AMD's Ryzen 7000 series, chiplet-based designs, Intel's Meteor Lake, Intel's Alder Lake and Raptor Lake, quantum processors and neuromorphic chips.

**Recommended Books:**

1. Elmarsi. Navathe, Somayajulu, Gupta, "Fundamental of Database Systems", 4th Edition, Pearson Education, 2007

2. R Ramakrishanan, "Database Management Systems", McGraw Hill International Editions, 1998

3. Date, Kannan, Swaminathan, "An Introduction to Database Systems", 8th Edition Pearson Education, 2007.

4. Silberscatz, Korth, Sudarshan, "Database System Concepts", McGraw Hill 6th Edition. 2006

**COURSE OUTCOMES**

After completion of this course, the students would be able to:

CO1: **Apply** suitable assembly code for different problems.
CO2: **Apply** and Understand Assembly Language Instructions.
CO3: **Analyse** the metrics for measuring microprocessor performance.
CO4: **Evaluate** the architecture and classify the instruction set of 8085/8086 microprocessor.
CO5: **Understand** the latest technologies involved in the microprocessors.

# Microprocessor Design & Interfacing

# 3290402

## List of Experiments

1. Basic 8085 Microprocessor Architecture
2. 8085 Assembly Program to Add Two Numbers
3. 8085 Program to Subtract Two Numbers
4. 8085 Program to Multiply Two Numbers (Using Repeated Addition)
5. 8085 Program to Find the Largest of Three Numbers
6. 8085 Program to Check for Even or Odd Number
7. 8085 Program to Find Factorial of a Number
8. 8086 Program to Add Two 16-bit Numbers
9. 8086 Program to Multiply Two 16-bit Numbers
10. 8086 Program to String Length Calculation
11. 8086 Program to Reverse a String
12. 8086 Program to Sort an Array

# Microprocessor Design & Interfacing

# 3290402

## List of Skill Based Mini Projects

1. Traffic Light Control Using 8086
2. Temperature Control System Using 8086
3. Automated Irrigation System Using 8086
4. Smart Home Automation System Using 8086
5. Matrix Keypad and LCD Interface Using 8086
6. Simple Binary Counter
7. LED Blinking at a regular interval
8. Simple Switch-based Alarm System
9. Displaying a Single Character on 7-Segment Display
10. Simple Digital Stopwatch

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# SOFTWARE ENGINEERING

# 3290403

## COURSE OBJECTIVES

- To understand fundamental concepts of software engineering, including software development life cycles, methodologies, and processes.
- To Develop the ability to systematically gather, analyze, and document software requirements.
- To gain proficiency in software design methodologies and principles, enabling the creation of robust, scalable, and maintainable software architectures.
- To learn and apply various software testing methodologies to ensure software quality and reliability.

---

### Unit-I

**Introduction to Software Engineering:** Definition, software engineering-layered Technology, Software Characteristics and Components, Software model: Software Development of Life Cycle Model (SDLC), The Waterfall Model, Iterative Waterfall Model, Prototyping Model, Spiral Model, RAD Model, Selection criteria of model: Characteristics of Requirements, Status of Development Team, Users participation, Type of Project and Associated Risk.

### Unit - II

**Requirement Engineering:** Definition, Requirement Engineering Activity , Types of Requirement- Functional and Non-functional Requirements, User and System Requirements, Requirement Elicitation Methods, Requirement Analysis Methods, Requirement Documentation (SRS), Requirement Validation, Requirement Management.

### Unit – III

**Design Concept, Principle and Methods:** Design Fundamentals, Design Principles, Effective Modular Design, Design Representations, Architectural design, Procedural design, data Directed design, Real Time Design, Object Oriented Design, Coupling and Cohesion.

**Unit – IV**

**Software Metrics, Project Management and Estimation:** Metrics in Process and Project domains, Software Measurement, Software Quality Metrics, Project Management- Basics- People, Product, Process, Project, Estimation- Software Project Estimation, Decomposition Techniques- Function Point Estimation, Line of Code (LOC) based estimation, Empirical Estimation, COCOMO Model, Project Scheduling Techniques.

