Centurion University of Technology and Management Odisha

Choice Based Credit System

Course Structure & Syllabus

BASKET - IV

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT

School of Engineering & Technology

2022

Course Structure

Basket - IV

Computer Science and Engineering

Course Code	Course Title	Credits	Type T+P+PJ	Prerequisite
CUTM1024	IT Infrastructure Management	6	2--2--2	
CUTM1025	Cloud Practitioners	2	0--2--0	
CUTM1026	Wireless Networks	3	1--2--0	
CUTM1027	Information Security	3	1--2--0	
CUTM1602	Programming in C	4	1--2--1	
CUTM1603	Data Structures	4	1--2--1	
CUTM1030	Advanced Web Programming	4	1--2--1	
CUTM1031	Java Technologies	4	2--1--1	
CUTM1032	Operating System Concepts	3	1--1--1	
CUTM1033	Database Creation and Maintenance	4	2--1--1	
CUTM1034	Database Cluster Administration and Security	4	2--1--1	
CUTM1035	Data Warehousing and Data Mining	4	2--2--0	
CUTM1036	Android App Development	6	2--2--2	
CUTM1037	Mathematical Problem Solving	4	2--2--0	
CUTM1038	Formal Language and Automata Theory	3	2--1--0	

IT Infrastructure Management

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1024	IT Infrastructure Management	2-2-2(6)	NIL

Objective

- To learn how to install DOS and NON-DOS OS
- Assembling and dissembling computer and laptop
- To configure network
- To develop skills to build and manage IT infrastructure in Enterprise level

Course outcome

- Able to install Windows and Linux both in Physical and VM
- Able to troubleshoot computer network
- Able to get job as network administrator

Course content

Module I: Introduction to BIOS (10 Hrs)

BIOS – Information, Configuration, Monitoring and Diagnostics, Types of Slots, Chipsets, Connectors and Bus Speed, Display Types, Refresh Types, Aspect ratios. Drives, Input and output devices, Assembling and dissembling of computer and Laptop.

Practice:

- BIOS setup
- Assembling and dissembling of computer and Laptop

Module II: Installation of DOS and Non-DOS OS (14 Hrs)

Installation of DOS and Non-DOS operating system – Automatic and Manual Configurations, Both in physical hardware and on Virtual machine (VMWare, VirtualBox). Application Installation both in DOS / Non DOS – Webserver, Email server, FTP server, DNS server.

Practice:

- Installation of DOS and Non-DOS operating system
- Application Installation both in DOS / Non DOS

Module III: Network Fundamentals (12 Hrs)

Network Fundamentals – Decimal and Binaries, Address Formats - TCP Model, IP Model, OSI Model, IPv4, IPv6 – Subnetting NetworkingOS - Software architecture, Control and forwarding planes, Routing Engine and Packet Forwarding Engine

Practice:

- Setting up a Network
- Configuring IPV4 and IPV6
- Implementing subnetting

Module IV: Introduction to CLI (12 Hrs)

Identify the concepts, operation or functionality of the networking user interface - CLI functionality, CLI modes, CLI navigation, Filtering output, Modifying, managing, and saving configuration files, Viewing, comparing, and loading configuration files.

Practice:

- Configuring CLI functionality and CLI modes, CLI navigation
- How to filter Managing and filtering output

Module V: Configuration and Monitoring (12 Hrs)

Configuration, User accounts, User authentication methods, Interface types and properties, Configuration groups, Configuration archival, Logging and tracing, Identify methods of monitoring or maintaining Junos device, Show commands, Monitor commands, Interface statistics and errors, Networking OS installation and upgrades

Practice:

- Setting up user account
- Implementing User authentication
- Networking OS installation and upgrades

Module VI: Fundamentals of Windows Administration (10 Hrs)

Windows Administration – Active Directory Configuration, Administration, Troubleshooting and User management. DHCP Server, Windows Deployment Services, Security Policies and Management, IIS, Server Management Tools – Event Viewer, Remote management.

Practice:

- Set up Windows Administration
- Troubleshooting and User management

- Implementing DHCP server
- Security Policies and Management

Module VII: Introduction to Linux Administration (10 Hrs)

Linux Administration - Basic Commands, Package Management, Text manipulation, network and system Diagnostics

Practice:

- Linux set up
- Linux package management
- Troubleshoot and system diagnotics

Text Books

- ICT Infrastructure Management, by Great Britain: Office of Government Commerce

Cloud Practitioner

Code	Course Title	Credit	T-P-PJ
CUTM1025	Cloud Practitioner	2	0-2-0

Objective

- Understanding fundamentals of Cloud and its basic infrastructure
- Learn about account management, billing and pricing
- Acquire knowledge on security model and compliance concepts
- Learn how to use different core services of Cloud
- Analyze and Understanding the functioning of different components involved in Amazon web services. Acquire cloud technology skill which helps students in getting jobs in different MNCs

Course outcome

- Explain AWS cloud values and Implement different policies using its services
- Get skills to analyze and manage billing and pricing used for the resources
- Design and deployment of different applications using its services
- Leads to the next level of preparation i.e. Associate and Professional level

Course content

Module I: Overview of Cloud Concepts and Billing (4 HRS)

Overview of Cloud Computing, Advantages of the Cloud, Overview of AWS, AWS Organizations, Fundamentals of Pricing, Total Cost of Ownership, Simple Monthly Calculator, AWS Billing & Cost Management, Billing Dashboard, AWS Whitepapers & Documentations

Practice:

- Exploring AWS Billing Dashboard
- Set up budget alarm
- Consolidated Billing

Module II: Cloud Security (4 HRS)

AWS Global Infrastructure, AWS Management Console, AWS Services & Service Categories, AWS Shared Responsibility Model, Cloud Security, AWS Identity and Access Management, Securing Accounts, Securing Data, Working to Ensure Compliance.

Practice:

- Setup account for AWS console management
- Exploring services of AWS
- Configuring AWS IAM
- Securing AWS Account

Module III: Networking and Content Delivery (5 HRS)

Networking Basics, Amazon Virtual Private Cloud, VPC Security, Build Your VPC and Launch a Web Server, Amazon Route53, Content Delivery Networks, Edge Location, Amazon CloudFront

Practice:

- Create a Virtual Private Cloud
- Configure Route53
- Application of CloudFront in S3

Module IV: Compute Services (5 HRS)

Overview of Compute Services, Amazon Elastic Compute Cloud, Amazon EC2 versus Managed Services, Amazon EC2 Cost Optimization, Container Services, AWS Lambda, Amazon Elastic Beanstalk

Practice:

- Launch an EC2 instance
- Writing AWS Lambda function
- Deploy code using Amazon Beanstalk

Module V: Storage Services and Databases (4 HRS)

AWS Elastic Block Store, Configuring EBS, AWS Simple Storage service, AWS S3 Glacier, Amazon Relational Database Service, Build a Database Server, Amazon Aurora, Amazon DynamoDB, Amazon Redshift..

Practice:

- Creating Buckets using S3
- Sic website hosting using S3
- Creating Database using DynamoDB
- Creating database using RDS
- Configuring Redshift

Module VI: Cloud Architecture (4 HRS)

Cloud Architecture, AWS Well-Architected Framework Design Principles, Operational Excellence, Security, Reliability and High Availability, Performance Efficiency, Cost Optimization, AWS Trusted Advisor

Practice:

- Well Architected framework design
- Implementing Cost optimization
- Application of AWS Trusted Advisor

Module VII: Auto Scaling and Load Balancing (4 HRS)

AWS Auto Scaling, Elastic Load Balancing, Amazon CloudWatch, Scale and Load Balance your Architecture, AWS Simple Notification Service and Simple Queue service.

Practice:

- Configure AWS Auto scaling
- Application of Load Balancing
- Implementing AWS CloudWatch
- Setup SNS email using CloudWatch

Online Resource: http://aws.amazon.com/training/awsacademy

http://aws.amazon.com/awseducate

Wireless Networks

Code	Course Title	(Credit)	T-P-PJ
CUTM1026	Wireless Networks	3	1-2-0

Objective

- Describe the features and functions of WLAN components.
- Skills needed to install, configure, and troubleshoot WLAN hardware peripherals and protocols.
- Understand the Wi-Fi communications process and security standards.

Course outcome

- Able to install, configure and troubleshoot WIFI Network

Course content

Module I: Wireless LAN Infrastructure Devices (7 Hrs)

Access Points, Bridges, Workgroup Bridges, PCMCIA Cards , Serial and Ethernet Converters , USB Devices, PCI/ISA Devices, Residential Gateways, Enterprise Gateways, Access Point Modes, Fixed or Detachable Antennas, Advanced Filtering Capabilities, Configuration and Management, Point-to-Point Protocol over Ethernet (PPPoE), Network Address Translation (NAT), Port Address Translation (PAT), Ethernet switching , Virtual Servers, Print Serving, Fail-over routing, Virtual Private Networks (VPNs) , Dynamic Host Configuration Protocol (DHCP) Server and Client, Configurable Firewall

Practice:

- Configuring PCMCIA Cards, Wireless Cards
- Configuring PPPoE, NAT, PAT
- Configuring DHCP
- Configuring Basic Firewall

Module II: Wireless LAN Organizations and Standards (7 Hrs)

802.11, 802.11b, 802.11a, 802.11g, Bluetooth, Infrared, HomeRF, FCC, IEEE, WECA, WLANA , IrDA , ETSI, ISM and UNII Bands, 900 MHz ISM Band, 2.4 GHz ISM Band, 5.8 GHz ISM Band, Lowe Band, Middle Band, Upper Band, Power Output Rules, Point-to-Multipoint (PtMP), Point-to-Point (PtP), Wireless Ethernet Compatibility Alliance, Competing Technologies.

