Centurion University of Technology and Management, Odisha

Course Structure & Syllabus

Master of Computer Application

(MCA)

2-Years Programme

School of Engineering & Technology

2022

Index

Programme Objectives; Job/Higher studies/Entrepreneurship

POs: Graduates will be able to;

PO	**Outcomes**
PO1	Engineering knowledge: Apply knowledge of mathematics, science, engineering fundamentals, and Computer Application to the solution of engineering problems
PO2	Problem analysis: Identify, formulate, review literature and analyze Computer Appication problems to design, conduct experiments, analyze data and interpret data
PO3	Design /development of solutions: Design solution for Computer Application problems and design system component of processes that meet the desired needs with appropriate consideration for the public health and safety, and the cultural, societal and the environmental considerations
PO4	Conduct investigations of complex problems: Use research based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions in Computer Application
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to Computer Appication activities with an understanding of the limitations
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to Computer Appication practice
PO7	Environment and sustainability: Understand the impact of the Computer Application solutions in societal and environmental contexts, and demonstrate the knowledge and need for sustainable development
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the Computer Application practice
PO9	Individual and team work: Function affectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings in Computer Application
PO10	Communication: Communicate effectively on complex engineering activities with the engineering committee and with society at large, such as, being able to comprehend and write affective reports and design documentation, make effective presentations in Computer Application

PO11	Project Management and finance: Demonstrate knowledge & understanding of the Computer Application principles and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments in Computer Application
PO12	Life- long learning: Recognize the need for, and the preparation and ability to engage in independent research and lifelong learning in the broadest contest of technological changes in Computer Application

PEOs/PSOs

PEO1: Prepare students to build competency in current technology and its application to meet the industry need for skilled Engineer

PEO2: Provide students with strong foundational concepts and also domain knowledge to pursue research to build solutions or systems of varying complexity to solve the problems identified

PEO3: Enable graduates to innovate, bring new idea and become an entrepreneur

PSO1. Graduate will be able work on high-end technology at IT Services industries.

PSO2. Graduate can acquire industry certified level of competency and work on real time IT application projects viz; Health/Agriculture/Security/Data Management etc.

PSO3. Graduate can start its own IT service company to provide technical solution Course Outcomes Attributes

Course Outcomes	**Attributes**
CO1	Knowledge
CO2	Analytical skill and Critical Thinking
CO3	Problem Solving and Decision taking ability
CO4	Use of Tool, Design and Development (Hands-on/Technical skill)
CO5	Research
CO6	Environment and Sustainability
CO7	Ethics & Team work
CO8	Soft skill

Course Structure

Master of Computer Application

SEMESTER - I				
Sl No	**Course Code**	**Name of the Course**	**Credits**	**T+P+J**
1	CUTM1024	IT Infrastructure Management	6	2+2+2
2	CUTM1602	Programming in C	4	2+1+1
3	CUTM1018	Data Analysis and Visualization using Python	4	0+1+3
4	CUTM1605	Software Engineering	3	2+0+1
5	CUTM1059	Database Management Systems	3	2+1+0
6	CUTM1016	Job Readiness	6	0+6+0
		Total	**26**	
SEMESTER – II				
1	CUTM1606	Computer Networks	4	2+1+1
2	CUTM1025	Cloud Practitioner	2	0+2+0
3	CUTM1031	Java Technologies	4	2+1+1
4	CUTM1603	Data Structures	4	1+2+1
5	CUTM1030	Advanced Web Programming	4	1+2+1
6	CUTM1032	Operating System Concepts	3	1+1+1
		Total	**21**	
SEMESTER – III is CBCS based (Domains)				
		Domain: Software Technology		
	CUST2010	Web Services Using JAVA	5	0+4+1
1	CUST2011	Advanced JAVA Programming	4	0+3+1
	CUST2012	Web Programming Using React	3	0+2+1
	CUST2013	Product Development	8	0+0+8
2	CUTM1036	Android App Development	6	2+2+2
		Data Science and Machine		
	CUML2010	**Learning**		
	CUML2011	ML for Predictive Analysis	4	(0+2+2)
	CUML2007	ML for Image Analytics	4	(0+2+2)
3	CUML2009	Digital video Processing	3	(0+2+1)
	CUML2008	Mathematics for ML	3	(2+1+0)
	CUML2005	IoT Analytics	4	(0+2+2)
	CUML2006	Internship	4	(0+0+4)
		Project	4	(0+0+4)
		Cloud Technologies		
	CUCT2030	AWS Solution Architect (SAA-		
4	CUCT2031	CO2).	6	(2-4-0)
	CUCT2032	AWS Developer (DVA-CO1)	6	(2-4-0)
		Project	6	(0-0-6)

5		**Gaming and Immersive Learning (AR-VR)**		
	CUAR2060	Introduction to Gaming & Simulation	2	(1 + 1+ 0)
	CUAR2061	Game Assets & Game Objects	3	(1 + 1 + 1)
	CUAR2062	Building Game Environment	3	(1 + 1 + 1)
	CUAR2063	Game Animation, Scripting & UI	3	(1 + 1 + 1)
	CUAR2064	Binary Deployment and Cross-Platform Controls	3	(1 + 1 + 1)
	CUAR2065	Project	6	(0 + 0 + 6)
		Total	**20**	
SEMESTER – IV is CBCS based				
2	CUTM1027	Information Security	3	2+1+0
3	CUTM1019	Machine Learning using Python	4	1+2+1
6	CUTM1037	Mathematical Problem Solving	4	2+2+0
7	CUTM2474	DotNet Programming	4	0+2+2
8	CUTM2475	Node JS	2	0+2+0
9	CUTM2476	Cloud Security Foundation	2	0+2+0
10	CUTM2477	Cloud Security Associate	2	0+2+0
		Total	**33 (required 23)**	
			96	

Semester - I

IT Infrastructure Management

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

Course Objectives

- 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
- Able to get job as network administrator

Course Outcomes

COs	**Course Outcomes**	**Mapping COs with POs (High-3, Medium-2, Low-1)**
CO1	Able to gain knowledge on installation of Windows and Linux both in Physical and VM	PO1 (3)
CO2	Able to identify and resolve the hardware and software problem	PO2(3), PO5(1)
CO3	Acquire hands-on skill in hardware administration	PO3 (2)

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

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

Programming in C

Course Title	Code	Type of course	T-P-PJ	Prerequisite
Programming in C	CUTM 1602	TPP	1-2-1	Nil

Course Objectives

- 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 Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Will gain Knowledge on programming	PO1 (3), PO12(1)
CO2	Analyse the problems for implementation	PO2 (3)
CO3	The student will able to write 150 to 200 line programs without any error to solve any problems	PO3(3), PO5(2)
CO4	Design and develop several applications with hands-on	PO3 (3)

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 Analysis and Visualization Using Python

Code	Course Title	Credit	T-P-PJ
CUTM1018	**Data Analysis and Visualization Using Python**	4	0-1-3

Course Objectives

- How to tell a story from data
- How to marshal the data for storyline
- Get skill to quickly and easily draw plot or visualize the information through visualization technique
- The ability to develop visualization to tell the story
- Get skill on analysis of data using visualization as a tool

Course Outcome

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain knowledge on visualization with good story line and perform job of a data analyst	PO1 (3)
CO2	Able to analyze and visualize the dataset	PO2(3)
CO3	Ability to design dashboard	PO3 (3), PO5(1)

COURSE CONTENT

Module-I

STORY BOARD DEVELOPMENT
The objective and flow of the story to be understood through cases

Module-II

DATA READING USING PYTHON FUNCTIONS
Python libraries: Pandas, NumPy, Plotly, Matplotlib, Seaborn, Dash
Data collection from online data sources, Web scrap, and data formats such as HTML, CSV, MS Excel, data compilation, arranging and reading data, data munging

Module-III

DATA VISUALSATION USING PYTHON LIBRARIES

Different graphs such as Scatterplot, Line chart, Histogram, Bar chart, Bubble chart, Heatmaps etc.

Dashboard Basics – Layout, Reporting, Infographics, Interactive components, live updating

Projects List

1. COVID 19
2. World Development Indicators
3. ERP dashboarding
4. Details of Social/ Empowerment schemes of Govt. etc.

References:

- https://www.programmer-books.com/wp-content/uploads/2019/04/Python-for-Data-Analysis-2nd-Edition.pdf
- https://towardsdatascience.com/data-visualization/home

Reading materials and videos available on internet on how to use ANACONDA, JUPYTER NOTEBOOK and Python Libraries

Software Engineering

Course Title	Course Code	Credit	T-P-PJ
Software Engineering	CUTM1605	3	2-0-1

Course Objectives:

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- To learn software development process that include process models, user requirements, specifications, software design, coding, software testing, maintenance, documentation, software process/product metrics, risk management, quality management.
- Describe and compare various software development methods and understand the context in which each approach might be applicable.
- Expand students' familiarity with mainstream languages used to model and analyze object designs (e.g., UML).
- Introduce state-of-the-art tools and techniques for large-scale software systems development.
- Implement the major software development methods in practical projects.

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	gain a comprehensive understanding of the different phases of the software development life cycle	PO1 (3)
CO2	develop skills in gathering, analyzing, and specifying software requirements	PO2(3)
CO3	gain knowledge of project management methodologies, including agile and waterfall approaches	PO3 (3), PO5(1)

Course Syllabus:

Module I - Introduction to Software Engineering and Process Models

The evolving role of software, changing nature of software, software myths, software development challenges.

Generic view of process: process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.