**Unit – V**

**Software Testing**: Definitions, Software Testing Life Cycle (STLC), Test Case Design, Strategic Approach to Software Testing- Verification & Validation , Strategic issues, Criteria for completion of Testing, Unit Testing, Integration Testing, Validation Testing, System Testing, Black Box Testing Techniques, White Box Testing Techniques, Acceptance Testing, introduction to security aspects and ethical AI in software engineering, Sustainable Software Lifecycle Management.

---

**RECOMMENDED BOOKS**


- Software Engineering, Sommerville, Pearson.
- Software Engineering: A Practitioner's Approach, Roger S. Pressman, McGraw Hill.
- Software Engineering, K.K. Agrawal & Yogesh Singh, New Age Publication.
- Software Engineering, Rajib Mall, PHI.


---

## COURSE OUTCOMES

After completion of the course students would be able to:

CO1. **Explain** the various fundamental concepts of software engineering.
CO2. **Recognize** the importance of requirements engineering in the software development lifecycle
CO3. **Identify Effective Software Design principles including cohesion and coupling**
CO4. **Implement** software metrics for estimating the cost, effort, and schedule of software projects.
CO5. **Examine** various testing techniques based on software requirements and design specifications.

---

## Course Articulation Matrix

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 2 | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | - |
| CO2 | 2 | 3 | 3 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 |
| CO3 | 2 | 2 | 3 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 |
| CO4 | 3 | 3 | 2 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 |
| CO5 | 3 | 2 | 2 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 |

1 - Slightly; 2 - Moderately; 3 – Substantially

# Department of Computer Science and Engineering

## Theory of Computation

## 3290404

### COURSE OBJECTIVES

- To understand computability, decidability, and complexity through problem solving.
- To analyse and design abstract model of computation & formal languages
- To understand and conduct mathematical proofs for computation and algorithms.

---

### COURSE OBJECTIVES

- To understand computability, decidability, and complexity through problem solving.
- To analyse and design abstract model of computation & formal languages
- To understand and conduct mathematical proofs for computation and algorithms.

---

### Unit-I

**Introduction to Theory of Computation:** Automata, Computability and Complexity, Alphabet, Symbol, String, and Formal Languages, Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and Mealy machines, Composite Machine, Conversion from Mealy to Moore and vice versa.

### Unit-II

**Types of Finite Automata:** Non Deterministic Finite Automata (NDFA), Deterministic finite automata machines, conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Pumping lemma, applications, Closure properties of regular languages, 2 way DFA.

### Unit-III

**Grammars:** Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, Rightmost and Leftmost derivations of Strings, ambiguity in grammar, simplification of context free grammar, killing null and unit productions, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar,. Chomsky Normal Form (CNF) and Greibach Normal Form (GNF).

### Unit-IV

**Turing Machine:** Techniques for construction. Linear bounded automata, Church Turing Thesis, Encoding of Turing Machine, Universal Turing machine Multitape, multihead and

multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem (PCB).

---

## RECOMMENDED BOOKS


- Introduction to Automata Theory Language & Computation, Hopcroft & Ullman, Narosa Publication.
- Element of the Theory Computation, Lewis & Christors,Pearson.
- Theory of Computation, Chandrasekhar & Mishra,PHI.
- Theory of Computation, Wood, Harper &Row.
- Introduction to Computing Theory, Daniel I-A Cohen,Wiley.