Practice

- Configure P2P and P2MP with different Frequencies

Module III: 802.11 Network Architecture(5 Hrs)

Authentication, Association, Open System authentication, Shared Key authentication , Secret keys & certificates, AAA Support, BSS, ESS, IBSS, SSID, Infrastructure Mode , Ad hoc Mode, Roaming, PSP Mode, CAM, Beacons, TIM, ATIM, ATIM Windows

Practice

- Configuring Multiple SSIDs - Open System
- Configuring Multiple SSIDs - Using Shared Key
- Configuring Multiple SSIDs - Using AdHoc Mode

Module IV: Wireless LAN Installations(6 Hrs)

Multipath, Effects of Multipath, Decreased Signal Amplitude, Increased signal Amplitude, Node, Near/Far, RF Interference, All-band interference, System throughput, Co-location throughput, Weather, Types of Interference, All-band Interference and Troubleshooting

Practice

- Connectivity between two devices and analysis interference between the two Access Points

Module V: Physical Layers(6 Hrs)

The difference between wireless LAN and Ethernet frames, Layer 3 Protocols supported by wireless LANs, Distributed Coordination Function (DCF), Point Coordination Function (PCF), CSMA/CA vs. CSMA/CD,

Practice

- Building theStar, Bus, Ring & Mesh Topology Connect Computer Networks in Organizations

Module VI: MAC(6 Hrs)

Interframe spacing, RTS/CTS, Dynamic Rate Selection, Modulation and coding, Collision Handling, Fragmentation, Dynamic Rate Shifting (DRS), Distributed Coordination Function, Point Coordination Function, Interframe Spacing.

Practice

- Filtering the MAC address in the firewall

Module VII: Authentication Methods(5 Hrs)

802.1x and EAP, Wireless DMZ, User Authentication

Practice

- Understanding the 802.1 Strandedfor the communication

Reference :-
Wireless Information NetworksTextbook by Allen H. Levesque and Kaveh Pahlavan
/

Information Security

Code	Course Title	(Credit)	T-P-PJ
CUTM1027	**Information Security**	3	1-2-0

Objective

- The objective of this course is to focus on the models, tools, and techniques for enforcement of security.
- Students will learn security from multiple perspectives.
- Get skills to understand, design and implement appropriate security technologies and policies to protect computers and digital information

Course outcome

- Will gain familiarity with computer network, defences against them, and forensics to investigate the aftermath.
- Develop a basic understanding of Risk assessment
- Develop an understanding of security policies as well as protocols to implement such policies
- Can perform job role of IT auditor

Course content

Module I: Threats, Attacks and Vulnerabilities (8Hrs)

Viruses, Crypto-malware, Ransomware, Worm, Trojan, Rootkit, Keylogger, Adware Spyware, Bots, Logic bomb Backdoor, Social engineering, Application/service attacks, Injection, Wireless attacks, Types of actors, Attributes of actors, Active reconnaissance , Passive reconnaissance

Practice: -

- Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars.

Module II: Technologies and Tools. (6 Hrs)

Weak security configurations, Personnel issues, Baseline deviation, License compliance violation (availability/integrity), Asset management, Authentication issues.

Practice: -

- Study of packet sniffer tools like wireshark, ethereal, tcpdump etc. Use the tools to do the following 1. Observer performance in promiscuous as well as non-promiscous mode. 2. Show that packets can be traced based on different filters.

Module III: Antivirus and Firewalls (8 Hrs)

Antivirus, File integrity check, Host-based firewall, Application whitelisting, Removable media control, Advanced malware tools , Patch management tools , UTM , DLP , Data execution prevention , Web application firewall , Connection methods , Mobile device management concepts , Enforcement and monitoring , Deployment models , Protocols.

Practice: -

- Use of iptables in linux to create firewalls

Module IV: Security Policies and Issues. (8 Hrs)

Regulatory compliance, Frameworks, Policies, Controls, Procedure, Patching,Verifications and quality control, Security issues associated with context-based authentication, Security issues associated with identities, Security issues associated with identity repositories**.**

Practice: -

- Implement a code to simulate buffer overflow attack

Module V: Cybercrime (8 Hrs)

Cybercrimes and data breaches, Licensing and intellectual property requirements, Import/export controls, Trans-border data flow, Privacy policy requirements, Identify threats and vulnerabilities.

Practice: -

- Download and install nmap. Use it with different options to scan open ports,performOS fingerprinting, do a ping scan, tcp port scan, udp port scan, etc.

ModuleVI: Risk Analysis (6Hrs)

Risk assessment/analysis, Risk response, Countermeasure selection and implementation, Applicable types of controls (e.g., preventive, detective, corrective).

Practice: -

- Detect ARP spoofing using open source tool ARPWATCH

ModuleVII: Access Control (6 Hrs)

Federated Role Based Access Control (RBAC), Rule-based access control, Mandatory Access Control (MAC), Discretionary Access Control (DAC), Attribute Based Access Control (ABAC)

Practice: -

- Use the Nessus tool to scan the network for vulnerabilities.

Text Books:

- William Stallings, "Cryptography and Network Security", Fourth edition, PHI
- Schneier, Bruce, "Applied Cryptography", John Wiley and Sons
- Douglas R. Stinson, "Cryptography: Theory and Practice", CRC Press
- Behrouz A. Forouzan , "Cryptography and Network Security", Mc-Graw Hill

Programming in C

Course Title	Code	Type of course	T-P-PJ	Prerequisite
Programming in C	CUTM 1602	Theory & Practice, Project	1-2-1	Nil

Objective

- Provide basic knowledge of programming tools and techniques.
- Familiarize the programming environment and syntax of C programming.
- Understand the working of basic programming constructs.
- Acquire C coding skill

Course outcome

- The students will able to apply programming skills to problem solving.
- The student will able to write 150 to 200 line programs without any error.
- Create job opportunities in system software development area

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Internal Theory	20	Written Examination
	Internal Practice	25 (20+05)	Lab Work + Learning Record
	Internal Project	25 (20+05)	Project (Work +Report)
External Examination	External Theory	30	Written Examination
	External Practice	25	Lab Work
	External Project	25	Project Work
Total		*150*	

Course outline

Module I :Problem Solving Techniques (10Hours)

Problem solving techniques : Algorithm, flow chart; Structure of C program, Character set, Identifiers, Keywords, Data Types, Constants and Variables, Input-output statements, relational and logical operators, increment and decrement operators, conditional operator, bit-wise operators, assignment operators, expressions, type conversions, conditional expressions, precedence and order of evaluation.

Module II : Control Structures (7Hours)
Statements and blocks, if and switch statements, loops : -while, do-while and for statements, break, continue, goto.

Module III : Array(7Hours)
Arrays-concepts, declaration, definition, accessing elements, two-dimensional and multi-dimensional arrays, applications of arrays.

Module IV : Functions(15 Hours)
Designing structured programsFunctions, parameter passing, user defined functions, recursive functions, storage classes- extern, auto, register, static, scope rules.

Module V : Pointer (15 Hours)
Pointers- concepts, initialization of pointer variables, pointers and function arguments, address arithmetic, Character pointers and functions, pointers to pointers, pointers and multidimensional arrays, dynamic memory management functions, command line arguments.

Module VI: Structures(11Hours)
Derived types-structures-declaration, definition and initialization of structures, accessing structures, nested structures, arrays of structures, structures and functions, pointers to structures, self-referential structures, unions,

Module VII: Files (10Hours)
Input and output–concept of a file, text files and binary files, file I/O operations, Typedef, bit fields, Command line arguments.

List of Projects:

1. LASER Demonstration
2. Electronic Voting Machine
3. Elementary Database Management System
4. Student Information System
5. Parking System
6. Bank Management System
7. Result System
8. Quiz Game Project
9. Library Management System
10. Pacman Game

Text Books :

1. E. Balaguruswamy “Programming in C”, Tata McGraw Hill 3rd Edition
2. Y. Kanetkar, “Let us C”, BPB Publications-9th edition.

Reference Books :

1. H. Scheldt, “C The Complete Reference”, Tata McGraw Hill
2. B.W. Kernighan & D.M. Ritchie, "C Programming Language", PHI.
3. Gotterfried, Schaum Series-“C Programming”.

Data Structures

Code	Course Title	Credit	T-P-PJ
CUTM1603	Data Structures	4	1-2-1

Objective

- To familiarize the students with techniques of algorithm analysis and recursive procedures
- To make students implement linked data structures such as linked lists and binary trees
- To introduce several sub-quadratic sorting algorithms including - quick sort, merge sort and heap sort
- To acquaint students solve graph traversal techniques
- Skill to choose appropriate data structure as applied to specified problem definition. Ability to analyze algorithms and algorithm correctness

Course outcome

- Evaluate algorithms and data structures in terms of time and memory complexity of basic operation steps
- Ability to formulate programs that cover basic static and dynamic data structures, and relevant algorithms as applicable to them covering search and sorting algorithms, traversal techniques, etc.
- Determine and demonstrate bugs in program, recognize needed basic operations with data structures
- Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures
- Able to get jobs in different IT firms as developer.

Course content

Module I:Basics Of Data Structures and Array (06 hrs)
Data Structures, Types of Data Structures, Algorithm analysis, Complexity Analysis :Time complexity and Space complexity , Asymptotic Notations (Big Oh, Big Omega, Big Theta, Little Oh, Little Omega), Data Structure Operations, Application of Data structures.

Arrays: Memory Representation of an array, Operations on an array.