Process Models and Software Development Life Cycle Phases, Traditional Life Cycle Models: Waterfall Model, ‘V’ Model, Prototyping Model, the Iterative Waterfall Model, the Spiral Model, Evolutionary model, Component-Based Software Engineering (CBSE). Rapid software development models: Agile methods, Extreme programming, Rapid application development (RAD).

Module II - Software Requirement Analysis and Specification

Software Requirement Engineering: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirement Engineering Process: Requirement Inception, Identification of Stakeholders, Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

System models: Context models, behavioral models, data models, object models, structured methods.

Module III – Object Oriented Analysis and Design

Introduction to OOAD with OO Basics – Unified Process – UML diagrams – Use Case –Case study – the Next Gen POS system, Inception -Use case Modelling – Relating Use cases – include, extend and generalization – When to use Use-cases. Class Diagram— Elaboration – Domain Model – Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies – Aggregation and Composition – Relationship between sequence diagrams and use cases – When to use Class Diagrams.

Module IV – Dynamic OO Modeling

Dynamic Diagrams – UML interaction diagrams – System sequence diagram – Collaboration diagram – When to use Communication Diagrams – State machine diagram and Modelling – When to use State Diagrams – Activity diagram – When to use activity diagrams Implementation Diagrams – UML package diagram – When to use package diagrams – Component and Deployment Diagrams – When to use Component and Deployment diagrams.

Module V - Software Testing

Structured Design Methodologies: Coupling and Cohesion, Software Testing: Testing Fundamentals, Verification & Validation, Black Box Testing, White Box Testing, Unit Testing, Integration Testing, Object Oriented Testing, System Testing, Alpha and beta testing, Regression testing.

Module VI - Software Quality Assurance

Software Quality Concepts, Software Quality Metrics and its Classification, Software Size Metrics: LOC Metrics, Function Point Metrics, Process point. Software Configuration Management (CM), Software Quality Assurance (SQA), Software Quality and Agile Methods, Automated and Manual Functional Testing: Acceptance testing, Mock objects, User interface testing (HTTPUnit, Canoo), Performance testing. Software Metrics and Analytics, Quality and Process Standards and Guidelines

Module VII - Risk Management in Software Engineering Projects

Risk management: Reactive vs. Proactive Software Risk, Risk Identification, Risk projection, Risk refinement, Risk mitigation, monitoring & management - The RMMM Managing people.

Text Books:

1. Roger S. Pressman, "Software Engineering: A practitioner's Approach", 7th edition, McGraw-Hill International edition
2. Ian Sommerville, "Software Engineering", 10th edition, Pearson Education
3. Rajib Mall, "Fundamentals of Software Engineering", 4th edition, Prentice Hall India Learning Private Limited.
4. Grady Booch. Object-oriented analysis and design with applications / Grady Booch. -. 2nd ed. ISBN 0-8053-5340-2.

Reference Books:

1. Grady Booch, James Rambaugh, Ivar Jacobson, "The unified modeling language user guide", Pearson Education.
2. James F. Peters, WitoldPedrycz, John Wiley, "Software Engineering, an Engineering approach"
3. Waman S Jawadekar, "Software Engineering principles and practice", The McGraw-Hill Companies.

Database Management Systems

Course Title	Course Code	Credit	T-P-PJ
Database Management Systems	CUTM1059	3	2-1-0

Objectives

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database Modeling, relational, hierarchical, and network models
- To understand and use data manipulation language to query, update, and manage a database
- To develop an understanding of essential Properties of DBMS concepts such as: database security, integrity, concurrency
- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Learning Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Acquire the knowledge on different database implementation for various applications	PO1 (3), PO12(1)
CO2	Solving different problems such as to convert the entity-relationship diagrams into relational tables, normalization and SQL queries	PO2(3)
CO3	Design database models and the entity-relationship diagrams of different databases using MySQL Workbench	PO3 (3)

Course content

Module-1: DBMS Concepts [5 Hrs]

Data Abstraction - Data models and data independence. Instances and Schemas. Components of a DBMS and overall structure of a DBMS- Life Cycle of a DBMS application- Database terminology.

Module-2: Data Modeling [5Hrs]

Basic concepts- Types of data models- Conceptual, physical and logical database models- E-R data model and Object-oriented data model. Components of ER Model- ER Modeling symbols. Entity and entity sets- Relations and relationship sets- E-R Diagrams- Reducing E-R Diagrams into tables.

Practice

Assume we have the following application that models soccer teams, the games they play, and the players in each team. In the design, we want to capture the following:

- We have a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs.
- Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses.
- Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team.
- For each match we need to keep track of the following:
- The date on which the game is played
- The final result of the match
- The players participated in the match. For each player, how many goals he scored, whether or not he took yellow card, and whether or not he took red card.
- During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place.
- Each match has exactly three referees. For each referee we have an ID (unique identifier), name, DoB, years of experience. One referee is the main referee and the other two are assistant referee.

Design an ER diagram to capture the above requirements. State any assumptions you have that affects your design (use the back of the page if needed). Make sure cardinalities and primary keys are clear.

Module-3: Relational DBMS Model [5 Hrs]

Basic concepts, Attributes and domains- Intention and extensions of a relation- concept of integrity and referential constraints- Relational Query Languages (Relational algebra and relational calculus (Tuple and domain relational calculus).

Module-4: Relational Database Design [6 Hrs]

Notion of normalized relations- Normalization using Functional Dependency- First Normal form- Second Normal Form- Third Normal form- BCNF.

Practice

Perform NF on the given table

CLICK HERE FOR TABLE

Module-5: SQL [6 Hrs]

Structure of a SQL query- DDL and DML, TCL- SQL queries and sub queries- Tables, views and indexes.

Practice

To study DDL-create and DML-insert commands.

(i) Create tables according to the following definition.

CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5) ,CNAME VARCHAR2(18) , BNAME VARCHAR2(18) , AMOUNT NUMBER(8,2) ,ADATE DATE);

CREATE TABLE BRANCH(BNAME VARCHAR2(18),CITY VARCHAR2(18)); CREATE TABLE CUSTOMERS(CNAME VARCHAR2(19) ,CITY VARCHAR2(18));

CREATE TABLE BORROW(LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME

VARCHAR2(18), AMOUNT NUMBER (8,2));

(ii) Insert the data as shown below.

DEPOSIT

CLICK HERE FOR TABLE

BRANCH

CLICK HERE FOR TABLE

CUSTOMERS

CLICK HERE FOR TABLE

BORROW

CLICK HERE FOR TABLE

(1) Describe deposit, branch.

(2) Describe borrow, customers.

(3) List all data from table DEPOSIT.

(4) List all data from table BORROW.

(5) List all data from table CUSTOMERS.

(6) List all data from table BRANCH.

(7) Give account no and amount of depositors.

(8) Give name of depositors having amount greater than 4000.

(9) Give name of customers who opened account after date '1-12-96'.

Module-6:Aggregate functions [4 Hrs]

Set Operations, predicates and joins, Set Membership- Tuple variables- Set comparison- Database modifications using SQL.

Practice

Create the below given table and insert the data accordingly.

Create Table Job (job_id, job_title, min_sal, max_sal)

COLUMN NAME DATA TYPE

job_id Varchar2(15)

job_title Varchar2(30)

min_sal Number(7,2)

max_sal Number(7,2)

Create table Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no)

COLUMN NAME DATA TYPE

emp_no Number(3)

emp_name Varchar2(30)

emp_sal Number(8,2)

emp_comm Number(6,1)

dept_no Number(3)

Create table deposit(a_no,cname,bname,amount,a_date).

COLUMN NAME DATA TYPE

a_no Varchar2(5)

cname Varchar2(15)

bname Varchar2(10)

amount Number(7,2)

a_date Date

Create table borrow(loanno,cname,bname,amount).

COLUMN NAME DATA TYPE

loanno Varchar2(5)

cname Varchar2(15)
bname Varchar2(10)
amount Varchar2(7,2)
Insert following values in the table Employee.
emp_n emp_name emp_sal emp_comm dept _no
101 Smith 800 20
102 Snehal 1600 300 25
103 Adama 1100 0 20
104 Aman 3000 15
105 Anita 5000 50,000 10
106 Sneha 2450 24,500 10
107 Anamika 2975 30
Insert following values in the table job.
CLICK HERE FOR TABLE
Insert following values in the table deposit.
CLICK HERE FOR TABLE
Perform following queries
(1) Retrieve all data from employee, jobs and deposit.
(2) Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.
(3) Display all jobs with minimum salary is greater than 4000.
(4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.
(5) Display employee no,name and department details of those employee whose department lies in(10,20)
To study various options of LIKE predicate
(1) Display all employee whose name start with 'A' and third character is ' 'a'.
(2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.
(3) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5 character long.
(4) Display the null values of employee and also employee name's third character should be 'a'.
(5) What will be output if you are giving LIKE predicate as '%_%' ESCAPE '\'
To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.
(1) List total deposit from deposit.
(2) List total loan from karolbagh branch
(3) Give maximum loan from branch vrce.
(4) Count total number of customers
(5) Count total number of customer's cities.
(6) Create table supplier from employee with all the columns.
(7) Create table sup1 from employee with first two columns.
(8) Create table sup2 from employee with no data
(9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.
(10) Delete all the rows from sup1.
(11) Delete the detail of supplier whose sup_no is 103.
(12) Rename the table sup2.