---

## COURSE OUTCOMES

After completion of the course students would be able to:

**CO1. Describe** the basic concepts of switching and finite automata theory & languages.

**CO2.Compute** abstract models of computing and check their power to recognize the languages.

**CO3. Analyse** the grammar, its types, simplification and normal form.

**CO4. Design** mathematical models to prove properties of languages, grammars and automata.

**CO5. Apply** automata theory, languages and computation in engineering application.

---

## Course Articulation Matrix

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 1 | 1 | | | 3 | 2 | | | | | | | | |
| CO2 | 1 | | 3 | 3 | 2 | | 3 | | 2 | | | | 1 | 2 |
| CO3 | | 1 | | 2 | 2 | 3 | | | 2 | | | 2 | 2 | 2 |
| CO4 | | | 1 | 1 | 1 | 2 | | | 2 | | | 2 | 1 | 2 |
| CO5 | | | | 1 | 2 | 2 | | | | | | 1 | 2 | 1 |

1 - Slightly; 2 - Moderately; 3 – Substantiall

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Computer Networks

## 3290405

### COURSE OBJECTIVES

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

---

### COURSE OBJECTIVES

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

---

### Unit-I

**Introduction:** Computer Network, Types- LAN, MAN & WAN,Data transmission modes- Serial & Parallel, Simplex, Half duplex & full duplex, Synchronous & Asynchronous transmission, Transmission medium- Guided & Unguided, Cables- Twisted pair, Coaxial cable & Optical fiber, Networking devices-Repeaters, Hub, Switch, Bridge, Router, Gateway, Modem, Proxy Server, Wireless router, & Wireless Access Point (WAPs). Performance Criteria- Bandwidth, Throughput, Latency (Delay), Propagation Time, Transmission time & Queuing Time, Network Standardization- OSI Reference Model & TCP/IP Reference Mode.

### Unit-II

**Physical Layer:** Network topologies- Bus, Ring, Star Topology & Mesh, Switching- Circuit switching, Message switching & Packet switching, Multiplexing; FDM – Frequency division multiplexing, WDM – Wavelength division multiplexing & TDM – Time division multiplexing, Wireless transmission- Electromagnetic spectrum, Radio transmission & Microwave transmission.

**Unit-III**

**Data Link Layer:** Introduction, Design issues, Services, Framing, Error control, Flow control, ARQ Strategies, Error Detection and correction, Parity bits, Cyclic Redundant Code (CRC), Hamming codes, MAC Sub Layer- The channel allocation problem, Pure ALOHA ,Slotted ALOHA, CSMA ,CSMA/CD,CSMA/CA,IEEE 802.3 frame format.

**Unit-IV**

**Network Layer& Transport Layer:** Introduction, Design issues, Services, Routing- Distance vector routing, Hierarchical routing , Link state routing, Shortest path algorithm- Dijkstra's Algorithm & Floyd–Warshall's Algorithm, Flooding, Congestion Control- Open Loop & Closed Loop Congestion Control, Leaky Bucket & Token bucket Algorithm. Connection Oriented & Connectionless Service, Port addressing basics

**Unit-V**

**Presentation, Session& Application Layer:** Introduction, Design issues, Presentation layer- Translation, Encryption & Compression. Session Layer – Dialog Control, Synchronization. Application Layer- Remote login, File transfer & Electronic mail.

Network Function Virtualization (NFV) - Architecture and Concepts, Programmable Networks - Introduction to P4, SmartNICS and P4 switches, Data Center Networking (DCN) – Introduction, DCN - Deep Dive (Network topologies, Container Network Interfaces)

.

---

**RECOMMENDED BOOKS**


- Behrouz A. Forouzan "Data Communication and Networking", McGraw – Hill Publications.
- Andrew Tanenbaum – Computer Networks, PHI
- Peterson and Davie, "Computer Networks, A systems Approach", 5th ed., Elsevier, 2011.
- Ying-Dar Liu, Ren-Hwang, Fred Baker, "Computer Networks: An open Source Approach", McGraw – Hill, 2001.


