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

Learning 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

Learning 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.

Learning 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

Learning 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

OOPs with C++ Programming

Code	Course Title	Credit	T-P-PJ
CUTM1028	OOPs with C++ Programming	4	1-2-1

Objective

- To understand how C++ improves C with object-oriented features
- To learn how to design C++ classes for code reuse
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism
- To learn how to use exception handling in C++ programs
- Obtain skill to understand the concept of classes and objects, inheritance, polymorphism.To develop art of writing efficient programs

Learning outcome

- Apply the object-oriented programming approach in connection with C++
- Illustrate the process of data file manipulations using C++
- Apply virtual and pure virtual function & complex programming situations
- Write an error free program of minimum 200 lines of code
- Acquire C++ coding skill which helps students in getting jobs in different IT firms

Course content

Module I: Revision of C programming (8 hrs)

Revision of C Programming, Pointers, Functions (Call by value and reference), Recursion, Arrays using Pointers, Structures, Union, Enumeration and Typedef, File handling.

Programs:

1. Write a Program to perform Parameter passing.
2. Write a program to create a scientific calculator.
3. Write a program to convert a decimal to binary number using recursion.
4. Write a program to Read 'n' employee details and display the top 10 employees as per the salary.
5. Write a program to evaluate MCQ questions of an examination and generate the results using files.

Module II: Basics of Object oriented concepts (8 hrs)

Object oriented concepts Classes and Objects, Encapsulation, Abstraction, Overloading, Inheritance, Polymorphism.

Beginning with C++, Tokens, Static Members, Constant Members, Expressions, Control Structure, Functions: parameter passing, inline function, function overloading.

Programs:

1.Write a program to read a number and check whether the number is Prime number , Palindrome number , Magic number , Armstrong number , Strong number or not.

2. Write definitions for two versions of an overloaded function. This function's 1st version sum() takes an argument, int array, and returns the sum of all the elements of the passed array. The 2nd

version of sum() takes two arguments, an int array and a character ('E' or 'O'). If the passed character is 'E', it returns the sum of even elements of the passed array and is the passed character is 'O', it returns the sum of odd elements. In case of any other character, it returns 0 (zero).

Module III: Class-Object-Constructor (10 hrs)

Classes: data members, member function, array of objects, static data members, constant members function, and friend function.

Constructors, Encapsulating into an object, Destructors.

Programs:

1. Define a class to represent a book in a library. Include the following members:

Data Members

Book Number, Book Name, Author, Publisher, Price, No. of copies issued, No. of copies

Member Functions

(i) To assign initial values

(ii) To issue a book after checking for its availability

(iii) To return a book

(iv) To display book information.

2. A bank maintains two kinds of accounts for customers, one called as savings and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

Define a class to represent a bank account. Include the following members: Data members: 1. Name of the depositor. 2. Account number. 3. Type of account. 4. Balance amount in the account. Member functions: 1. To assign initial values. 2. To deposit an amount. 3. To withdraw an amount after checking the balance. 4. To display the name and balance. Write a main program to test the program

3. Declare a class to represent fixed-deposit account of 10 customers with the following data members:

Name of the depositor, Account Number, Time Period (1 or 3 or 5 years), Amount.

The class also contains following member functions:

(a) To initialize data members.

(b) For withdrawal of money (after alf of the time period has passed).

(c) To display the data members.

4. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

Module IV: Inheritance (8 hrs)

Associations, Inner Classes, Memory Management and pointers

Inheritance: Derived classes, member accessibility, forms of inheritance, virtual base classes.

Programs:

1. Write a Program to describe about all types of inheritance.

2. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.
3. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure. The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required.

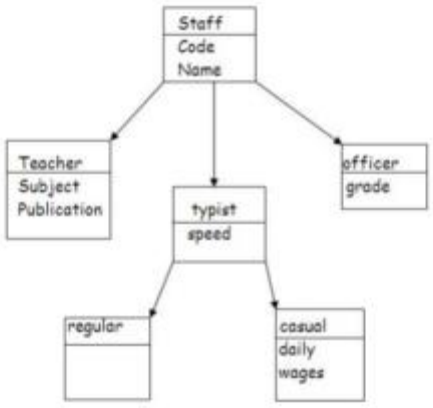

fig: class relationships (for exercise 8.3)

Module V: Polymorphism (8 hrs)
Polymorphism (Compile time Polymorphism, Run time Polymorphism), Virtual Functions, Abstract class, virtual destructors, Interfaces.
Programs:
1. Write a Program to overload ++ operator.
2. Write a program to overload + operator by concatenating strings.
3. Write a program to describe about virtual function.

Module VI: Exception Handling (8 hrs)
Exception Handling, Managing Console I/O Operations, Streams & Files: streams, hierarchy of stream classes, working with files

Programs:
1. Write a Program to describe about exception handling mechanism.
2. Write a Program to describe multi catch statement.
3. Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below.

Name	Code	Cost
Turbo C++	1001	250.95
C primer	905	95.70
.....	.....	.....
.....	.....	.....

Note that the name and code are left-justified and the cost is right justified with a precision of two digits. Trailing zeros are shown.

4. Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.
5. Write a program that reads character from the keyboard one by one. All lower case characters get store inside the file LOWER, all upper case characters get stored inside the file UPPER and all other characters get stored inside OTHERS.