Programs:
1.Write a program to perform the following menu driven program on the input array.
a. Insertion
b. Deletion
c. Searching
d. Sorting
e. Merging
f. Display
g. Exit

Module II: Stack & Queue (12 hrs)
Stacks, operations on stack and its applications (Infix to Postfix Conversion, Infix to prefix conversion, Prefix evaluation and Postfix Evolution).

Queues, Operations on Queue, De-queues, Operations on Dequeue, Priority Queues .

Programs:

1.Write a program to perform the following menu driven program on the STACK.
a. Push
b. Pop
c. Display
d. Exit

2. Write a program to perform the following menu driven program on the Queue.
a. Insertion
b. Deletion
c. Display
d. Exit

Module III: Linked List (14 hrs)

Linked lists: Single Linked List and Operations on Single Linked List (Creation Insertion , Deletion , Sorting and Reverse).Circular linked list, Double linked list ,

Programs:

1.Write a program to create a single linked list perform the following menu driven program.
a. Insertion at front

b. Insertion at end
c. Insertion at particular position
d. Deletion at front
e. Deletion at end
f. Deletion at particular position
g. Display

2.Write a program to create a circular linked list perform the following menu driven program.
a. Insertion at front
b. Insertion at end
c. Insertion at particular position
d. Deletion at front
e. Deletion at end
f. Deletion at particular position
g. Display

3.Write a program to create a Double linked list perform the following menu driven program .
a. Insertion at front
b. Insertion at end
c. Insertion at particular position
d. Deletion at front
e. Deletion at end
f. Deletion at particular position
g. Display

Module IV: Stack & Queue Implementation Using Linked List (4 hrs)
Stack implementation using Linked List and Queue implementation using Linked List and operations.

Programs:

1.Write a program to implement Stack Using Linked List.
2.Write a program to implement Queue Using Linked List.

Module V: Trees (12 hrs)
Trees and hierarchical orders ,Introduction to trees ,Representation of tree, Abstract trees , Tree traversals , Forests , Ordered trees , Binary trees , Perfect binary trees , Complete binary trees , Search trees : Binary search trees , AVL trees.

Programs:
1.Write a program to create Binary tree and display it .
2. Write a program to create a BST and display it.
3. Write a program to Print all pairs from two BSTs whose sum is greater than the given value.
4.Write a program to remove duplicate entries from the BST.
5.Write a program to create a AVL tree and display it.

Module VI: Searching & Sorting (14 hrs)
Searching & Sorting algorithms , Objectives of Searching , The Sequential Search , Analysis of Sequential Search , The Binary Search , Analysis of Binary Search , Introduction to sorting , Insertion sort , Bubble sort , Heap sort ,Merge sort ,Quick sort, Radix Sort.

Programs:
1.Write a program to perform linear and binary search.
2. Write a program to perform selection sort , Bubble sort and Insertion sort.
3. Write a program to perform merge and quick sort.
4.Write a program to perform Heap sort.

Module VII: Graphs and Hashing (8 hrs)

Graph Terminology, Graph Traversal Techniques: Breadth First Search, Depth First Search, Hashing: Introduction to hash tables, Hash functions, Collision Resolution : Chained hash tables, Linear Probing.

Programs:
1.Write a program to perform Linear Probing.
2. Write a program to perform Double Hashing

Text Books:

1. Data Structures, Algorithms and Applications , Sartaj Sahani, 2nd Edition.
2. Data Structures and Algorithms, Michael T.Goodrich, R, Tamassia and D.Mount, wiley Student Edition, 7th edition, John Wiley and Sons.

Reference Books:

1. Data Structures and Algorithms Analysis by Mark Allen Weiss.
2. Data Structures and Algorithms , 3rd edition, Adam Drozdek, Cengage Learning.

Projects:

- Contact book application using Doubly Linked List
- Dictionary using Binary trees
- Snake Game
- Chess Game
- AVL -Insertion and Deletion
- Travel Planner (Graphs)
- Tic-Tac-Toe Game
- Library Management System
- Project management System
- User defined(Student Choice)

Text Books:

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahani, 2nd Edition.

2. Data Structures and Algorithms in C++, Michael T.Goodrich, R, Tamassia and D.Mount, wiley Student Edition, 7th edition, John Wiley and Sons.

Reference Books:

1. Data Structures and Algorithms Analysis in C++ by Mark Allen Weiss.

2. Data Structures and Algorithms in C++, 3rd edition, Adam Drozdek, Cengage Learning.

List Of Projects:

1. Contact book application using Doubly Linked List
2. Dictionary using Binary trees
3. Snake Game
4. Chess Game
5. AVL -Insertion and Deletion
6. Travel Planner (Graphs)
7. Tic-Tac-Toe Game
8. Library Management System
9. Project management System
10. User defined(Student Choice)

Note: 1 credit theory=10 hrs lecture, 1 credit practice/project=12.5 hrs lab/workshop/field work in a semester

Advanced Web Programming

Code	Course Title	Credit	T-P-PJ
CUTM1030	Advanced Web Programming	4	1-2-1

Objective

- Understand client server architecture and able to use the skills for web project development.
- Create job opportunities as a web developer

Course outcome

- Develop a static, interactive and well-formed webpage using JavaScript, CSS3 and HTML5.
- Use PHP7 to improve accessibility of a web document.
- Gain necessary skills for designing and developing web applications.

Course content

Module I: Web Programming Concepts(7hrs)

Architecture of the Web (1)

HTTP Protocols(1)

Difference HTTP1.0 and HTTP 1.1,Stateless nature of the protocol,Methods (GET, POST, HEAD, PUT, DELETE),HTTP session,Statuscodes,Persistentconnections,HTTPS

HTML(1)

Document Object Model (DOM),Elements,Events

HTML 5(2)

Elements,Objects,Events,Canvas,Audio& Video Support,Geo-location Support

CSS(2)

Styling HTML with CSS,Inline Styling (Inline CSS),External Styling (External CSS),CSS Fonts,The CSS Box Model,The id Attribute,The class Attribute,HTML Style Tags

Practice

1. Write an HTML code to display your CV on a web page.
2. Write an HTML code to create a Home page having three links: About Us, Our Services and Contact Us. Create separate web pages for the three links.
3. Write an HTML code to create a Registration Form. On submitting the form, the user should be asked to login with this new credentials.
4. Write an HTML code to create your Institute website, Department Website and Tutorial website for specific subject.

5. Write an HTML code to create a frameset having header, navigation and content sections.
6.Write an HTML code to demonstrate the usage of inline CSS.
7. Write an HTML code to demonstrate the usage of internal CSS.
8. Write an HTML code to demonstrate the usage of external CSS.
9: Design your own website using HTML CSS
10: Design form using HTML and apply CCS

Module II: JavaScript & jQuery(14 hrs)

JavaScript (10)

Introduction to JavaScript:Variable, statements, Operators, Comments, constructs, Functions, expressions,Javascriptconsole,Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date,Methods,Arrays, Array Methods,Booleans, Comparisons

Control Structures:Conditions, Switch,Loop For, Loop While, Break

Functions: Function Definitions, Function Parameters, Function Invocation, Function Closures

Objects: Object Definitions, Object Properties, Object Methods, Object Prototypes

Object Oriented Programming:
Method,Constructor,Inheritance,Encapsulation,Abstraction,Polymorphism,JavascriptValidations, Document Object Model, Document and Events (DOM Manipulation)

HTML DOM: DOM Intro, DOM Methods, DOM Document, DOM Elements, DOM HTML,

DOM CSS,DOM Animations, DOM Events, DOM EventListener, DOM Navigation, DOM

Nodes, DOM Nodelist,Debugging,Type Conversion, Regular expressions, Errors, Debugging

Forms: Forms Validation, Forms API,JS Browser BOM, Window, Screen, Location, History,

Navigator, Popup Alert, Timing, Cookies,JavascriptWindows,Pushing code quality via

JSLinttool,Security in Java Script

jQuery(4)

Basics of jQuery,jQuery selection and events, jQuery Effects,jquery traversal and

manipulation,Data attributes and templates, jQuery Plugins,Jquery / Google Web Toolkit

Practice:

1.Write a Java script to prompt for users name and display it on the screen.

2. Design HTML form for keeping student record and validate it using Java script.

3. Write programs using Java script for Web Page to display browsers information.

4: Validate form page using JavaScript

5: use JQuery effect in page

6. Write a jQuery Code to Find the data passed with the on() method for each element.

7.Find the position of the mouse pointer relative to the left and top edges of the document.

8.Count the number of milliseconds between the two click events on a paragraph

9.Find all the text nodes inside a paragraph and wrap them with an italic tag

Module III: AJAX& JSON(8 hrs)

AJAX(3)

Design Introduction to Ajax,Web services and Ajax,Ajax using HTML, CSS, JavaScript,Ajax Framework and DOM,XMLHttpRequest,Ajax Architecture

Working with JSON (5)

JSON – Introduction,Need of JSON,JSON Syntax Rules,JSON Data - a Name and a Value,JSONObjects,JSONArrays,JSON Uses JavaScript Syntax,JSONFiles,JSON& Security Concerns, Cross Site Request Forgery (CSRF), Injection Attacks,JSXMLHttpRequestfunctions,JavaScriptXMLHttpRequest& Web APIs,JSON& Client Side Frameworks,JSON& Server Side Frameworks,Replacing XML with JSON,JSON parsing,AJAX using JSON and jQuery

Practice:

1.Create an simple application using AJAX to show the table of numbers given by user at runtime.