(13) Destroy table sup1 with all the data.
(14) Update the value dept_no to 10 where second character of emp. name is 'm'.
(15) Update the value of employee name whose employee number is 103.
To study Single-row functions.
(1) Write a query to display the current date. Label the column Date
(2) For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary
(3) Modify your query no 4.(2) to add a column that subtracts the old salary from the new salary. Label the column Increase
(4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.
(5) Write a query that produces the following for each employee:
earns monthly
(6) Display the name, hire date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.
(7) Display the hiredate of emp in a format that appears as Seventh of June 1994 12:00:00 AM.
(8) Write a query to calculate the annual compensation of all employees (sal+comm.).
Displaying data from Multiple Tables (join)
(1) Give details of customers ANIL.
(2) Give name of customer who are borrowers and depositors and having living city nagpur
(3) Give city as their city name of customers having same living branch.
(4) Write a query to display the last name, department number, and department name for all employees.
(5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output
(6) Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.
(7) Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.
(8) Create a query to display the name and hire date of any employee hired after employee SCOTT.

Module-7: Transaction Management [8 Hrs]

Subqueries,Manupulating Data,Transaction management and Concurrency control

Practice

To apply the concept of Aggregating Data using Group functions.
(1) List total deposit of customer having account date after 1-jan-96.
(2) List total deposit of customers living in city Nagpur.
(3) List maximum deposit of customers living in bombay.
(4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.
(5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.
(6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998
(7) Find the average salaries for each department without displaying the respective department numbers.

(8) Write a query to display the total salary being paid to each job title, within each department.
(9) Find the average salaries > 2000 for each department without displaying the respective department numbers.
(10) Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary.
(11) List the branches having sum of deposit more than 5000 and located in city bombay.
To solve queries using the concept of sub query.
(1) Write a query to display the last name and hire date of any employee in the same department as SCOTT. Exclude SCOTT
(2) Give name of customers who are depositors having same branch city of mr. sunil.
(3) Give deposit details and loan details of customer in same city where pramod is living.
(4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.
(5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000
(6) Display the last name and salary of every employee who reports to ford.
(7) Display the department number, name, and job for every employee in the Accounting department.
(8) List the name of branch having highest number of depositors.
(9) Give the name of cities where in which the maximum numbers of branches are located.
(10) Give name of customers living in same city where maximum depositors are located.
Manipulating Data
(1) Give 10% interest to all depositors.
(2) Give 10% interest to all depositors having branch vrce
(3) Give 10% interest to all depositors living in n
agpur and having branch city bombay.
(4) Write a query which changes the department number of all employees with empno 7788’s job to employee 7844’current department number.
(5) Transfer 10 Rs from account of anil to sunil if both are having same branch.
(6) Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.
(7) Delete depositors of branches having number of customers between 1 to 3.
(8) Delete deposit of vijay.
(9) Delete borrower of branches having average loan less than 1000.
To apply the concept of security and privileges.
To study Transaction control commands
VIRTUAL LAB

TEXT BOOKS

Database Management Systems: Raghu Ramakrishnan
ORACLE PL/SQL Programming – Scott Urman BPB Publications.

REFERENCES

Database Systems Concepts – Henry F Korth, Abraham Silberschatz.
Database Management Systems – Alexis Leon, Mathews Leon – Leon, Vikas Publications

Job Readiness

Code	Course Title	Credit	T-P-PJ
CUTM1016	**Job Readiness**	6	0-6-0

Course Objectives

To develop additional skills (verbal, logical, quantitative and reasoning) required to enhance employability as well as the entrepreneurial ability of the students

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Ability to enhance listening, speaking, reading and writing skills of the students	PO1(3),
CO2	Students must achieve the scores in IELTS 6.5	PO2(2)
CO3	Increase the Problem solving and Decision taking ability in students.	PO2(1)
CO4	It can enable ethical principles among students and they will commit to professional ethics and responsibilities.	PO8(2)
CO5	It can improve students' soft skills to communicate while working in a team. Also it helps to engage in independent research and lifelong learning in the technology context.	PO9(2) , PO10(2), PO11(1) , PO12(2)

Note: A student will be awarded the credits and grades as outlined in the attached presentation: https://drive.google.com/file/d/1Wst-jdAJuHHVtYC4F-p3SKuw1PHWOI1U/view?usp=sharing

COURSE SYLLABUS

Course Division

Course I: IELTS - Reading, Listening, Speaking, and Writing

Course II: IELTS Verbal

Course III: Quantitative Aptitude

Course IV: Logical Reasoning

Course I: IELTS - Reading, Listening, Speaking and Writing

Module I: IELTS Reading (18hrs)

- Skimming and Scanning
- Sentence Completion
- Choose the Correct options (A, B, C, D)
- Locating the Specific Information
- Assessment on Reading Skill

Module II: IELTS Listening (6hrs)

- Notes/ Form/Table completion
- Label the Map/Passage, Multiple Choice Questions
- Complete the Sentences, Listening to Find Information
- Assessment on Listening Skills

Module III: IELTS Speaking (18hrs)

- Speaking about yourself, your family, your work and your interests
- Introduction & Interview
- Topic Discussion (e.g, Environment, Covid 19, Job)
- Assessment on Speaking Skills

Module IV: IELTS Writing (6 hrs)

- Summarising the chart, table or graph
- Comparing and contrasting graphs and tables
- Describing maps & diagrams
- Agreeing & disagreeing
- Expressing a personal view & opinion
- Assessment on Writing Skill
- CV Writing (2nd year)
- Letter Writing
- Email Writing(2nd year)
- Getting Started –writing an introduction

Course II: IELTS Verbal

Module I: Grammar (4 Hrs)

- Articles
- Prepositions
- Subject-Verb
- Spotting Errors
- Sentence Correction

Module II: Vocabulary (5 Hrs)

- Synonyms
- Antonyms

- Contextual Vocabulary

Module III: Reading Comprehension (3 Hrs)

- Paragraph/ Sentence Completion
- Jumbled Sentences/ Jumbled Paragraph
- Reading Comprehension

Module IV: Verbal Analogies (3 Hrs)

Course III: Quantitative Aptitude

Module I: Number System & Operation (14 Hrs)

- Speed Math-1 : Multiplication tricks, Square, cube, square root, Cube root tricks
- Speed Math-2 : Speed Calculations
- Number System-01 : Operation on Numbers, Classification of Numbers, Tests of Divisibility, Unit Digit Calculation
- Number System-02 : Arithmetic Progression, Geometric Progression, Factors & Factorials, Trailing Zeroes, Remainder Theorem
- HCF & LCM : Concepts, short tricks, question discussion
- Average : Concepts, short tricks, question discussion
- Assessments

Module II: Basic Arithmetic (16 Hrs)

- Percentage-01 : Basics of Percentage, Effective percentage, shortcuts
- Percentage-02 : Advanced questions and discussions
- Profit & Loss-01 : Basics and advanced questions of Profit & Loss and shortcuts
- Profit & Loss-02 : MRP, Discount, Successive discount
- Ratio & Proportion : Types of ratios, Basics & Advanced Question
- Age : Concepts & Shortcuts
- Partnership : Concepts & Shortcuts
- Mixture &Alligation : Rule of Alligation, Basics & Advanced question, Short tricks
- Assessments

Module III: Time & Analysis (17 Hrs)

- Time, Speed, Distance : Concepts, Problems based on relations, Average speed, Stoppage time
- Trains : Relative Speed & All types of train problems
- Boats & Streams : Basics, Upstream, Downstream & Shortcuts
- Race : All concepts & Shortcuts
- Time & Work : Efficiency, wages, alternative day, chain rule
- Pipes & Cistern : Positive & Negative work
- Simple Interest : Concepts & Shortcuts on Simple Interest &Installments
- Compound Interest : Concepts & Shortcuts on Simple Interest &Installments
- Logarithm : All Formulae, concepts & Shortcuts
- Assessments

Module IV: Advanced Arithmetic (16 Hrs)

- Equation : Linear & Quadratic
- Permutation : All concepts & Shortcuts on factorial, fundamental principles of counting
- Combination : All concepts & Shortcuts on Selection (Groups/teams)

- Probability : Terms related to Probability, Event, Theorems related Probability, Conditional Probability. Shortcuts on coins, dices, balls, cards, etc
- Data Interpretation : (Bar/Pi-Chart /Line) graph
- Mensuration : Area & Volume
- Height & Distance : Lines of Sight, Horizontal line, Angle of Elevation, Angle of Depression
- Assessments

Course IV: Logical Reasoning

Module I: Verbal Reasoning-I (14 Hrs)

- Series-1 : Number series (Missing & Wrong)
- Series-2 : Letter, Alpha numeric, Miscellaneous series
- Coding & Decoding : Letter Coding, Number coding, Message coding, Substitution coding, Conditional coding
- Word Problem : Analogy, Odd man out, word formation, letter pair
- Logical Thinking : Brain Riddles
- Assessments

Module II : Verbal Reasoning-II (14 Hrs)

- Order & Ranking : Ranking & Sequence
- Direction Sense Test : Shortest Distance, Angular movement concept and Dusk & Dawn
- Clock : Concepts of Angle, Reflex angle, Right angle Opposite, Coincide and Incorrect clock
- Calendar : All concepts & Shortcuts
- Blood Relation : Jumbled-up descriptions, coded relations, Relation Puzzles
- Assessments

Module III : Non Verbal Reasoning (14 Hrs)

- Cubes & Dices
- Cubes & Cuboids
- Embedded Figure& Figure series
- Figure Puzzle & Figure grouping
- Figure Counting
- Mirror & Water Image
- Paper Cutting & Paper folding
- Assessments

Module IV: Advanced Reasoning (16 Hrs)

- Sitting Arrangement : Circular, Square, Rectangular, Linear, Triangular
- Puzzle : Box, Floor, Month, Day
- Advanced Puzzle : 3 variable
- Logical Venn Diagram
- Syllogism
- Statement & Conclusion
- Data Sufficiency
- Assessments

Semester - II

Computer Networks

Course Title	Code	Credit	T-P-PJ
Computer Networks	CUTM1606	4	2-1-1

Course Objectives

- Build an understanding of concepts of computer networking.
- Familiarize the student with the taxonomy and terminology of the computer networking area.
- Preparing the student for appearing in different competitive exam

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	able to grasp the basic concepts of computer networks, including the different types of networks, network topologies, protocols, and the OSI	PO1 (3)
CO2	able to design and evaluate network architectures for different scenarios, considering factors like scalability, reliability, security, and performance	PO2(3)
CO3	gain hands-on experience in configuring network devices such as routers, switches, and firewalls	PO3 (3), PO5(1)

COURSE OUTLINE

Module-I (5 Hrs):

Introduction to Computer Networks: Introduction to Networks, Components of Computer Networks, Network Topologies, Types of Computer Network.