Module VII: Templates (8 hrs)

Advance Topics in C++ Object Design and Templates STL (Standard Type Libraries)RTTI (Run Time Type Identification) Advanced Typecasting ,new data types, new operators, class implementation, namespace scope , operator keywords, new headers , C++ Containers

Programs:

1. Write a function template for finding the minimum value contained in an array.
2. Imagine a publishing company that markets both books and audio-cassette versions of its works. Create a class called Publication that stores the title (a string) and price of a publication. From this class derive two classes: Book, which adds a page count (type int); and Tape, which adds a playing time in minutes (type float). Each of the three class should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data. Write a main() program that creates an array of pointers to Publication. In a loop, ask the user for data about a particular book or Tape, and use new to create a object of type Book or Tape to hold the data. Put the pointer to the object in the data for all books and tapes, display the resulting data for all the books and taps entered, using a for loop and a single statement such as pubarr[i]->putdata();to display the data from each object in the array.

Text Books:

1. E Balagurusamy, “Object Oriented Programming with C++”, Tata McGraw Hill, Sixth Edition.
2. Herbert Schlitz, “The Compete Reference C++”, Tata McGraw Hill, Fourth Edition.

Reference Books:

1. Ashok Kamthane, “Object Oriented Programming with ANSI and Turbo C++”, Pearson.
2. Behrouz A. Forouzan & Richard F. Gilberg “A Structured approach using C++” Cengage Learning Indian Edition.

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

Data Structures using C++

Code	Course Title	Credit	T-P-PJ
CUTM1029	Data Structures using C++	4	1-2-1

Objective

- Be familiar with techniques of algorithm analysis and Recursive method
- Be familiar with implementation of linked data structures such as linked lists and binary trees
- Be familiar with several sub-quadratic sorting algorithms including quick sort, merge sort and heap sort
- Be familiar with some graph algorithms such as shortest path and minimum spanning tree
- Skill to choose appropriate data structure as applied to specified problem definition.Ability to analyze algorithms and algorithm correctness

Learning outcome

- Evaluate algorithms and data structures in terms of time and memory complexity of basic operations
- Define basic static and dynamic data structures and relevant standard algorithms for them: stack, queue, dynamically linked lists, trees, graphs, heap, priority queue, hash tables, sorting algorithms, min-max algorithm
- 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: Problem Solving Analysis (6 hrs)

Define the problem, Identify the problem, Introduction to Problem Solving, Problem solving basics, Defining creativity v/s innovation

Find Creative Solutions using creativity tools

Effective problem solving approaches , Critical thinking and information analysis , Brainstorming, Reverse Brainstorming, Imagineering, Mind Mapping, Six Thinking Hats: A Tool to Strengthen Critical Thinking, Collaboration, Communication, and Creativity Skills , Analyzing the situation, Gathering information, Identifying solution criteria , Decision Making Methods , Charts and Diagrams , Applying outcome-based thinking

Evaluate and Select solution

Pro's and Con's, Force field analysis, Feasibility/Capability Analysis, Decision analysis, evaluating problems , Choosing among alternatives , Qualitative analysis, discussing qualitative analysis techniques , Establishing objectives , Assigning weight to objectives in order to make the best decision , Creating a satisfaction scale to choose between alternatives

Implementing Decisions

Create an action plan, Break solution into action steps, Prioritize actions and assign roles (setting priorities for taking action) ,Follow-up at milestones

Programs:

1. Problem solving (Control structures, Arrays) using Raptor Tool.

Module II: Array & Stack (9 hrs)

Analysis of different Algorithms, Asymptotic analysis, Algorithm analysis, Complexity Analysis, Application of Data structures

Basic Data Structures, Arrays, Stacks and its applications (Recursion, Infix to Postfix Conversion and Postfix Evolution

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

2. Write a program to perform the following menu driven program on the STACK.

a. Push
b. Pop
c. Display
d. Exit

Module III: Queue & Linked List (9 hrs)

Queues, Priority Queues, Dequees.

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

Programs:

1. Write a program to perform the following menu driven program on the Queue.

a. Insertion
b. Deletion
c. Display
d. Exit

2. Write a program to create a single linked list performs 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 Using Linked List (8 hrs)

Circular linked list and Double linked list, Stack implementation using Linked List and Queue implementation using Linked List

Programs:

1. Write a program to create a Double linked list performs 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 and display it.
3. Write a program to implement Stack Using Linked List.
4. Write a program to implement Queue Using Linked List.

Module V: Trees **(10 hrs)**

Trees and hierarchical orders ,Introduction to trees , 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 **(8 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

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: Hashing **(8 hrs)**

Hash functions and hash tables ,Hashing & Introduction to hash tables ,Hash functions , Mapping down to 0 ... M – 1 , Chained hash tables , Scatter tables , Open addressing , Linear probing , Quadratic probing , Double hashing, Poisson distribution , Collision Resolution
Graph Terminology and Traversals.

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 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.

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

Learning 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

Learning 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):
1111111111222222222233333333334444444444 55
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/

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

Learning 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:

1. *Fundamentals of Database System – Elmasari & Navathe - Pearson Education-5th, Edition.*
2. *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

ndroid 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

Learning 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