2.Access web service using Ajax and handle using JSON

Module IV: Responsive Web Design (5 hrs)

Introduction
The Best Experience for All Users
• Desktop
• Tablet
• Mobile
Bootstrap
Overview of Bootstrap
Need to use Bootstrap
Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid
Typography
Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress

Bars,
Pagination, List Groups, Panels, Dropdowns, Collapse, Tabs/Pills, Navbar, Forms, Inputs
Bootstrap Grids, Grid System, Stacked/Horizontal
Bootstrap Themes, Templates

Practice:

1.Create a responsive website using bootstap

Module V: PHP(10 hrs)

PHP(10):

Introduction to PHP,Working with arrays,Functions,Forms,Handling date and Times,Working with Files,Session and state management,Database operations from PHP

Practice:

1.Develop student registration web appication using PHP

2.Write a PHP database application that collects comments from users and makes it possible for users to view all the comments that have been submitted. You will need three files: an HTML page with a form where the user can enter a comment; a PHP program to process the input from this form by adding the comment to the database; and a PHP program that displays all the comments.

Module VI: Introduction to Drupal(5 hrs)

Drupal Basics, Content Management System, Content Management Framework, Web Application,Framework,Drupal Workflow, Bootstrap, hooks, callbacks, output,Modules (Core and Contributed), Nodes, Blocks, Regions,The Admin Interface (Overview),Content Management, Site Building, Site Configuration, User Management, Reports, Help,ContentTranslation,User Contributed Modules,Layouts in Drupal, File Systems

Practice:

1.Setup Drupal server and develop a site on it

Module VII: XML & Web Security (6 hrs)

(6 hrs)

XML (2)

Introduction to XML,XML Validation,Reason for XML,XML Tree Structure, XML DOM,XML DTD,XML Schema

XML style language(2)

XML and XSLT, XML Parsing,XML parsers (DOM & SAX),XML WSDL,RSS Feed

Web Security(2)

SQL Injection,Cross-Site Scripting (XSS),Security standards (OWASP)

Practice:

1. Creating XML Document

2.DTD creation

3.Test SQL Injection for student resgistration application

Text/Reference Books

1.Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Black Book Kindle Edition,byKogent Learning Solutions Inc.

2.HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed Kindle Edition,by DT Editorial Services

3.Programming PHP: Creating Dynamic Web Pages, Third Edition, by Kevin Tatroe, O'REILLY

4.Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON kindle Edition by Lindsay Bassett,O'REILLY

5.Bootstrap: Responsive Web Development by Jake Spurlock, Paperback

Project Work

1.Online Quiz System

2.Online Student feedback System

3.. Online Tutorial System

4.Restaurant Billing System

5.Online MCQ Database Bank System

Java Technologies

Code	Course Title	Credit	T-P-PJ
CUTM1031	**Java Technologies**	4	2-1-1

Objective

- Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- Be aware of the important topics and principles of software development
- Have the ability to write a computer program to solve specified problems
- Have the ability to write a computer program to solve specified problems
- Be able to use the Java SDK environment to create, debug and run simple Java programs
- Acquire java coding skill
- It helps students in getting jobs in different IT firms

Course outcome

- Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs
- Read and make elementary modifications to Java programs that solve real-world problems
- Identify and fix defects the common safety issues in code
- Document a Java program using Javadoc
- Use a version control system to track source code in a project
- Qualify confidently any interview process where Java is the requirement

Course content

Module I: Introduction to Java (8 hrs)

Features and Installation, Java Programming Basics, Decision Making and Looping, Class and Object, Inheritance
Practice 1 (1 Hr)
Practice 2 (1 Hr)

Module II: Package and Safe Code (5 Hr)

Interfaces, Packages and Access Protection, Exception Handling (Fault Tolerant Programming)

Practice 3 (1 Hr)

Module III: Collection and Threads (5 Hr)

ArrayList, Vector, Set, Map, Multi-threaded Programming, Synchronization

Practice 4 (1 Hr)

Module IV: Language and Utility Packages (5 Hr)

String Handling, Wrappers, Runtime Memory Management, Cloning, Calendar, Date and Time Facilities, Scanner, Internationalization

Practice 5 (1 Hr)

Practice 6 (1 Hr)

Module V: Input/ Output and Applets (5 Hr)

Byte and Character Stream I/O, Persistence, Applet: Architecture, Skeleton, and Implementation

Practice 7 (1 Hr)

Practice 8 (1 Hr)

Module VI: GUI Programming (5 Hr)

AWT: Container, Components, Layout Managers, Event Handling

Practice 9 (1 Hr)

Practice 10 (1 Hr)

Module VII: Networking and Advanced (5 Hr)

Networking Fundamental, Client-Server Communication, Remote Method Invocation (RMI), Java Virtual Machine (JVM) Tuning, Java Profiler

Practice 11 (1 Hr)

Practice 12 (1 Hr)

Text Book(s):

1. Java The Complete Reference, Fifth Edition, C25 Herbert Schildt, McGraw-Hills

Reference Book(s):

1. Murach's Java Programming, 5th Edition, Joel Murach, Mike Murach & Associates, 2011, ISBN-78-1-943872-07-7

2. Introduction to Java Programming, Comprehensive, 10th ed., Y. Daniel Liang, 2014. ISBN-10: 0133813460, ISBN-13: 9780133813463

https://nqr.gov.in/qualification-title?nid=3002
https://www.cdac.in/index.aspx?id=DAC&courseid=0#
https://canvas.harvard.edu/courses/63117/assignments/syllabus
https://canvas.harvard.edu/courses/69911/assignments/syllabus
https://xid.harvard.edu/xid-apps/submitAccountForm.do

YouTube Resources:	freeCodeCamp.org
	Codearchery
	Edureka
	free project
	Jenkov

Online Source(s):

1. https://docs.oracle.com/javase/tutorial/java/index.html

2. https://www.programiz.com/java-programming

3. https://marcus-biel.com/

Software/Tool(s): Java 8, Eclipse IDE

Online Compiler: https://ideone.com/

Online Coding Practice: https://www.hackerrank.com/

List of Practices:

Practice 1 (Module-I)

Program-1:

Write a program that computes the standard deviation of a set of floating point numbers that the user enters. First the user says how many numbers N are to follow. Then the program asks for and reads in each floating point number. Finally it writes out the standard deviation. The standard deviation of a set of numbers Xi is:

SD = Math.sqrt(avgSquare - avg2)

Here, avg is the average of the N numbers, and avg2 is its square.

avgSquare is the average of Xi * Xi. In other words, this is the average of the squared value of each floating point number.

For example, if N = 4, say the numbers were:

Xi Xi * Xi

2.0 4.0

3.0 9.0

1.0 1.0

2.0 4.0

sum 8.0 18.0

Now:

avg = 8.0/4 = 2.0

avg2 = 4.0

avgSquare = 18.0/4 = 4.5

SD = Math.sqrt(4.5 - 4.0) = Math.sqrt(.5) = 0.7071067812

To do this you will need to do several things inside the loop body for each floating point value as it comes in: add it to a sum, square it and add it to a sum of squares. Then after the loop is finished apply the formula.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 2 (Module-I)

Program-1:

Better encapsulation of the Goods class would call making instance variables private and using getter and setter methods to access them. A further refinement would be to make the class abstract and to define additional child classes. Here is a revised Goods class:

```
public abstract class GoodsSGA
{
```

```
private String description;
private double price;
private int quantity;
public GoodsSGA( String des, double pr, int quant )
{description = des;
price = pr;
quantity = quant;}
double getPrice()
{return price;}
void setPrice( double newPrice)
{price = newPrice;}
int getQuantity()
{return quantity;}
void setQuantity ( int newQuantity )
{quantity = newQuantity;}
public String toString()
{return "item: " + description + " quantity: " + quantity + " price: " + price ;}
```

Revise the source code for the classes Food, Toy, and Book. (Perhaps call the revised classes FoodSG, ToySG, and BookSG.) create a new class ToiletrySG for things like bubble bath. Create a new testing class, StoreSG to test your revised classes.

Note: the child classes will need to use the getter and setter methods to access the instance variables that are declared as private in GoodsSG.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 3 (Module-II)

Program-1:

User-Friendly Division Practice:

Put in a loop so that the user is repeatedly asked for the numerator and the divisor. For each set of data, the program prints out the result, or an informative error message if there is a problem (division by zero or poor input data).

The program continues looping, even if there is a problem Exit the loop when data entered for the numerator start with characters "q" or "Q". Don't print out an error message in this case.

Don't ask for the divisor if the user just asked to quit.

Here is sample output from one run:

Enter the numerator: 12

Enter the divisor: 4

12 / 4 is 3

Enter the numerator: 12

Enter the divisor : 0

You can't divide 12 by 0

Enter the numerator: glarch

You entered bad data.

Please try again.

Enter the numerator: quit

You will need to use the method charAt() from the String class.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 4 (Module-III)

Program-1:

In mathematics, several operations are defined on sets. The union of two sets A and B is a set that contains all the elements that are in A together with all the elements that are in B. The intersection of A and B is the set that contains elements that are in both A and B. The difference of A and B is the set that contains all the elements of A except for those elements that are also in B.

Suppose that A and B are variables of type set in Java. The mathematical operations on A and B can be computed using methods from the Set interface. In particular:

A.addAll(B) computes the union of A and B; A.retainAll(B) computes the intersection of A and B; and A.removeAll(B) computes the difference of A and B. (These operations change the contents of the set A, while the mathematical operations create a new set without changing A, but that difference is not relevant to this exercise.)