Reference models: TCP/IP Protocol Suite: The OSI Model

Practice:

Practice 1: Navigating the IOS: Basic Connections, Accessing the CLI and Exploring Help, Exploring EXEC Modes, Setting the Clock

Practice 2: Investigating the TCP/IP and OSI models in action: Examine HTTP web traffic, Display elements of the TCP/IP protocol suite

Module-II (8 Hrs):

Physical Layer: Guided transmission media, wireless transmission media, Transmission Modes, line coding, multiplexing.

Module-III (10 Hrs):

Data Link Layer: Error Detection and Error Correction, sliding window protocol, High Level Data Link Control.

Multi Access Protocols: ALOHA, CSMA: CSMA/CD, CSMA/CA, Ethernet, bridges, repeaters, hubs, bridges, switches, routers and gateways.

Practice:

Practice 3: Implement Basic Connectivity: Perform a Basic Configuration on S1 and S2, Configure the PCs, and Configure the Switch Management Interface

Practice 4: Connecting a wired and wireless LAN: Connect to the Cloud, Connect Router0, Connect Remaining Devices and Verify Connections, Examine the Physical Topology

Module-IV (10 Hrs):

Network Layer: Internetworking, Network Layer Design issues, store and forward packet switching connection less and connection oriented.

Network Address: IPv4, IPv6, CIDR, Protocol: ICMP, ARP, RARP, DHCP.

Networks-routing algorithms: optimality principle, shortest path, flooding, Distance Vector Routing, Control to Infinity Problem, Hierarchical Routing, Congestion control algorithms.

Practice:

Practice 5: Configuring IPv6 Addressing: Configure IPv6 Addressing on the Router, Configure IPv6 Addressing on Servers, Configure IPv6 Addressing on clients, Test and Verify Network Connectivity

Practice 6: Verifying IPv4 and IPv6 Addressing: Complete the Addressing Table Documentation, Test Connectivity Using Ping, Discover the Path by Tracing the Route

Module-V (10 Hrs):

Transport Layer: Introduction to TCP, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Sliding Window, The TCP Congestion Control.

Practice:

Practice 7: TCP and UDP Communications: Generate Network Traffic in Simulation Mode; Examine the Functionality of the TCP and UDP Protocols

Module-VI (6 Hrs):

Application Layer: Introduction, Client server model, HTTP, FTP, electronic mail, TELNET, DNS,WWW

Practice:

Practice 8: Web and Email Servers: Configure and Verify Web Services, Configure and Verify Email Services

Practice 9: DHCP and DNS Servers: Configure Static IPv4 Addressing, Configure and Verify DNS Records

Practice 10: FTP Servers: Configure FTP Services on Servers, Upload a File to the FTP Server, and Download a File from the FTP Server

MODULE-VII (6 Hrs):

Network security and cryptography: Introduction, Symmetric key cryptography: Traditional cipher, block cipher, Public key cryptography: RSA

Text Books

- Data Communications and Networking - Behrouz Forouzan, Fifth Edition TMH, 2013.
- Computer Networks - Andrew S Tanenbaum, 4th Edition, Pearson Education

References

- An Engineering Approach to Computer Networks - S. Keshav, 2nd Edition, Pearson Edication.
- Understanding communications and Networks, 3rd Edition, W. Shay, Cengage Learning.
- Introduction to Computer Networks and Cyber Security, Chwan-Hwa (John) Wu, David Irwin, CRC Press.
- Computer Networks, L. L. Peterson and B. S. Davie, 4th edition, ELSEVIER

Cloud Practitioner

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

Course Objectives

- 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 Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain knowledge on AWS cloud values and Implement several policies using its services	PO1 (3)
CO2	Acquire skills to analyze and manage billing and pricing used for the resources	PO5(3)
CO3	Able to design and deployment of different applications using its services	PO3 (2)
CO4	Leads to the next level of preparation i.e. Associate and Professional level	PO8(1), PO9(2)

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

Java Technologies

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

Course Objectives

- 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 Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Will gain Knowledge on object oriented java programming	PO1 (3), PO12(2)
CO2	Identify and fix defects the common safety issues in code	PO2 (3)
CO3	Read and make elementary modifications to Java programs that solve real-world problems	PO3(3), PO5(2)
CO4	Design and develop several applications with hands-on	PO3 (3)

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):

111111111122222222223333333333444444444455

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:

- Program to calculate area of a triangle
- Program to solve quadratic equation
- Program to swap two zariables (with and without using third variable)

- Program to generate random numbers in various ways
- Program to convert miles to kilometers and vice-versa
- Program to convert celsius to fahrenheit and vice-versa
- Program to check if a number is odd or even
- Program to check if input year is leap year
- Program to test primality
- Program to print all prime numbers in an interval using "Sieve of Eratosthenes"
- Program to generate factorial of all elements in an array
- Program to display the multiplication table up to 20
- Program to print the fibonacci sequence
- Program to check armstrong number, perfect number, Harshad number
- Program to generate armstrong numbers in an Interval
- Program to find the sum of Harshad numbers in an interval
- Program to display powers of two Using lambda
- Program to perform conversions among decimal to binary, octal and hexadecimal
- Program to display ASCII table
- Program to find HCF/GCD and LCM
- Program to find factors of given natural number
- Program to make a simple calculator
- Program to shuffle deck of cards
- Program to generate fibonacci sequence using recursion
- Program to find sum of natural numbers using recursion
- Program to find factorial of number using recursion
- Program to convert decimal to binary using recursion
- Program to add two matrices
- Program to obtain transpose of a matrix
- Program to multiply two matrices
- Program to check if a string is palindrome
- Program to remove punctuations from a string
- Program to sort words lexicographically
- Program to illustrate different set operations
- Program to count frequency of each vowel in a string
- 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/

Data Structures

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

Course Objectives

- 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
- Able to get jobs in different IT firms as developer

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain the knowledge to evaluate algorithms and data structures in terms of time and memory complexity of basic operation steps	PO1 (3)
CO2	Ability to formulate programs that cover basic static and dynamic data structures, and relevant algorithms	PO2 (3)
CO3	Able to solve different problems such as search and sorting algorithms, traversal techniques, etc.	PO2 (3), PO3(3)
CO4	Design and demonstrate bugs in program, recognize needed basic operations with hands-on	PO3(3), PO5(2)

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

Course Objectives

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

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain the knowledge for designing and developing web applications	PO1 (3)
CO2	Apply PHP7 to improve accessibility of a web document	PO2 (3)
CO3	Develop a static, interactive and well-formed webpage using JavaScript, CSS3 and HTML5	PO3(3), PO5(1)

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)

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

Operating System Concepts

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

Course Objectives

- 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 Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Will gain Knowledge on understanding of the concepts, structure and design of operating Systems	PO1 (3)
CO3	Demonstrate competence in recognizing and using operating system features such as semaphore, deadlock and scheduling problems	PO2(3)
CO4	Demonstrate understanding of operating system design and its impact on application system design and performance	PO3 (2)

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/

Semester - III

Software Technology

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUST2010	Software Technology	0-9-11	• Java Technologies • Advanced Web Programming

		DOMAIN		
CUTM	**STCU2010**	**Software Technology**	**20**	0+9+11
	CUST2010	Web Services Using JAVA	5	0+4+1
	CUST2011	Advanced JAVA Programming	4	0+3+1
	CUST2012	Web Programming Using AngularJS	3	0+2+1
	CUST2013	Product Development	8	0+0+8

Course Objectives

- Develop knowledge-based force to serve the IT industry with the latest technologies.
- To explore methods of capturing, specifying, visualizing and analyzing software requirements.
- To learn and explore Spring Framework, Android, AngularJS, GIT.
- Able to get jobs in software industry

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain knowledge on design and implementation of various software applications	PO1 (3), PO12(3)
CO2	Develop skills to analyze, design, and prepare SRS	PO2(3)
CO3	Ability to solve the real world problems	PO2(3)
CO4	To design and develop web applications using Spring, AngularJS and android applications	PO3 (2), PO5(2)
CO5	Able to do research for publishing articles.	PO4 (1)

1. CUST2011:Advanced JAVA Programming (0-3-1) (75 hrs)

1.1 Understanding Web Architecture
1.2 Installation and configure java environment.
1.3 Understanding MVC Architecture
1.4 Create GIT repository for version control and teamwork
1.5 Create a web project and define a landing page.
1.6 Create controller using servlet in web project
1.7 Retrieve value from page in controller through the request parameter.
1.8 Establish Connection with database
1.9 Perform CRUD operation
1.10 Present data in JSP page using scriptlet, expression and action tag.
1.11 Declare method and variable in JSP page.
1.12 Transferring control from one resource to another using RequestDispatcher and ResponseRedirect
1.13 Set and get values in ServletContex and ServletConfig parameter.
1.14 Set and get values in application, session and request attribute.
1.15 Perform session tracking.
1.16 Upload file to the server.
1.17 Working with JSTL