---

## COURSE OUTCOMES

After completion of the course students would be able to:

**CO1.** Outline the Data Communications System and its components.

**CO2.** Identify the different types of network topologies and protocols.

**CO3.** Enumerate the layers of the OSI model and function(s) of each layer.

**CO4.** Identify the different types of network devices and their functions within a network

**CO5.**Analyze the problems associated with various networking protocols and measure the Performance

---

**Course Articulation Matrix**

1 -

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 1 | 3 | | | | | | | | | | | | |
| CO2 | | | 2 | | | | 3 | | | | | | 1 | |
| CO3 | | | | | 2 | | | | | | | 2 | 2 | |
| CO4 | | | | | 1 | | | | | | | | 1 | |
| CO5 | | | | | 2 | | | | | | | 1 | 3 | 1 |

Slightly; 2 - Moderately; 3 – Substantially

# Computer Networks

# 3290405

## List of Experiments

### 1. Basic Network Configuration

- Setting up a local area network (LAN) using routers and switches.
- Configuring static IP addresses and testing connectivity using **ping** and **tracert/traceroute**.
- Verifying network connectivity with tools like **netstat**, **arp**, and **ipconfig/ifconfig**.

### 2. Protocol Implementation and Analysis

- **TCP/UDP Socket Programming**: Writing client-server programs for data exchange.
- Understanding and simulating the **handshake process in TCP**.
- Packet sniffing and analysis using tools like **Wireshark**.

### 3. Network Simulation

- Simulating networks with tools like **Cisco Packet Tracer**, **NS3**, or **GNS3**.
- Setting up and analyzing **routing protocols** (e.g., RIP, OSPF, EIGRP).

### 4. Routing and Switching

- Configuring routers and switches for VLANs.
- Setting up and verifying **static routing** and **dynamic routing protocols**.
- Troubleshooting network connectivity issues with routing tables.

### 5. Application Layer Experiments

- Setting up web servers and file servers (e.g., **Apache**, **NGINX**, or **FTP servers**).
- Configuring and testing **DNS** and **DHCP** servers.

## 6. Network Security

- Implementing **firewalls** and configuring access control lists (ACLs).
- Setting up secure connections using **VPNs**.
- Analyzing and mitigating common network attacks (e.g., ARP spoofing, DDoS attacks).

## 7. Network Performance Analysis

- Measuring **latency**, **throughput**, and **packet loss** using tools like **iperf**.
- Monitoring bandwidth usage and network performance using tools like **Nagios**, **Zabbix**, or **SolarWinds**.

# Computer Networks

# 3290405

## List of Skill Based Mini Project

1. Develop a simple client-server chat system using TCP or UDP.
2. Create a tool that mimics the working of the ping command, including ICMP packet creation and response handling.
3. Build a tool to calculate subnets, including network IDs, broadcast addresses, and usable host ranges.
4. Create an application that resolves domain names to IP addresses without using the system DNS.
5. Build a basic HTTP server to serve static web pages.
6. Develop a file-sharing system using TCP sockets.
7. Implement a packet-capturing application using Python's scapy or similar libraries.
8. Create an application to measure network bandwidth and latency.
9. Build a tool to monitor active connections, traffic usage, or specific port activity.
10. Design a basic firewall to block specific IPs or ports using Python or Linux iptables.
11. Build a basic brute force or dictionary attack tool for ethical hacking purposes (on test cases).
12. Create a tool that detects ARP spoofing attacks in a network.
13. Develop a system to stream video over UDP or TCP.

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Computer Networks

## 2290405(OLD)

### COURSE OBJECTIVES

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

---

### Unit-I

**Introduction:** Computer Network, Types- LAN, MAN & WAN,Data transmission modes- Serial & Parallel, Simplex, Half duplex & full duplex, Synchronous & Asynchronous transmission, Transmission medium- Guided & Unguided, Cables- Twisted pair, Coaxial cable & Optical fiber, Networking devices-Repeaters, Hub, Switch, Bridge, Router, Gateway, Modem, Proxy Server, Wireless router, & Wireless Access Point (WAPs). Performance Criteria- Bandwidth, Throughput, Latency (Delay), Propagation Time, Transmission time & Queuing Time, Network Standardization- OSI Reference Model & TCP/IP Reference Mode.

### Unit-II

**Physical Layer:** Network topologies- Bus, Ring, Star Topology & Mesh, Switching- Circuit switching, Message switching & Packet switching, Multiplexing; FDM – Frequency division multiplexing, WDM – Wavelength division multiplexing & TDM – Time division multiplexing, Wireless transmission- Electromagnetic spectrum, Radio transmission & Microwave transmission.