For this exercise, you should write a program that can be used as a "set calculator" for simple operations on sets of non-negative integers. (Negative integers are not allowed.) A set of such integers will be represented as a list of integers, separated by commas and, optionally, spaces and enclosed in square brackets. For example: [1,2,3] or [17, 42, 9, 53,108]. The characters +, *, and - will be used for the union, intersection, and difference operations. The user of the program will type in lines of input containing two sets, separated by an operator. The program should perform the operation and print the resulting set.

Here are some examples:

Input	Output
[1, 2, 3] + [3, 5, 7]	[1, 2, 3, 5, 7]
[10,9,8,7] * [2,4,6,8]	[8]
[5, 10, 15, 20] - [0, 10, 20]	[5, 15]

To represent sets of non-negative integers, use sets of type TreeSet<Integer>. Read the user's input, create two TreeSets, and use the appropriate TreeSet method to perform the requested operation on the two sets. Your program should be able to read and process any number of lines of input. If a line contains a syntax error, your program should not crash. It should report the error and move on to the next line of input. (Note: To print out a Set, A, of Integers, you can just say System.out.println(A). We've chosen the syntax for sets to be the same as that used by the system for outputting a set.)

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 5 (Module-IV)

Program-1:

Password Checker:

Write a program that repeatedly asks the user for a proposed password until the user enters an acceptable password. When the user enters an acceptable password, the program writes a message and exits.

Acceptable passwords:

Are at least 7 characters long.

Contain both upper and lower case alphabetic characters. Contain at least 1 digit. The logic of this program can be quite tricky. Hint: use toUpperCase(), toLowerCase, and equals(). You will also need nested ifs.

Here is a run of the program:

Enter your password:

snowflake

That password is not acceptable.

Enter your password:

SnowFlake

That password is not acceptable.

Enter your password:

snowflake47

That password is not acceptable.

Enter your password:

Snowflake47

Acceptable password.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 6 (Module-IV)

Program-1:

Secret Code:

A text message has been encoded by replacing each character of the message with an integer. Each integer is an index into a key-phrase that contains all the lower case letters of the alphabet as well as the space character. The key-phrase may contain the same character in several locations. The encoded text is series of integers, like this:

35 10 10 33 9 24 3 17 41 8 3 20 51 16 38 44 47 32 33 10 19 38 35 28 49

To decode the message, look up each integer in the key-phrase and output the corresponding character. For example, say that the key-phrase is this (the index of each character has been written above it):

1111111111222222222233333333334444444444455

0123456789012345678901234567890123456789012345678901

six perfect quality black jewels amazed the governor

using each integer from the encoded text as an index into the phrase results in the decoded message:

attack the bridge at dawn

Write a program that decodes a secret message contained in a text file. The first line of the text file contains the key-phrase. Then the file contains a sequence of integers, each of which indexes the key-phrase. Find the character corresponding to each integer and output the secret message. Note if a character character such as 'e' occurs several places in the key-phrase it may be encoded as different integers in different parts of the secret message.

(The recipient of the secret message gets only the file of integers and must put the key-phrase at the top of the file.) For example, here is the contents of a secret message file ready for the program:

six perfect quality black jewels amazed the governor

35 10 10 33 9 24 3 17 41 8 3 20 51 16 38 44 47 32 33 10 19 38 35 28 49

Here is a sample run of the program:

C:\> java Decode < secretFile.txt

attack the bridge at dawn

You will need the charAt() method of String.

Here is another secret message file, with key-phrase inserted, that you can use to test your program:

six perfect quality black jewels amazed the governor

31 16 2 3 4 42 48 7 27 9 10 43 12 13 35 15 1 40 18 3

20 15 33 23 24 32 26 29 28 27 21 31 25 14 34 14 36

42 38 19 40 41 27 3 44 50 46 42 48 49 50 6

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 7 (Module-V)

Program-1:

Stop Word Remover:

Write a program that reads in a file of text, perhaps the text of a novel. The program copies the same text to an output file, except that all the useless words such as "the", "a", and "an" are removed. (Decide on what other words you with to remove. The list of words removed is called a stop list.) Do this by reading the text file token by token using hasNext() and next(), but only writing out tokens not on the stop list. Prompt the user for the names of the input and output files.

Fairly Easy: The output file will have only N tokens per line. Do this by counting tokens as you output them. N will be something like 10 or 12.

Improved Program: Preserve the line structure of the input file. Do this by reading each line using nextLine() and then creating a new Scanner for that line. (Look at the on-line documentation for Scanner.) With each line's Scanner, use hasNext() and next() to scan through its tokens.

Harder: Write out no more than N characters per line. N will be something like 50. Do this by keeping count of the number of characters written out per line. The length() method of String will be useful. If X characters has already been written to the current line, and if X plus the length of the current token exceeds N, then start a new line.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 8 (Module-V)

Program-1:

E-Mail Address Extractor:

Write a program that scans a text file for possible e-mail addresses. Addresses look like this:

someone@somewhere.net

Read tokens from the input file one by one using hasNext() and next(). With the default delimiters of Scanner, an entire e-mail address will be returned as one token. Examine each token using the indexOf() method of String. If a token contains an at sign @ followed some characters later by a period, regard it as a possible e-mail address and write it to the output file.

Programs such as this scan through web pages looking for e-mail addresses that become the targets of spam. Because of this, many web pages contain disguised e-mail addresses that can't easily be automatically extracted.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 9 (Module-VI)

Program-1:

User-friendly Fat Calculator, with Advice:

Further modify the calories from fat calculator so that it includes another TextField that will be set with the text "Too many fat calories" if the percentage of calories from fat is equal or greater than 30 percent, or to "Healthy amount of fat" if the percentage is less than that.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 10 (Module-VI)

Program-1:

Three Button Monte:

Write a program to implement a game:

There are three buttons in the frame. Two of the buttons cause the program to quit using System.exit(0); the remaining button changes the frame to green (a win!) The winning button is different each time the game is played.

The easy way to do this (although it seems unfair to the user) treats each button the same way. The actionPerformed() method does not check which button was clicked. When any button is clicked, the method picks a random integer from 0 to 2 and performs the "winning" action if the integer happens to be 0. Otherwise, it performs the "losing" action. To the user, it seems like there is a "winning" button and two "losing" buttons. But, in fact, it does not matter which button was clicked.

This is similar to some electronic gambling devices in casinos, where it appears to the user that there are "winning moves" and "losing moves" but in fact the machine actually ignores what the user has done and just declares a "win" every now and then, according to predetermined odds.

You will need the Random class:

```
Random randNum = new Random(); // create a Random number object
. . .
int someInt = randNum.nextInt(3); // someInt gets a number from 0 to 2
```

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 11 (Module-VII)

Content Delivery with Networking:

Write a Client-Server program where the client queries with a name of file and the server delivers the content of requested files to the client over the network.

(Improve the program by making the server multi-threaded)

Practice 12 (Module-VII)

Greet the user with Remote Method Invocation:

Write a program using RMI, where the user invokes a method on remote object with username as parameter and receives a greeting message based on time of the day along with username.

Projects

However, not limited to:

1. Chat application
2. Text Editor application
3. GUI based Scientific Calculator
4. Paint application
5. Slam book

(*PROJECT REVIEWS WILL COMMENCE BEYOND CLASS HOURS)

Monitoring:
Credit will be received only on making an honest effort. It is expected that students will finish watching all lecture video and complete all challenge problems by the end of each lecture week.
Borrowing code from other sources is allowed only with proper attribution and credit given to the original author(s).

List of Common Programs to solve using Java:

1.Program to calculate area of a triangle

2.Program to solve quadratic equation

3.Program to swap two zariables (with and without using third variable)

4.Program to generate random numbers in various ways

5.Program to convert miles to kilometers and vice-versa

6.Program to convert celsius to fahrenheit and vice-versa

7.Program to check if a number is odd or even

8.Program to check if input year is leap year

9.Program to test primality

10.Program to print all prime numbers in an interval using "Sieve of Eratosthenes"

11.Program to generate factorial of all elements in an array

12.Program to display the multiplication table up to 20

13.Program to print the fibonacci sequence

14.
Program to check armstrong number, perfect number, Harshad number

15.Program to generate armstrong numbers in an Interval

16.Program to find the sum of Harshad numbers in an interval

17.Program to display powers of two Using lambda

18.Program to perform conversions among decimal to binary, octal and hexadecimal

19.Program to display ASCII table

20.Program to find HCF/GCD and LCM

21.Program to find factors of given natural number

22.Program to make a simple calculator

23.Program to shuffle deck of cards

24.Program to generate fibonacci sequence using recursion

25.Program to find sum of natural numbers using recursion

26.Program to find factorial of number using recursion

27.Program to convert decimal to binary using recursion

28.Program to add two matrices

29.
Program to obtain transpose of a matrix

30.Program to multiply two matrices

31.Program to check if a string is palindrome

32.Program to remove punctuations from a string

33.Program to sort words lexicographically

34.Program to illustrate different set operations

35.Program to count frequency of each vowel in a string

36.Program to find hash value of a file

Note: 1 credit theory=12 hrs lecture, 1 credit practice/project=15 hrs lab/workshop/field work in a semester

This course on courseware: http://courseware.cutm.ac.in/courses/java-technologies/

Operating System Concepts

Code	Course Title	(Credit)	T-P-PJ
CUTM1032	**Operating System Concepts**	3	1-1-1

Objective

- To understand the services provided by and the design of an operating system.
- To understand the structure and organization of the file system.
- To understand what a process is and how processes are synchronized and scheduled.
- To understand different approaches to process management, management and resource management
- To understand the data structures and algorithms used to implement an OS.

Course outcome

- Demonstrate understanding of the concepts, structure and design of operating Systems
- Demonstrate understanding of operating system design and its impact on application system design and performance
- Demonstrate competence in recognizing and using operating system features.