2. CUST2010 : Web Services Using JAVA (0-4-1) (60 hrs)

2.1 Setting Spring environment
2.2 Understanding Maven and define POM
2.3 Create project using Spring and Understanding Spring Architecture
2.4 Setting bean in IOC container and understand Dependency Injection
2.5 Working with Spring AOP
2.6 Access data using JdbcTemplate; CRUD operation
2.7 Create Spring Boot projects using Spring Initializr
2.8 Implement DevTools for rapid application development
2.9 Implement application logs using application.properties
2.10 Implement Global Exception handling mechanism
2.11 Implement Security using Spring Boot
2.12 Configure Email in application.properties
2.13 Implement Testing using Spring Boot Test
2.14 Implement application monitoring using Actuator
2.15 Create Spring MVC project and define controller
2.16 Implement form handling using SpringMVC
2.17 Implement RESTful Web Service using GET Method
2.18 Implement RESTful Web Service using POST, PUT, DELETE and validations
2.19 Implement End to End testing using Spring Boot Testing - @AutoConfigureMockMvc
2.20 Handle cyclic dependency during JSON creation in Web Service
2.21 Understanding ORM and Hibernate
2.22 Hibernate Configuration using XML and annotation

2.23 Implement DML using Spring Data JPA on a single database table
2.24 Implement Query Methods feature of Spring Data JPA
2.24 Implement O/R Mapping using Spring Data JPA
2.25 Implement Hibernate Query Language (HQL) and Native Query
2.26 Explain the need and benefit of Criteria Query

CUST2012: Web Programming Using React (0-2-1) (45 hrs)

3.1 Configure the environment (install node and react), what is React? React version history, Create React app and debug

Templating using JSX

3.2 Working with React. createElement, Expressions, using logical operators, specifying attributes, Specifying children,

It's all about components

3.3 Significance of component architecture, Types of components - Functional, Class based, Pure, Component Composition

Working with state and props

3.4 What is state and its significance, read state and set state, passing data to component using props
3.5 Validating props using propTypes, supplying default values to props using defaultProps

Rendering lists

3.6 Using React key prop, using map function to iterate on arrays to generate elements, Event handling in React
3.7 Understanding React event system, Understanding Synthetic event, Passing arguments to event handlers

Understanding Component Lifecycle & Working with Forms

3.8 Controlled components, Uncontrolled components, Understand the significance to defaultValue prop, using react ref prop to get access to DOM element

Routing with React Router

3.9 Setting up react router, understand routing in single page applications, working with BrowserRouter and HashRouter components, configuring route with Route component, Using Switch component to define routing rules, making routes dynamic using route params
3.10 Working with nested routes, navigating to pages using Link and NavLink component, Redirect routes using RedirectComponent, Using Prompt component to get consent of user for navigation, Path less Route to handle failed matches

Just Redux

3.11 What is redux, why redux, Redux principles, Install and setup redux, Creating actions, reducer and store

Immutable.js for immutable data structures

3.12 What is Immutable.js? Immutable collections, Lists, Maps, Sets

React Redux

Redux saga(Redux middleware)

3.13 What is React Redux, Why React Redux, Install and setup, Presentational vs Container components
3.14 Understand high order component, Understanding mapStateToProps and mapDispatchtToProps usage
3.15 Why redux middleware, Available redux middleware choices, What is redux saga, Install and setup redux saga

3.16 Working with Saga helpers, Sagas vs promises

4. CUST2013 :Product Development (0-0-8) (45 hrs)

Text Books:

1. *Craig Walls, Spring in Action, Fifth Edition*
2. *Keogh Jim, J2EE: The Complete Reference*
3. *Ken Williamson, Learning AngularJS: A Guide to AngularJS Development*

Reference Books:

1. *Kogent Learning, Java Server Programming Java Ee7 (J2EE 1.7), Black Book*
2. *Alef Arendsen, Professional Java Development with the Spring Framework*
3. *Adam Freeman, Pro AngularJS*

http://courseware.cutm.ac.in/courses/software_technology/

Cloud Technology

		DOMAIN		
CUTM	**CTCU2030**	**Cloud Technology**	**18**	4--8--6
	CUCT2030	AWS Solution Architect (SAA-CO2)	6	2+4+0
	CUCT2031	AWS Developer (DVA-CO1)	6	2+4+0
	CUCT2032	Project	6	0+0+6

Course Objectives

- To learn architectural principles and services of AWS
- Able to develop scalable and secure Cloud applications
- Get skills to use resources in Cloud
- Able to get jobs in various industries

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain knowledge on architectural principles and services of AWS	PO1 (3)
CO2	Develop skills to analyze, design, and prepare resources in Cloud	PO2(3)
CO3	Ability to solve the real world problems	PO2(3)
CO4	Able to develop scalable and secure Cloud applications	PO3 (2), PO5(2)

AWS Solution Architect (SAA-CO2)

Module I : Overview of AWS (10 Hrs)

On Premise Data Center , What is Cloud Computing, Cloud Computing Offerings - Public, Private and Hybrid Cloud, AWS as the Public Cloud IaaS Leader, AWS Global Infrastructure - Regions and Availability Zones, Elastic Compute Cloud (EC2) Foundation, Create your first EC2 instance, VPC Components - Implied Router and Route Tables,Configuring VPC, IP Addressing - Internet Gateway - Subnet Types, VPC - Network ACLs and Security Groups, Configuring Security groups and NACL, VPC Security Scenarios - Applying Security Group and NACLs, VPC - Network Address Translation – NAT

Practice:

- Launching an EC2 instance
- Configuring Security groups and NACL
- IP Addressing
- Subnetting

Module II: Virtual Private Cloud (VPC) (14 Hrs)

VPC Peering, Transit Gateway, VPC Virtual Private Networks (VPN), VPC Direct, Implementing AWS Direct Connect, Connect Knowledge - Direct Connect Routing and Link Aggregation Groups (LAGs), Hybrid Connectivity, AWS Direct Connect Gateway, AWS Direct Connect Limits, AWS VPC, Creating a VPC, VPC Endpoints - Gateway Endpoint, AWS VPC - VPC Gateway Endpoint, Configuring VPC gateway, Troubleshooting VPC Endpoints and gateways, AWS VPC - VPC Interface Endpoint, Egress Only Internet Gateway (for IPv6), VPC Flow log and DHCP Option Sets

Practice:

- Configuring VPC
- Troubleshooting EC2 and VPC
- Creating a VPC
- Implementing AWS Direct Connect
- Configuring VPC gateway
- Troubleshooting VPC Endpoints and gateways

Module III: Elastic Block Store, Elastic Network Interfaces (14 Hrs)

Elastic Block Store Types, EC2 Enhanced Networking and Placement Groups, EC2 Placement Groups, EC2 Status Checks and Monitoring, EC2 Instance States, EC2 Instance Termination and Termination Protection, EC2 Instance Metadata and User, EC2 Instance Launch Modes VPC and EC2, Instance Tenancy Attribute, Elastic Compute Cloud - Elastic Network Interfaces (ENIs, Elastic Network Interface (ENI) - IP Addressing, NAT instance Source/Destination Check, Public IPv4 address auto assignment, Configring Elastic Block Store, Monitoring and checking health of EC2 instances, Elastic Network Interface, Troubleshooting ENI, Configuring and troubleshooting with EBS

Practice:

- Configuring Elastic Block Store
- Monitoring and checking health of EC2 instances
- Configuring and troubleshooting with EBS

Module IV: Elastic Load Balancer (10 Hrs)

Intoduction, How ELB works , ELB health checks, ELB Cross Zone Load Balancing, ELB Positioning - Internet-facing vs Internal ELB, Refresher for TCP IP Packet flow, ELB - Security Group and ELB - Network ACLs, ELB - Layer 4 TCP/SSL Listeners, ELB - Layer 7 HTTP/HTTPS Listeners, Attaching Elastic load bancer to instances, ELB Gealth check and ELB Security groups, NACL configuration, troubleshooting ELB - Application and Network

Practice:

- Attaching Elastic load balancer to instances
- ELB Health check
- ELB Security groups
- NACL configuration
- Troubleshooting ELB - Application and Network

Module V: Auto Scaling, Simple Notification Service **(10 Hrs)**

Auto Scaling Components, Auto Scaling Features, Auto Scaling Availability Zone Rebalance feature, Adding/Detaching EC2 instances to/from Auto Scaling Groups, Auto Scaling and Elastic Load Balancing Service, Auto Scaling Health Checks, Simple Notification Service, SNS Notifications & Merging Auto Scaling groups, Implementing Auto Scaling, Troubleshooting Autoscaling with new EC2 instances

Practice:

- Implementing Auto Scaling
- Adding/Detaching EC2 instances to/from Auto Scaling Groups
- Auto Scaling health checks
- Simple Notification Service
- Troubleshooting Autoscaling with new EC2 instances

Module VI: AWS Elastic File System (EFS) **(16 Hrs)**

Introduction to the Service and Mount Targets, AWS EFS - Use Cases,Use in On Premise Servers,Storage Classes, Pricing, AWS EFS - Data Encryption, EFS Data backup, EFS and AWS Datasync, Monitoring EFS, AWS FSx for Windows File Server - Introduction, Deployment options & Encryption, Amazon FSx - Data Protection, Backup/Restore, Access, Monitoring and Security, Amazon FSx for Lustre, Amazon EFS vs FSx for Windows vs FSx for Lustre, Creating Elastic File System, Monitoring EFS, Implementing Data Encryption, Setting up AWS FSx Windows File Server

Practice:

- Creating Elastic File System
- Monitoring EFS
- Implementing Data Encryption
- Setting up AWS FSx Windows File Server