### Unit-III

**Data Link Layer:** Introduction, Design issues, Services, Framing, Error control, Flow control, ARQ Strategies, Error Detection and correction, Parity bits, Cyclic Redundant Code (CRC), Hamming codes, MAC Sub Layer- The channel allocation problem, Pure ALOHA ,Slotted ALOHA, CSMA ,CSMA/CD,CSMA/CA,IEEE 802.3 frame format.

### Unit-IV

**Network Layer& Transport Layer:** Introduction, Design issues, Services, Routing- Distance vector routing, Hierarchical routing , Link state routing, Shortest path algorithm- Dijkstra's Algorithm & Floyd–Warshall's Algorithm, Flooding, Congestion Control- Open

Loop & Closed Loop Congestion Control, Leaky Bucket & Token bucket Algorithm. Connection Oriented & Connectionless Service, Port addressing basics

**Unit-V**

**Presentation, Session& Application Layer:** Introduction, Design issues, Presentation layer- Translation, Encryption & Compression. Session Layer – Dialog Control, Synchronization. Application Layer- Remote login, File transfer & Electronic mail.

---

**RECOMMENDED BOOKS**


- Behrouz A. Forouzan "Data Communication and Networking", McGraw – Hill Publications.
- Andrew Tanenbaum – Computer Networks, PHI
- Peterson and Davie, "Computer Networks, A systems Approach", 5th ed., Elsevier, 2011.
- Ying-Dar Liu, Ren-Hwang, Fred Baker, "Computer Networks: An open Source Approach", McGraw – Hill, 2001.


---

**COURSE OUTCOMES**

Aftercompletionofthecoursestudentswould beable to:

**CO1.** Outline the Data Communications System and its components.

**CO2.** Identify the different types of network topologies and protocols.

**CO3.** Enumerate the layers of the OSI model and function(s) of each layer.

**CO4.** Identify the different types of network devices and their functions within a network

**CO5.**Analyze the problems associated with various networking protocols and measure the Performance

**CO6.** Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation

---

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Programming Lab (Java Programming)

## 3290406

### COURSE OBJECTIVES

- ☐ To understand fundamentals of Java programming such as variables, conditional and iterative execution, and methods.
- ☐ To understand fundamentals of object-oriented programming in Java, including defining Classes, invoking methods, using class libraries.
- ☐ To create a computer program to solve specified real world problems.

### Unit 1

Introduction to Java Programming, Introduction to Java and its features, Java Development Kit (JDK) installation and setup, Java development environment (IDE) usage, Java syntax and basic programming concepts, Variables, data types, and operators, Control structures: decision-making and loops

### Unit 2

Object-Oriented Programming in Java, Object-oriented programming (OOP) concepts: classes, objects, inheritance, polymorphism, and encapsulation, Java classes and objects, Constructors and methods, Inheritance and interfaces, Packages and access control.

### Unit 3

Exception Handling and File Handling, Exception handling: try-catch blocks, multiple catch clauses, and exception hierarchy, Throwing and catching exceptions, File I/O operations: reading from and writing to files, Working with streams and readers/writers, File handling best practices and error handling

### Unit 4

Java Collections Framework, Overview of Java Collections Framework (JCF), Lists, Sets, and Maps in JCF, ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, etc., Working with

collections: adding, retrieving, updating, and deleting elements, Iterators and iterating over collections

**Unit 5**

Multithreading and Java GUI Programming, Multithreading concepts: threads, synchronization, and inter-thread communication, Creating and managing threads in Java, Thread synchronization and deadlock prevention, Introduction to Java GUI (Graphical User Interface) programming Event-driven programming and handling GUI events, Swing components and layout management

## Reference Books

1. "Java: A Beginner's Guide" by Herbert Schildt (McGraw-Hill Education)
2. "Effective Java" by Joshua Bloch (Addison-Wesley Professional)
3. "Head First Java" by Kathy Sierra and Bert Bates (O'Reilly Media)
4. "Java: The Complete Reference" by Herbert Schildt (McGraw-Hill Education)
5. "Java Concurrency in Practice" by Brian Goetz et al. (Addison-Wesley Professional)

## Course Outcomes

After completion of the course students would be able to:

CO1. Apply object-oriented programming principles, including inheritance, polymorphism, and encapsulation, to design and implement robust Java applications.