Course content

Module I: Overview and System Call (4Hrs)

What is an OS, Computer System Organization, OS Design , Major OS Services System Call and Its Types, Concurrent Programming

Practice:

1.Copy from one file to another using unbuffered I/O system calls

2.Print contents of a directory using system calls to open and read directory contents

3.Infinite loop - kill by sending SIGINT signal through ctrl C or kill command

Module II: Linux Commands and Utilities (3 Hrs)

su,sudo, cat, man,ls,kill, ps, top, more, less, head, tail,cp,rm, info, passwd, chage, ssh(Remote Login), Changing Login Shell, grep ,lpr, lpq, sort, uniq, file, pipe(Communication between processes),echo, date, script, tar,bzip2,xz,gzip,which, whereis, locate, who ,w, free

Practice:

4.Gaining Root privilege with su and sudo

5.Archiving and compressing of files

6.Creating user , setting up password and managing password policies

7.Login to remote system through SSH

8.Monitoring Linux process using top and ps

9.Communication between process using pipe

Module III: File System (3 Hrs)

The Hierarchical Filesystem, Directory Files and Ordinary Files , The Working Directory ,Your Home Directory, Pathnames, Relative Pathnames ,Working with Directories , Access Permissions ,ACLs: Access Control Lists, Hard Links Symbolic Links

Practice:

10.Demonstration of Hierarchical File System in Linux

11.Controlling Access to Files with Linux File System Permission

12.Controlling Access to Files with ACL

Module IV: Vi Editor (3 Hrs)

Using vim to Create and Edit a File , Introduction to vim Features, Moving the Cursor, Input Mode, Deleting and Changing Text, Searching and Substituting, Copying, Moving, and Deleting Text, The General-Purpose Buffer

Practice:

13.Create , Edit, Save and Exit a File

14.Edit a System File with Vim by Using Maximum Editing Feature

Module V: Shell Scripting (4 Hrs)

About bash , Variables, Array, Operators, Decision Making, Loop Controls, Functions, I/O redirections

Practice:

15.Write a script to demonstrate decision making and loop control

16.Write a script to demonstrate array and function

17. Write a script to demonstrate I/O redirection

Module VI: Process and Thread Management (5 Hrs)

Process Concepts, Process State , Process Schedulers , Context switch , Inter-process Communication,Thread Overview, Multi-threading Models, Process Synchronization, Scheduling Concepts, Scheduling Criteria , Scheduling Algorithms (FCFS, SJF, PRIORITY,RR)

Practice:

18.Process creation using fork()

19.Write a program to simulate producer-consumer problem using semaphores

20.Write a multi-threaded program

21.Write a program to simulate the following CPU scheduling algorithms to find turnaround time and waiting time.

a) FCFS b) SJF c) Round Robin d) Priority

Module VII: Resource and Memory Management (5Hrs)

Deadlock Characterization , Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery,Swapping ,Contiguous Memory Allocation , Paging ,segmentation

Practice:

22.Write a program to simulate Bankers algorithm for the purpose of deadlock avoidance.

23.Write a program to simulate page replacement algorithms

a) FIFO b) LRU c) LFU

Text Books:

1.A Practical Guide to Linux Commands, Editors, and Shell Programming, by Mark G. Sobell, Prentice Hall, 2013

2.Operating System Concepts, by ABRAHAM SILBERSCHATZ , PETER BAER GALVIN, GREG GAGNE , Wiley, Ninth Edition, 2013

Sample Projects :

1.Making Your Own Linux Shell

2.Creating Linux Kernel Modules

3.The Dining Philosophers Problem

4.Adding Your Own System Call To Linux Kernel

5.Disk Scheduler

6.Library Management System Using Shell Scripting
Courseware Link : http://courseware.cutm.ac.in/courses/operating-system-concepts/

Database Creation and Maintenance

Code	Course Title	(Credit)	T-P-PJ
CUTM1033	Database Creation and Maintenance	4	2-1-1

Objective

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Make the students understand the principles behind relational database management systems, including the database environment, the relational model, relational languages, develop simple SQL queries using MySQL Workbench.
- Strong practice in SQL programming through a variety of database problems.
- Obtain skills in designing, developing and administrating the relational database

Course outcome

- Construct database models for different database applications.
- Construct the entity-relationship diagrams of different databases using MySQL Workbench.
- Convert the entity-relationship diagrams into relational tables.
- Design, implement and normalize a relational model for a given problem domain.
- Write SQL queries for performing database operations.
- Different functions to update or delete data from the databases.
- Create job opportunities in database design, development and administration.

Course content

Module I: Database System Concepts and Data Models (9 hrs)

Database Approach: Database, Approaches to Building a Database, Data Models, Database Management System (DBMS), DBMS Architecture, Building a DBMS, Components of DBMS Environment.

File Based Systems and Database Systems: File Based Approach, Database Systems, File-oriented Systems vs. Database Systems, Advantages and Disadvantages of DBMS.

Data Source: Types of Data sources, Access Documents and Information from Data Sources.

Roles in Database Environment: Database Users, Database Administrators (DBA)

Database installation procedure: Functionalities of MySQL Workbench, Creating a connection in MySQL Workbench, Setup the MySQL Workbench, Working of SQL Editor, Data Export & Import, Database table creation & insertion of values.

Practice:

1. Collect data from different data sources and identify the type of data source.
2. Installation of MySQL Workbench. Import and export the database into MySQL Workbench.
3. Create instances of database in MySQL.

Module II: Database System Architecture and ER Diagram (9 hrs)

Three Level Architecture, External Level, Conceptual Level, Internal Level, Schemas, Mappings, Instances, Data Independence, Data Abstraction, E/R Model - Conceptual data modeling - motivation, entities, entity types, various types of attributes, relationships, relationship types, E/R diagram notation, examples, Database Design & Modeling using MySQL Workbench, Create EER Diagram with MySQL Workbench.

Practice:

1. Analyze the problem and come with the entities in it. Identify what Data has to be persisted in the databases.
2. Use MySQL Workbench to construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.
3. Use MySQL Workbench to construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.
4. A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Use MySQL Workbench to construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.

Module III: Relational Data Model and Relational algebra (9 hrs)

Relational Data Model: Concept of relations, schema-instance distinction, keys, referential integrity and foreign keys.

Relational algebra operators: selection, projection, cross product, various types of joins, division, example queries, tuple relation calculus, domain relational calculus, converting the database specification in E/R notation to the relational schema.

Practice:

1. Converting an E-R diagram of a given student-section-course database to a relational schema.
2. Converting an E-R diagram of a given hospital database to a relational schema.

Module IV: Structured Query Language-DDL (7 hrs)

Database installation procedure: Database table creation & insertion of values.

Database Languages: SQL - DDL, DML, TCL, DCL

SQL: Data definition in SQL, table, key and foreign key definitions, update behaviors. Querying in SQL - basic select-from-where block and its semantics, nested queries, notion of aggregation, aggregation functions group by and having clauses.

Data Definition Language: Creating a Database, Table Operations (Create, Alter, Drop, Truncate, Comment and Rename), Creating and Removing an Index.

Practice:

1. Perform the following operations using MySQL Workbench:
   - Creating a Database
   - Viewing all databases
   - Viewing all Tables in a Database
   - Creating Tables (With and Without Constraints)
   - Inserting/Updating/Deleting Records in a Table
   - Saving (Commit) and Undoing (rollback)
2. Perform the following Altering a Table operations using MySQL Workbench:
   - Rename the table
   - Add a new column
   - Rename the column name
   - Change the data type of the column name
   - Delete table

Module V: Structured Query Language-DML, DCL, TCL (7 hrs)

Data Manipulation Language: Using different DML commands (Insert, Delete, Update, and Select), Sorting Results (Order By), Aggregate Functions, Join, Grouping Results (Group By)

Data Control Language & Transaction Control Language: Using different DCL commands (Grant, Revoke) & using different TCL commands (Commit, Rollback and Savepoint).
Practice:

1. For a given set of relation schemes, create tables and perform the following using MySQL Workbench:

    - Simple Queries with Aggregate functions
    - Queries with Aggregate functions (group by and having clause)
    - Queries involving- Date Functions, String Functions , Math Functions
2. For a given set of relation schemes, create tables and perform the following Join Queries using MySQL Workbench:
    - Inner Join
    - Outer Join
3. For a given set of relation schemes, create tables and perform the following Subqueries using MySQL Workbench:
    - With IN clause
    - With EXISTS clause
4. For a given set of relation tables perform the following using MySQL Workbench:
    - Creating Views (with and without check option), Selecting from a view
    - Dropping views

Module VI: Normalization (8 hrs)

Normalization: Dependencies and Normal forms - Importance of a good schema design, problems encountered with bad schema designs, motivation for normal forms, dependency theory - functional dependencies, Armstrong's axioms for FD's, closure of a set of FD's, minimal covers, Types of Normalization: 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them, algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, join dependencies and definition of 5NF.

Practice:

1. To check whether the given database table is normalized or not. If yes find out the status of normalization and reasoning.
    - First Normal Form
    - Third Normal Form
    - BCNF

Module VII: Transaction Management and Concurrency Control (5 hrs)

Transaction processing and Error recovery: Concepts of transaction processing, ACID properties, concurrency control, locking based protocols for concurrency control, error recovery and logging, undo, redo, undo-redo logging and recovery methods.