Module VII: Elastic Map Reduce **(10 Hrs)**

Amazon Elastic Map Reduce Introduction, AWS EMR - Clusters, Nodes, and deployment in an AZ, Amazon Elasticache Introduction, Amazon ElastiCache - Caching Strategies, Amazon Elasticache for Memcached, Amazon Elasticache for Redis, Amazon CloudFormation, Implementing AWS EMR, Deployment in Availability Zone, Amazon ElastiCache, Amazon Elasticache for Redis, Amazon CloudFormation

Practice:

- Implementing AWS EMR
- Deployment in Availability Zone
- Amazon ElastiCache
- Amazon Elasticache for Redis
- Amazon CloudFormation

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

AWS Developer

Module I: IAM and EC2 (14 Hrs)

IAM Introduction, IAM Hands-On, EC2, Connect to SSH using Linux / Mac and Windows and Troubleshooting, Security Groups, Private vs Public vs Elastic IP, Configuring Webserver - Apache and Tomcat, Elastic Network Interfaces
ELB and ASG: High Availability and Scalability, Elastic Load Balancing (ELB), Classic Load Balancer (CLB), Application Load Balancer (ALB), Network Load Balancer (NLB), Elastic Load Balancer - Cross Zone Load Balancing, Elastic Load Balancer - SSL Certificates, Elastic Load Balancer - Connection Draining, Auto Scaling Groups - Scaling Policies
EBS and EFS: EBS Volume Types Deep Dive, EBS vs Instance Store
RDS ,Aurora and ElastiCache: AWS RDS Overview, RDS Read Replicas vs Multi AZ, RDS Encryption and Security, Aurora, ElastiCache and its Strategies
Practice:

- Launching EC2 instance and Connect to SSH using Linux / Mac and Windows
- Configuring Webserver - Apache and Tomcat
- Implementing Elastic Load Balancing
- Configuring EBS volume
- Create database using AWS Aurora

Module II: Route53 (10 Hrs)

Route 53: EC2 Setup, Route 53 - TTL, CNAME vs Alias, Routing Policy - Simple, Weighted, Latency Route 53 Health Checks, Routing Policy - Failover, Geolocation, Multi Value
VPC: VPC, Subnets, IGW and NAT, NACL, SG, VPC Flow Logs, VPC Peering, Endpoints, VPN, DX, VPC Cheat Sheet & Closing Comments, Three Tier Architecture.
AWS CLI, SDK, IAM Roles & Policies: AWS CLI Setup, Configuration on Windows and Linux, CLI Installation Troubleshooting, Using CLI - EC2, S3, IAM Roles and Policies, AWS Policy Simulator, AWS CLI Dry RunAWS CLI STS Decode, AWS EC2 Instance Metadata, AWS CLI Profiles, AWS CLI with MFA, AWS SDK, Exponential Backoff & Service Limit Increase, AWS Credentials Provider & Chain, AWS Signature v4 Signing
Practice:

- Route 53
- Create Amazon VPC
- AWS CLI Setup and Configuration on Windows and Linux
- CLI Installation Troubleshooting
- AWS CLI with MFA and AWS SDK

Module III: AWS S3 (10 Hrs)

Amazon S3: Amazon S3 - S3 Buckets and Objects, S3 Versioning and Encryption, S3 Security & Bucket Policies, S3 Consistency Model
S3 and Athena: S3 MFA Delete, S3 Default Encryption, S3 Access Logs, S3 Replication (Cross Region and Same Region), S3 Pre-signed URLs, S3 Storage Tiers + Glacier, S3 Lifecycle Policies, S3 Performance, S3 & Glacier Select, S3 Event Notifications, Athena, S3 Lock Policies & Glacier Vault Lock

CloudFront: CloudFront - CloudFront Caching & Caching, CloudFront Security, CloudFront Signed URL / Cookies

Practice:

- Creation of S3 bucket and uploading of objects to it
- S3 Versioning and Encryption
- S3 Replication and S3 glacier
- Application of CloudFront with S3

Module IV: AWS ECR (14 Hrs)

ECS, ECR, Fargate - Docker in AWS: ECS, What is Docker, ECS Clusters, ECS Task Definition, ECS Service, ECS Service with Load Balancers, ECR, Fargate, ECS IAM Deep Dive & Hands On, ECS Task Placement and Constraints, ECS Auto Scaling

Elastic Beanstalk: AWS Elastic Beanstalk, Beanstalk First Environment, Second Environment, Beanstalk Deployment Modes, Beanstalk CLI and Deployment Process, Beanstalk Lifecycle Policy

Beanstalk Extensions, Beanstalk & CloudFormation, Beanstalk Cloning, Beanstalk Migrations, Beanstalk with Docker, Beanstalk Advanced Concepts, Code Commit, Code Pipeline, Code Build in VPC, Code Deploy for EC2 and ASG, AWS Code Star

CloudFormation: AWS CloudFormation - Create Stack, Update and Delete Stack, YAML, Resources, Parameters, Mappings, Outputs, Conditions, Conditions, Intrinsic Functions, RollBacks, ChangeSets, Nested Stacks & StackSet

Practice:

- ECS Service with Load Balancers
- ECS Auto Scaling
- Elastic Beanstalk CLI and Deployment Process
- Applying Code Commit and Code Pipeline
- Implementing AWS CloudFormation

Module V: Monitoring and Audit (10 Hrs)

Monitoring and Audit: AWS Monitoring, AWS CloudWatch Metrics, Alarms and Logs, CloudWatch Agent, Logs Metric Filters, CloudWatch Events, EventBridge Overview, X-Ray, X-Ray: Instrumentation and Concepts, Sampling Rules, APIs, Beanstalk, AWS CloudTrail, CloudTrail vs CloudWatch vs X-Ray

Integration and Messaging - SQS, SNS and Kinesis: AWS Integration & Messaging, AWS SQS, Dead Letter Queue, SQS CLI, FIFO Queues, SQS Advanced, AWS SNS, AWS Kinesis, KCL, Kinesis Security, Firehose and Analytics, SQS vs SNS vs Kinesis, Data Ordering for Kinesis vs SQS FIFO

Practice:

- Monitoring and checking health of the resource
- Application of AWS X-Ray
- Messing using SNS and SQS
- Streaming of Data using AWS Kinesis

Module VI: Serverless Using Lambda (18 Hrs)

Serverless Using Lambda: AWS Lambda, Lambda Synchronous Invocations, Lambda & Application Load Balancer, AWS Lambda@Edge, Lambda Asynchronous Invocations & DLQ, Lambda & CloudWatch Events / Event Bridge, Lambda - S3 Event Notifications, Lambda Event Source Mapping, Lambda Destinations, Lambda Permissions - IAM Roles & Resource Policies, Lambda Monitoring & X-Ray Tracing, Lambda in VPC, Lambda Function Performance, Lambda Concurrency, Lambda External Dependencies, Lambda and CloudFormation, Lambda Layers, Versions and Aliases, CodeDeploy, Limits

Serverless using DynamoDB: DynamoDB - Throughput, Basic APIs, Indexes (GSI + LSI), Optimistic Concurrency, DAX, Streams, TTL, CLI, DynamoDB Transactions, Session State, Partitioning Strategies, Conditional Writes, Concurrent Writes & Atomic Writes, Patterns with S3, Operations and Security

Serverless using API Gateway: API Gateway - API Gateway Stages and Deployment, Canary Deployments, Integration Types & Mappings, Swagger & Open API, Caching, Usage Plans & API Keys, Monitoring, Logging and Tracing, Authentication and Authorization, REST API vs HTTP API vs WebSocket API

Serverless Application Model: SAM, Installing the SAM CLI, Creating first SAM Project, Deploying SAM Project, API Gateway, DynamoDB, CloudFormation Designer and Application Repository, Policy Templates, CodeDeploy

Practice:

- Creating AWS lambda functions
- Lambda Permissions - IAM Roles & Resource Policies
- Creating and Maintaining database using DynamoDB
- API Gateway configuration
- Deploying Serverless Application

Module VII: AWS cognito (08 Hrs)

Cognito: Cognito, User Pools, Identity Pools, Cognito User Pools vs Cognito Identity Pools, Sync

Security and Encryption: AWS Security, Encryption 101, KMS, KMS using CLI, KMS Encryption Patterns and Envelope Encryption, Limits, S3 Security Advanced, SSM Parameter Store Overview, Secrets Manager, SSM Parameter Store vs Secrets Manager, CloudWatch Logs Encryption, CodeBuild Security

Practice:

- Implementation of AWS Cognito
- AWS Key Management Service
- Key Management Service Encryption Patterns

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

Project

- Locally Debug a Serverless App
- Build a Serverless Web Application
- Design a database for a mobile app with Amazon DynamoDB
- Create a Load Balanced WordPress website
- Building serverless applications
- Add Voice to Your WordPress Site

- Deploy a Python Web App
- Migrate a Git Repository to AWS
- Build a Drupal Website
- Build a Modern Application
- Build, Train, and Deploy a Machine Learning Model
- Handle Errors in Serverless Applications
- Create and Manage a Nonrelational Database
- Launch and Configure a LAMP Website
- Deploy and Host a ReactJS App
- Set Up a Compliant Archive
- Host a Static Website
- Launch a Linux Virtual Machine
- Launch a WordPress Website
- Remotely Run Commands on an EC2 Instance
- Launch a Windows Virtual Machine
- Create and Query a NoSQL Table
- Introduction to Deep Learning
- Train a Deep Learning model
- Create a machine learning model automatically with Amazon SageMaker Autopilot
- Setting up a Document Database With Amazon DocumentDB (with MongoDB compatibility) and AWS Cloud9
- Detect, Analyze, and Compare Faces
- Create an AWS DeepLens Project
- Provision Desktops in the Cloud
- Publish Amazon SNS Messages Privately
- Deploy WordPress with Amazon RDS
- Create and Manage a Nonrelational Database
- Create an Audio Transcript