CO2. Utilize the Java Collections Framework to effectively manage and manipulate data structures, such as lists, sets, and maps.

CO3. Create interactive graphical user interfaces (GUI) using Java Swing components, incorporating event-driven programming to enhance user experience.

# Programming Lab (Java Programming)

## 3290406

## List of Experiments

Experiment 1: Setting up the Java Development Environment

- Install the Java Development Kit (JDK) and an Integrated Development Environment (IDE).
- Write a simple "Hello, World!" program and execute it.

Experiment 2: Implementing Basic Control Structures

- Write a program that demonstrates the use of if-else statements for decision-making.
- Implement loops (for, while) to iterate over a set of numbers or perform a specific task.

Experiment 3: Creating and Manipulating Objects

- Design a class representing a student with relevant attributes and behaviors.
- Create multiple instances of the class and invoke methods to perform operations on the student objects.

Experiment 4: Inheritance and Polymorphism

- Create a base class and derived classes to showcase inheritance.
- Demonstrate polymorphism by invoking methods overridden in derived classes.

Experiment 5: Exception Handling

- Write a program that throws and catches different types of exceptions.
- Handle exceptions using try-catch blocks to prevent program termination.

Experiment 6: File Handling

- Read data from a text file and display its content.
- Write data to a file and verify the successful write operation.

Experiment 7: Working with Java Collections

- Create a collection (e.g., ArrayList) and perform operations like adding, retrieving, and removing elements.
- Iterate over a collection using iterators and demonstrate different collection classes.

Experiment 8: Multithreading

- Create multiple threads and execute them concurrently.
- Implement synchronization mechanisms to prevent thread interference.

Experiment 9: GUI Application Development

- Design a graphical user interface (GUI) using Swing components.
- Implement event handlers for GUI components, such as buttons or text fields.

Experiment 10: Comprehensive Project

- Design and implement a comprehensive Java project that incorporates concepts covered throughout the syllabus.
- Examples could include creating a student management system or a simple game using GUI elements.

**3290406**

# List of Skill Based Mini Projects

1. Student Management System:
   - Design a console-based application to manage student information.
   - Implement functionalities like adding, deleting, and displaying student records.
2. Library Management System:
   - Create a program to manage library operations, including book borrowing, returning, and searching.
   - Implement data structures to store book records efficiently.
3. Calculator Application:
   - Develop a GUI-based calculator application using Swing components.
   - Implement basic arithmetic operations and handle user input.
4. File Encryption and Decryption:
   - Create a program to encrypt and decrypt files using encryption algorithms.
   - Provide options for the user to select the encryption method and specify the file to encrypt/decrypt.
5. Quiz Application:
   - Develop a quiz application that presents multiple-choice questions to the user.
   - Implement a scoring system and display the result at the end of the quiz.
6. Bank Account Management System:
   - Design a program to manage bank accounts, including features like account creation, deposit, withdrawal, and balance inquiry.
   - Implement object-oriented concepts to model bank accounts and transactions.
7. Contact Management Application:
   - Develop a console-based application to manage contacts.
   - Implement functionalities like adding contacts, searching by name, and displaying contact details.
8. Tic-Tac-Toe Game:
   - Create a GUI-based Tic-Tac-Toe game using Swing components.
   - Implement game logic to handle player turns and determine the winner.
9. Weather Forecast Application:
   - Develop a program that retrieves weather data from an API and displays it to the user.
   - Implement features like displaying current weather, forecast for multiple days, and location-based search.
10. Online Shopping System:

- Design a simple online shopping system with features like browsing products, adding items to the cart, and placing orders.
- Implement shopping cart functionality and user authentication.