Project:

Mini project (Application Development using MySQL Workbench)

- Leave management system
- Blood Donation Management system
- CUTM Cafeteria System

- Student subject registration system
- CUTM stock monitoring system

(The above projects, including but not limited to, assign to the students)

Text Books:

2. *Fundamentals of Database System – Elmasari & Navathe - Pearson Education-5th, Edition.*
3. *Database System Concepts by Sudarshan, Korth (McGraw-Hill Education) - 6th, edition*

Reference Books:

1. *An introduction to Database System – Bipin Desai, Galgotia Publications*
2. *Database System: concept, Design & Application - S.K.Singh (Pearson Education)*

Note: 1 credit theory=12 hrs lecture, 1 credit practice/project=15 hrs lab/workshop/field work in a semester

Database Cluster Administration and Security

Code	Course Title	Credit	Prerequisite
CUTM1034	Database Cluster Administration and Security	**4**	2-1-1

Objective

- To teach students about managing access rules and configurations for computing clusters for internal and external clients.
- To provide knowledge on manage cluster deployment.
- To provide hands-on experience of monitor capacity, performance and database security.

Course outcome

- Students will gain knowledge about concepts of Construction of Clusters.
- Students will develop skill of developing new techniques in cluster deployment in real world Applications.
- Student will develop skill on database security establishment in different networks.

\

Course content

Module I: Construction of Clusters
Define the size of the cluster based on data to be stored, Identify quicker ways to add or remove nodes using cluster monitoring tools, Deploy and manage clusters, Setup cluster configurations using cookbooks.

Practice:
1. Different functions to update or delete data from the databases such as CRUD (Create Read Update Delete)
2. Organizational policies, procedures and guidelines which relate to maintaining existing databases.

Module II: Manage cluster deployment
Define encryption standards at rest and in motion for data stored on the cluster, create documentation on computing clusters, Validate computing clusters, Administer cluster access.

Practice:
3. Different databases, including: relational (MySQL, PostgreSQL),
4. NoSQL (Redis, Amazon DynamoDB)

Module III: Monitor capacity and performance
Evaluate cluster requirement estimates through capacity planning, Monitor cluster connectivity and performance and recommend efficiencies, Troubleshoot issues with cluster connectivity and performance with appropriate people.

Practice:
5. How to migrate a database to a new architecture
6. How to automate database maintenance

Module IV: Database Security
Secure a Database System applying Information Security, Security Threats, Security Vulnerabilities and Risk Management.

Practice:
7. How to store and retrieve information
8. How to provide the security in databases

Module V: Database backup
Implement Database System backup and restoring backup, Implement Database System security by creation of roles and functions for Database Users.

Practice:
9. Write a program for database backup maintenance.
10. Write a program for database recovery.

Module-VI: Message Transfer
Operations browsing, downloading messages, pictures from internet, sending and receiving Emails, sending attachments of pictures and text information, E-chatting.

Practice:
11. Write a program for message transfer between databases

Module-VII: Cluster Management
Master-Master, Master-Slave, Multimaster, Master-Multislave

Practice: Using MySQL, PostgreSQL
12. Write a program to configure master slave communication

Projects
1.Configuring master-slave server in MySQL without delay
2.Configuring master-slave server in MySQL with time constraint
3.Configuring single master and 2 slaves with one as RO and other as RW
4.Network Communication Optimization
5.Cluster File system and Parallel I/O
6.Cluster Scheduling

Textbooks:
1. Implementing database security and auditing, Author: Ron Ben-Natan
2. Database Security, Authors: Silvana Castano, Giancarlo Martella, M. Fugini.
3. Fundamentals of Database Systems, Author: Ramez Elmasri.

Note: 1 credit theory=12 hrs lecture, 1 credit practice/project=15 hrs lab/workshop/field work in a semester

Data Warehousing and Data Mining

Code	Course Title	(Credit)	T-P-PJ
CUTM1035	Data Warehousing and Data Mining	4	2-2-0

Objective

- To identify the scope and essentiality of Data Warehousing and Mining.
- To analyze data, choose relevant models and algorithms for respective applications.
- To study spatial and web data mining.
- To develop research interest towards advances in data mining.

Course outcome

- Understand Data Warehouse, Data Mining Principles
- Design data warehouse with dimensional modeling and apply OLAP operations.
- Identify appropriate data mining algorithms to solve real world problems
- Can access the data from different files like Excel, Word, SQL, PDF etc.
- Describe complex data types with respect to spatial and web mining
- Benefit the user experiences towards research and innovation integration

Course content

Module I: Data Mining Functionalities (09 hrs)

Data Mining Functionalities, Data Mining Task Primitives, Integration of a Data Mining System with a Database or a Data Warehouse System, Major issues in Data Mining. Data Pre-processing: Need for Pre-processing the data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

Practice:

1. Write a program to demonstrate Pre-processing program on Cancer.arff
2. Write a program to demonstrate Visualization on Crop-yield. Arff.

Module II: Data Warehouse (09 hrs)

Data warehousing; The need for Data Warehousing, the Building blocks of Data Warehouse, Data Warehouses and Data Marts, an overview of the components, metadata in the Data Warehouse, trends in Data Warehousing, Multidimensional Data Model, Data Warehousing Architecture, Data Warehouse Implementation, Further Development of Data Cube Technology, From Data Warehousing to Data Mining, Data Cube Computation and Data Generalization

Practice:

3. Write a program to analyze Weather.arff using Naive Bayesian algorithm.
4. Write a program to analyze Covid.arff using J-48.

Module III: OLAP Technology for Data Mining (07 hrs)

Data Warehouse and OLAP Technology, MOLAP, ROLAP, HOLAP, Difference between OLTP and OLAP.

Practice:

5. Do Analytical operations in 1) Roll-up 2) Drill-down 3) Slice 4) Dice and 5) Pivot.

Module IV: Mining Frequent Patterns, Associations and Correlations (07 hrs)

Mining Frequent Patterns, Associations and Correlations: Efficient and Scalable Frequent Itemset, Mining Methods, Mining various kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining.

Practice:

6. Write a program to demonstrate association rule mining using Apriori algorithm (Market-basket-analysis.arff).
7. Writing programs to access the data from PDF and SQL file.

Module V: Accessing data from different databases (07 hrs)

Accessing data from Excel file, Notepad file, Access file, Word file, SQL file, PDF file and Image file.

Practice:

8. Writing programs to access the data from Excel, Notepad, Access and Word file.

9. Writing programs to write program output into different files, appending columns (output/results) into an existing file using Python.
10. Write a program to demonstrate Pre-processing on Soil.arff

Module VI: Mining Streams, Time Series and Sequence Data (08 hrs)

Mining Streams, Time Series and Sequence Data: Mining Data Streams, Mining Time- Series, Data, Mining Sequence Patterns in Transactional Databases, Mining Sequence Patterns in Biological Data, Graph Mining, Social Network Analysis Multi Relational Data Mining and Spatial Data Mining.

Practice:

11. Write a program to demonstrate Pre-processing and Visualization on Student arff

Module VII: Mining Object, Spatial, Multimedia, Text and Web Data (07 hrs)

Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Spatial Data Mining, Multimedia Data Mining, Text Mining, Mining the World Wide Web, Applications and Trends in Data Mining.

Practice:

12. Write programs to analyze text dataset.

Text Books:

1. Data Mining Concepts and Techniques - Jiawei Han &MichelineKamber, Morgan Kaufmann Publishers, Elsevier,2nd Edition, 2006.
2. Introduction to Data Mining Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.

REFERENCE BOOKS:

1. Data Mining Techniques Arun K Pujari,2nd edition, Universities Press.
2. Data Warehousing in the Real World Sam Aanhory& Dennis Murray Pearson Edn Asia.
3. Insight into Data Mining, K.P.Soman, S.Diwakar,V.Ajay,PHI,2008.
4. Data Warehousing Fundamentals PaulrajPonnaiah Wiley student Edition

Android App Development

Code	Course Title	(Credit)	T-P-PJ
CUTM1036	**Android App Development**	6	2-2-2

Objective

- Introduction to the Android platform for Mobile Application Development.
- Understand Native Android Application, Android SDK features, Android Virtual Device (AVD), SDK manager, The Android Application Lifecycle.
- Understand Application Priority and Process state.
- Fundamental Android UI Design, Introduction Views, Creating Activity with UI to lunch the Activity.
- Explicitly Starting new Activities, Implicit Intent, and Runtime Binding
- Saving simple Application Data, creating and Saving Preferences, Retrieving Shared Preference.
- Introduction the Preference Activity and Preference Framework.
- Introduction Android Database, Introduction SQLite, and Content value working with SQLite Databases.
- Develop Android mobile Apps

Course outcome

- Individual after acquiring the knowledge of Android is able to Create Activities, Applications, Network-Based Application With Database Individual
- After Acquiring Skill To Create Files, Saving Files And Understanding Database Is Able To Manage Application With Database In order to implement the various process
- After Acquiring The Knowledge Intents Advance Skills To Understand Broadcast Receiver, Adapters And Internet Are Able To Create Network-Based Application.

Course content

Module-I: Introduction to Android (10 Hrs)-
Follow the concepts of Android; understand Features and Installation of Android Studio and Android Virtual Devices.

Practice -

- Installation of Android Studio
- Create Hello world Project

Module-II: Introduction to Android Activities and Layouts (10 Hrs)-
Create Applications; understand Activities and Layouts of Android, and the Activity Lifecycle.

Practice –

- Create Project by Implementing deferent Layouts
- Create an activity and implement the Activity Lifecycle

Module-III: Navigation and Data Passing (8 Hrs)-
Understand how data passing using Intent, Navigation between two Activity

Practice -

- Receive data from the user by Edit Text and pass the data to another activity using intent.