Gate Process for Project

Gate 0: Planning
Gate 1: Analysis
Gate 2: Design
Gate 3: Implementation
Gate 4: Testing
Gate 5: Deployment

Course Outline Prepared by: Prof. K.V. Kalyan/ Prof. Raj Kumar Mohanta

Courseware link: http://courseware.cutm.ac.in/courses/domain-cloud-domain/

Gaming and Immersive Learning (AR & VR)

		DOMAIN		
CUTM	**ARCU2060**	**Gaming and Immersive Learning (AR & VR)**	**20**	5+5+14
	CUAR2060	Introduction to Gaming & Simulation	2	1+1+0
	CUAR2061	Game Assets & Game Objects	3	1+1+1
	CUAR2062	Building Game Environment	3	1+1+1
	CUAR2063	Game Animation, Scripting & UI	3	1+1+1
	CUAR2064	Binary Deployment and Cross-Platform Controls	3	1+1+1
	CUAR2065	Project	10	0+0+10

Course Objectives

- Students will know about the History of Computer Graphics
- Know about Gaming Industry and get a job in AR/VR field
- Understanding of Individual Roles in a Gaming Industry
- End to End Game Development skill

Course Outcomes

COs	**Course Outcomes**	**Mapping COs with POs (High-3, Medium-2, Low-1)**
CO1	Able to gain knowledge on game Assets Development, efficient Import/ Export of assets for Games.	PO1 (3)
CO2	Able to design Animation in Games, User Interface, Cross-Platform Support of a Game, Technical and Specification Document of a Game etc	PO3 (3), PO5(3)

1. Introduction to Gaming & Simulation

Module I: Welcome to Game Engine (2+2.5)

Importance of Storyboarding a Game Idea, The Economics of Game Development, Assessing Game Markets and Platforms, Marketing Methods for Games, Monetizing Games and Upgrades.

Module II: Introduction to Game Production (1)

Video Game Platforms and Genres, Describing the Game Production Pipeline, Game Development Jobs and Roles, The Game Design Document, The Technical Design Document, Getting Started in Unity, Creating a New Unity Project, Using the Unity Asset Store, Source Control for Working in Team.

Module III: The Game Engine User Interface (2+2)

Introduction to the Unity Editor Interface, Analyzing the Unity Editor User Interface, Utilizing the Unity Editor User Interface, Navigating the Scene View Window, Utilizing the Game View Window, Navigating the Hierarchy Window, Using the Inspector Window, Managing Assets in the Project Window, Searching and Filtering in the Project Window, Organizing the Scene with Layers.

Module IV: Using Game objects and assets (1+2)

Creating and Modifying Game Objects, Defining Unity Editor Units, Describing Assets in the Production Pipeline, Review: Defining an Asset, Organizing Assets in the Unity Editor, Defining a Game Object.

Module V: Defining a Game Object (2+2)

Creating Unity-native Game Objects, Manipulating Game Objects in the Unity Editor, Describing What is a Unity-native Game Object, The Role of Components in the Unity Editor, Defining Prefabs and Scene Structure, Defining the Role of the Prefab in Unity, Creating and Saving a Scene.

Module VI: The Hierarchy of Scenes within a Game (1+2)

Importing Assets into a Project, Importing and Configuring a 3D Model, Importing Textures for Use in Materials, Importing FBX Files with Animation, Working with Sprites, Introduction to Sprites in Game Development.

Module VII: Managing Projects and Assets (1+2)

Project Management in Unity, Introduction to Game Project Management, Managing Assets, Using the Unity Asset Store (Reprise), Importing Offline Content, Creating Project Structure Based on Assets, Sorting the Zombie Toys Prop Model Assets, Setting Resolution and Type of Texture Files.

2. Game Assets and Objects

Module I: Preparing Assets for implementation (1+2.5)

Best Practices in 3D Content Creation, Modelling for Games, Animating for Games, UV Mapping and Texturing Techniques, and Exporting to Unity, Importing into Unity, Materials in Unity, The Interaction of Lighting and Materials.

Module II: Discovering the Standard Shader in Unity (1+2)

Exploring other Material Types, Analyzing the Benefits of Customer Shaders, Creating the Materials for Zombie Toys Props, Duplicating and Modifying Materials, Case Studies in Material Creation, Managing and Using Textures in the Unity Editor, Texturing for Game Development, Optimization and Reuse of Textures.

Module III: Assembling the Game Level (2+2)

Branching and Hierarchies, Creating Hierarchies in Unity, Using Empty Game Objects as Pivots, Introduction to Physics in Unity, Understanding the Physics System in Unity, Introduction to the Rigid body Component.

Module IV: Introduction to Colliders (1+2)

Creating the Colliders for Zombie Toys Props, Introduction to Game Level Design, Introduction to Game Level Design, The Level Design in Zombie Toys, Placing Objects in a Scene, Importing the Prop Prefabs into the Scene, Cloning the Stars, Creating the Level Boundaries.

Module V: Lighting in Games (1+2)

Introduction to Game Lighting, Introduction to Game Lighting, Differences in Lighting for Games and for Film, Placing and Adjusting Lights in a Scene, Analyzing the Different Lights and Properties, Light Types and Behaviors, Using Layers to Exclude Objects from Lighting, Casting and Modifying Shadows, Mesh Renderer Attributes for Shadows.

Module VI: Differentiating Shadow Types (1+2)

Creating Cookies to Shape Lights, Faking Shadows for Better Performance, Benefits of Faking Shadows in Games, Utilizing Painted Shadows, Using Projectors to Project Shadow Cookies, Lighting the Zombie Toys Game, Lighting the Zombie Toys Scene, Lighting Variations for Changing the Mood.

Module VII: Baking Lighting in Game Production (1+2)

Light Baking in Video Games, Introduction to Light Baking in Video Games, Setting Objects to Participate in Light Baking, Marking Objects as Static for Light Baking, Creating UV Coordinates for Light Baking, Baking Lighting, Continuous and Manual Light Baking, Placing Light Probes for Moving Objects, Creating Reflection Probes, Baking the Lighting in Zombie Toys, Creating the Light Probes in Zombie Toys.

3. Building Game Environment

Module I: Building the Player and Allies (2+2)

Creating a Player Controller, Examining Why to Use a Custom Controller, Creating the Player Controller Game Object, Adding a Game Manager, Explaining the, Purpose of the Game Manager, Making the Controller Functional, Adding Scripts for Behavior, Configuring the Camera, Creating the Sheep Ally, Building the Sheep Ally From a Model, Creating the Dog Ally, Building the Dog Ally From a Model.

Module II: Building the Enemies (2+2)

Creating an Enemy, Designing the Enemy Behaviors, Creating the First Enemy Character, Creating the Enemy Animator Controller, Creating Additional Enemies, Creating the Zombear

Enemy, Creating the Zombie Duck Enemy, Creating the Other Enemies, Integrating Enemies into the Game, Placing the Spawn Points, Spawning the Enemies.

Module III: Introduction to Unity's Particle System **(1+1)**

Analyzing Existing Particle Effects, Setting Up the Interface for Effects, Case Study: Developing the Lightning Attack, Overview of the Lightning Attack, Building the Lightning Attack Hit, Building the Lightning, Attack Emitter, Building the Lightning Bolt, Integrating the Lightning Attack into the Game.

Module IV: Creating Particle Systems **(1+2)**

Intro to the Particle Systems in the Unity Editor, Examples of Unity Particles in Video Games, the Role of the Effects Artist in Video Games, Comparing Game Effects with Other Media, And Production Best Practices for Particle Systems.

Module V: Case Study **(1+2)**

Developing the Frost Attack, Introduction to the Frost Attack, Building the Frost Debuff, Building the, Frost Attack Emitter, Building the Frost Cone Effect, Integrating the Frost Attack into the Game, Case, Study: Developing the Stink Bomb Attack, Introduction to the Stink Bomb Attack, Creating the Stink Bomb Hit Effect.

Module VI: Case Study **(1+2)**

Developing the Slime Attack, Introduction to the Slime Attack, Creating the Slime Hit Effect, Creating the Slime Debuff, Creating the Slime Attack Reticle, Building the Slime Attack Emitter, Building the Slime Projectile, Integrating the Slime Attack into the Game, Finalizing Player Attacks, Adding the Ally Manager.

Module VII: Adding Audio to Game Levels **(1+2.5)**

Introduction to Audio in Game Development, Introduction to Audio in Game Development, Importing Audio into Unity, Introduction to Importing Audio in Unity, Supported Audio Formats in Unity, Playing Audio in the Unity Editor, , Testing Audio Sources in the Scene, Mixing Audio in Unity, Using Audio Mixers and Audio Mixer Groups, Setting up the Zombie Toys Audio Mixers, Creating Audio Effects, Introduction to Audio Effects.

4. Game Animation, Scripting & UI

Module I: Animating Game Objects in the Unity Editor **(1+2)**

Introduction to Animation in Game Development, Introduction to Animation in Game Development, Animating in the Unity Editor, Creating Animation in the Unity Editor, Refining Animation in the Unity Editor.

Module II: Bringing Animation into the Game **(1+2)**

Importing Animated Characters, Introduction to Rigging and Imported Animation, Recognizing Asset Data when Importing, Differentiating Available Rig Animation Types.

Module III: Animation Creation and Controlling **(1+2)**

Creating and Naming AnimationClips, Creating an Animator Controller, Introduction to the Animator Controller, Creating and Modifying Animation States, Creating Parameters to Control Transitions, Creating an Animator Override Controller.