Module-IV: Broadcast Receiver & Content Provider (10 Hrs)-
Learn the use of Broadcast Receiver, Content Provider

Practice –

- Retrieve the device's battery info. And show in a project
- Use Broadcast Receiver & Content Provider in a Project

Module-V: List, Adapters, and Permission (12 Hrs)-
Android Permissions, List, and use of Adapter.

Practice –

- Retrieve data from a given URL and arrange them in a recycler view/ List View.

Module-VI: Create Files, Saving Files (12 Hrs)-
To Create Files, Saving Files in Android

Practice –

- Make one user input Form store the information in a separate Activity, Convert that Activity into a PDF format and store the PDF in device's internal storage.

Module-VII: Network Call (18 Hrs)-
Network call/ API call using Retrofit, OkHttp. Data (XML/JSON) Parsing &
Understand & Implement SQLite database, Firebase. Saving Data in the database.

Practice –

- Top 10 Downloaded App in IOS
- YouTube App using Google API
- Android Hybrid app development with flutter
- Android Hybrid app development using Ionic

- Android Hybrid app development using ReactJS

Text Book(s):

1. Head First Android Development Book by David Griffiths and Dawn Griffiths
2. http://yuliana.lecturer.pens.ac.id/Android/Buku/professional_android_4_application_development.pdf

Online Reference (s):

1. https://developer.android.com/guide
2. https://developer.android.com/docs
3. https://www.tutorialspoint.com/android/index.htm

Software Tool (s):

1. Android Studio

Mathematical Problem Solving

Code	Course Title	(Credit)	T-P-PJ
CUTM1037	Mathematical Problem Solving	4	2-2-0

Course Objectives

- To understand and analyze algorithms
- To understand efficiency of algorithms and alternative approaches
- To understand data structures and major algorithms and how they together play a role in efficiency
- To apply important algorithm design techniques using a programming language

Course Outcome:

- Ability to decide the appropriate data type and data structure for a given problem.
- Ability to select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- Ability to compare algorithms with respect to time and space complexity

Course Syllabus:

Module I: Introduction

Mathematics in Computer Science, Problem Solving and Algorithms, Data Structures and Algorithms, Algorithm Efficiency and Importance.

Module II: Problem Types

Sorting, Searching, String Processing, Graph and Numerical Problems

Module III: Algorithm Efficiency

Orders of Growth, Best-Case, Worst-Case and Average-Case Efficiencies, Analysis of Recursive and Non-Recursive Algorithms

Module IV: Brute Force and Exhaustive Search

Selection Sort and Bubble Sort, Sequential Sort and Brute-Force String Matching, Exhaustive Search, Depth-First Search and Breadth-First Search

Module V: Divide and Conquer

Merge Sort, Quick Sort, Binary Tree Traversal, Closest-Pair and Convex-Hull Problems

Module VI: Dynamic Programming and Greedy Technique

Dynamic programming- Floyd ‘s algorithm, Optimal Binary Search Trees, Knapsack Problem and Memory functions.

Greedy Technique- Prim’sAlgorithm, Kruskal’s Algorithm, Dijkstra’s Algorithm, Huffman Trees.

Module VII: Limitations of Algorithm Power

Lower-Bound Arguments, Decision Trees, P, NP and NP- Complete Problems.

Text Books:

1. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni and S. Rajasekharan
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin

Reference Books:

1. Design and Analysis of Algorithms: S. Sridhar
2. Design and Analysis of Algorithms: P. H. Dave, H. B. Dave
3. Algorithm Design: Foundations, Analysis and Internet Examples: M. T. Goodrich and R. Tomassia, John Wiley and sons
4. Algorithms Illuminated (Part 2): Graph Algorithms and Data Structures:Tim Roughgarden

Formal Languages Automata Theory

Code	Course Title	(Credit)	T-P-PJ
CUTM1038	Formal Languages Automata Theory	3	2-1-0

Objective

- This course covers the theoretical computer science areas of formallanguages and automata, computability and complexity. Topics covered include: regular and context-free languages; finite automata and pushdown automata; Turing machines; computability - halting problem, solvable and unsolvable problems.
- Study of the "lexical analyzer" of a typical compiler, that is, the compiler component that breaks the input text into logical units, such as identifiers, keywords, and punctuation; Software for scanning large bodies of text, such as collections of Web pages, to find occurrences of words, phrases, or other patterns.

Course outcome

- Acquire a full understanding and mentality of Automata Theory as the basis of all computer science
  languages design
- Have a clear understanding of the Automata theory concepts such as RE's, DFA's, NFA's, Stacks,
  Turing machines, and Grammars
- Be able to design FAs, NFAs, Grammars, languages modeling, small compilers basics
- Be able to design sample automata
- Be able to design parsers

Course content

Module-1: Finite Automata (5Hrs)
Introduction to Finite Automata; The central concepts of Automata theory; Finite Automata: NFA (with and without ε), DFA, FA conversions cycle, Reduced DFA
Practice

1.Finite Automata Design
2.Construction of NFA
3.Construction of DFA
4.Construction of DFA from RE
5.Construction of NFA from RE
6.Conversion of NFA with epsilon to NFA without epsilon
7.FA Conversion Cycle
8.Construction of Regular Expressions

VIRTUAL LAB

Module-2: Regular languages(5Hrs)

Regular languages; Closure properties of regular languages, Decision properties of regular languages, Minimization of automata, Grammar – Types of Grammar - CFG and Languages– Derivations and Languages, Ambiguity Relationship between derivation and derivation trees.

Practice

1.Minimization of automata using JFLAP
2.Identification of grammar using JFLAP
3.Language generation using JFALP
4.Derivation tree generation using JFLAP
5.Left Most and Right Most Derivation
6.Minimization of CFG using JFLAP
7.Verification of Ambiguous grammar
8.Chomsky grammar verification using JFLAP

Module-3: Context-Free Languages(5Hrs)

Context-Free Languages: Simplification of CFG, Elimination of Useless symbols Unit productions - Null productions, Chomsky normal form, Problems related to CNF, CYK Algorithm.

Practice

1.Elimination of Useless symbols using JFLAP
2.Unit productions using JFLAP
3.Null productions using JFLAP
4.Chomsky normal formal using JFLAP
5.CFG to CNF using JFLAP
6.CFG to GNF using JFLAP
7.CYK algorithm implementation using JFLAP

Module-4: Pushdown Automata (5Hrs)

Pushdown Automata- Definitions – Moves, Instantaneous descriptions, Deterministic pushdown automata, Equivalence of Pushdown automata and CFL, pumping lemma for CFL

Practice

1.Construction of PDA using JFLAP
2.Construction of DPDA using JFLAP
3.Construction of NPDA using JFLAP
4.Construction of PDA from CFG

Module-5: Linear bounded Automata (5Hrs)

Linear bounded Automata, Context Sensitive Language, Chomsky Hierarchy of Languages and automata, Organization of Linear bounded Automata, Properties of context-sensitive languages.

Practice

1.Combine two automata using JFLAP
2.Minimization of FA using JFLAP
3.Identification of CFL using JFLAP

4.Implementation of LBA
5.Identification of Type-1 grammar using JFLAP
6. Check the properties of CFL using JFLAP

Module-6: Turing Machines(5Hrs)
Turing Machines- Formal definition of Turing machines, Instantaneous descriptions, Turing Machine as Acceptors Languages and functions, Turing Machine constructions.

Practice
1.Construction of One-Tape Turing Machine using JFLAP
2.Construction of Turing Machine for for language L = {0n1n2n | n≥1}
3.Construct a Turing Machine for a language L = {aibjck | ik} ∩ {aibjck | i>j>k or i>j>k}
4.Construction of Multi-Tape Turing Machine

Module-7: Decidable and undecidable problems and Compilers Basics(6Hrs)
Undecidability- Basic definitions, Decidable and undecidable problems, Properties of Recursive and Recursively enumerable languages, Phases of the compiler, Lex, Parsing Techniques

Practice
1. Design a lexical analyzer for a given language and the lexical analyzer should ignore redundant spaces, tabs, and newlines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value
2.Write a LEX program to identify whether a given line is a comment or not.
3.Write a LEX program to recognize strings under 'a', 'a*b+', 'abb'
4.Implementation of SRP using LEX program
5. Write a LEX program for constructing of LL (1) parsing.
6.Lex Program to Count the Number of Lines words, and Characters in the Input File
7.Lex program that distinguishes keywords, integers, floats, identifiers, operators, and comments.
8. Generate regular expressions for the language accepting all combinations of a's, over the set ∑ = {a}
9. Generate regular expression for the language accepting all combinations of a's except the null string, over the set ∑ = {a}
10. Generate regular expression for the language accepting all the string containing any number of a's and b's.
11.Lex program to count the number of words, small and capital letters, digits and special characters in a C file
12.Lex Program for infix to postfix conversion
13.Implementation of LL(1) Parser
14.Implementation of RD Parser
15.Implementation of LR parser

Textbooks
Automata and Computability, Dexter C. Kozen, Springer Publisher.
Introduction to Automata Theory, Languages and Computation, Hopcroft, Motwani, and Ullman,

Pearson Publishers, Third Edition.
Principles of Compiler Design: M. Ganga Durga, T. G. Manikumar MJP Publisher, 06-Jun-2019.

References

Elements of the Theory of Computation, H. R. Lewis and C.H. Papadimitriou, Prentice Hall Publishers.
Introduction to Languages and the Theory of Computation, John. C. Martin, Tata McGraw-Hill.

Note: 1 credit theory=12 hrs lecture, 1 credit practice/project=15 hrs lab/workshop/field work in a semester