Module IV: Scripting in Game Development **(2+2.5)**

Intro to Scripting in Game Development, Intro to Scripting in Game Development, Creating Scripts in Unity, Creating and Saving a Script in Unity, Analysing the Default, Script Methods, Scripting Primer and Best Practices, Attaching a Script to a Game Object. Declaring Variables, List of Variable Types, Creating Conditions in Scripting, Introduction to Conditions, The "if' Condition, The "if else" Condition, Complex Conditions, Looping.

Module V: Designing User Interfaces for Games **(1+2)**

Introduction to Designing the User Interface, Assessing User Interface Design Needs, Examining the UI Tools in the Unity Editor, Creating a User Interface, Investigating the Canvas Functionality, Utilizing the Power of the Rect Transform, Creating a UI Button, Creating a UI Image, Creating UI Text, Creating Interaction in the UI with Events.

Module VI: Introduction to Looping **(1+2)**

The "while" Loop, The "for" Loop, Creating Custom Methods, The Benefits of Using Custom Methods, Utilizing Arguments, Utilizing Method Return Types, Coroutines, Introduction to Coroutines, Accessing Components via Script, Utilizing the GetComponent() Function, Common Code Cases, Common Pieces of Zombie Toys Code..

Module VII: Implementing Navigation and Path Finding **(1+2)**

Introduction to Navigation and Path Finding, Introduction to Navigation in Unity, Describing a NavMesh, Defining a NavMesh Agent, Describing a NavMesh Obstacle.

5. Binary Deployment and Cross-Platform Controls

Module I: Building the Camera and Player Selection System **(1+3)**

Intro to the Camera and Player Selection Behaviors, Analyzing the Player Selection System.

Module II: Creating another Player Option **(2+2.5)**

Making the Player Selectable, Adding Another Player, Finalizing the Camera.

Module III: Adding Camera Animations **(2+2)**

Configuring the Camera Animator Controller, Applying Behavior to the Camera, Adding Character Selection Spotlights.

Module IV: Building and Deploying the Game **(1+1)**

Building the Game, Introduction to the Build Process, Adjusting the Player Settings, Building the Game.

Module V: Protecting Your Creation **(1)**

Legal Considerations for Your Game, Unity Services, Unlocking the Unity Platform Potential, Surveying Unity Services

Module VI: Understanding of Cross-Platform Inputs **(1+2)**

Different Input types like, Mobile, WebGL, OpenVR & other unity supported platforms.

Module VII: Preparing for Mobile Deployment **(2+2)**

Modifying Zombie Toys for Mobile, Introduction to Mobile Development in Unity. Changing the Build Platform to Mobile, Adding the Mobile Interface UI, Implementing Mobile Input Behaviors.

Text Books:

1) Jared Halpern, Developing 2D Games with Unity: Independent Game Programming with C#, Apress, Final Edition
2) Jon Manning, Paris Buttfield-Addison, and Tim Nugent, Unity Game Development Cookbook: Essentials for Every Game, O'Reilly Media, Inc.
3) Reference Books:
4) Jason Gregory, Game Engine Architecture, CRC Press, Third Edition
5) Linowes Jonathan, Unity Virtual Reality Projects, Packt, Second Edition

Course outline Prepared by: Abhi Mitra ***Date: July 6, 2020***
Source of reference: UCA Courseware.

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

Data Science and Machine Learning

Code	Course Title	T-P-Pj (Credit)	Prerequisite
MLCU2000	**Data Science and Machine Learning**	2-9-15	NIL

Course Objectives

- Understand the scope, stages, applications, effects and challenges of ML.
- Understand the mathematical relationships within and across ML algorithms and the paradigms of supervised and unsupervised learning.
- Able to get jobs in AI/ML field

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain knowledge on design and implementation of various machine learning algorithms in a range of real-world applications.	PO1 (3)
CO2	Able to analyze prediction and classification using different ML and deep learning methods	PO2(3)
CO3	Ability to solve the real world problems using ML solutions in their respective fields of study.	PO2(3)
CO4	Ability to design product	PO3 (2), PO5(2)
CO5	Able to do research for publishing articles/ file patents.	PO4 (3)

ML for Predictive Analysis (0+2+2)

https://careerfoundry.com/en/blog/data-analytics/regression-vs-classification/

Project/Task: (Choose two projects, Prediction and Classification)
Time Series Analysis
Health Care System

Concept Required:

3.1 Data pre-processing: -

- Accessing / collecting the datasets from different online repository.
- Missing values handling, noise reduction, finding Correlation between features, outlier elimination.
- Label Encoding / Encoding the categorical data
- Splitting the dataset
- Data Normalization

3.2 Learning Algorithms: -

- Supervised Learning Algorithms

- Unsupervised Learning Algorithms

3.3 Feature extraction and selection: -

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Different Feature Selection Techniques / Algorithms

3.4 Model building: -

- Regression (Linear, Polynomial, multiple, logistic), Decision Tree, Random Forest.
- Artificial Neural Network (Feed Forward Neural Network, Gradient Descent, Back Propagation Neural Network).
- Convolutional Neural Network
- Other Pretrained Models

3.5 Performance measures: -

- Perdition: Root Mean Square Error (RMSE), Average Percentage Error (APE), Mean Average Percentage Error (MAPE).
- Classification: Confusion Matrix (TN, TP, FP, FN), Sensitivity, Specificity, Gmean, F-score, Overall Accuracy, (Receiver Operating Characteristic) ROC Curve. Area under Curve (AUC)

3.6 Reading and Writing Research Articles

ML for Image Analytics (0-2-2)

Project/Task: (Choose one among six Tasks)

Detection of optometry diseases using retinal fundus imaging.

1. Diabetic Retinopathy
2. Glaucoma
3. Cataract

Detection of various diseases using X-ray imaging.

1. Covid19

Leaf disease classification using RGB images.

1. Tomato leaf
2. Potato leaf

Concept Required:

Image Pre-processing:-

- Accessing individual pixels using matrix concept
- Image resize, grey scale conversion, Colour channel splitting
- Histogram equalisation (CLACH).

Image Feature Extraction: -

- Edge detection (Sobel, Canny), Morphological operations
- Image segmentation, Image Thresholding, Binary conversion
- Cluster based segmentation
- Feature extraction based on size, shape and colour
- Feature extraction using predefined functions: SIFT, SURF, STAR, ORB.
- Feature Extraction using convolutional neural network (CNN).

Creation of Feature Matrix by combining Extracted Features: -

- Matrix flattening, Horizontal stacking, Vertical stacking, padding.
- Splitting the feature matrix (training/testing) and labelling.

Classification algorithms: -

- Support vector machine (SVM)
- Different kernels of SVM (linear, polynomial, radial basis function).
- Gradient Boosting (GB)
- Multi-layer Perceptron (MLP), deep learning.

Mathematics for ML (2+1+0)

When Models Meet Data:-

- Data, Models, and Learning
- Empirical Risk Minimization
- Parameter Estimation
- Probabilistic Modeling and Inference
- Directed Graphical Models
- Model Selection

Linear Regression:-

- Problem Formulation
- Parameter Estimation
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection

Dimensionality Reduction with Principal Component Analysis:-

- Problem Setting
- Maximum Variance Perspective
- Projection Perspective
- Eigenvector Computation and Low-Rank Approximations
- PCA in High Dimensions
- Key Steps of PCA in Practice
- Latent Variable Perspective

Density Estimation with Gaussian Mixture Models:-

- Gaussian Mixture Model
- Parameter Learning via Maximum Likelihood
- EM Algorithm
- Latent-Variable Perspective

Classification with Support Vector Machines:-

- Separating Hyperplanes
- Primal Support Vector Machine
- Dual Support Vector Machine

- Kernels
- Numerical Solution

Practice:

- Curve Fitting in Python.
- Exploratory Data Analysis in Python.
- Kernel Density Estimation in Python.
- Probability Distribution Function Plotting in Python.
- Cumulative Distribution Function Plotting in Python.
- Dimensionality Reduction and Feature Extraction in Python.

References:

1. Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong.
2. https://youtube.com/playlist?list=PLLy_2iUCG87D1CXFxE-SxCFZUiJzQ3IvE

IoT Analytics (0+2+2)

Defining IoT Analytics and Challenges

IoT
Benefits of Deploying IoT
End to End IoT architecture
IoT challenges

IoT Protocols

1 Wireless Protocol

Connectivity Protocols (when Power is Limited)
Bluetooth Low Energy (BLE)
Zigbee
LoRaWAN
NFC

2 Connectivity Protocols (when Power is Not a problem)

Wifi

3 Data Communication Protocol

MQTT
Web-Socket
HTTP

2 Sensors

Types of Sensors based on communication-I2C, SPI
Types of Sensors based on Application

3 Overview of 32 -bit Controller

ESP8266
ESP32
Raspberry Pi

4 AWS IoT for Cloud

AWS IoT Core services
AWS IoT Analytics services
AWS DynamoDB Services

5 Thingspeak for IoT

Getting and posting Data to IoT Cloud using ESP devices
Posting Data to IoT Cloud using Raspberry Pi

6 ThingWorx for Industrial IoT

Building Dashboard on Thingworx platform
Binding the senor value to the dashboard

Text Book:

1. Minteer, Andrew. *Analytics for the Internet of Things (IoT)*. Packt Publishing Ltd, 2017.

Reference Books:

1. Geng, Hwaiyu, ed. *Internet of things and data analytics handbook*. John Wiley & Sons, 2017.

Digital video Processing (0+2+1)

UNIT 1:

Fundamentals of Video Processing: Digital Video Acquisition, Principles of Color Video, Video Camera, Video Display, **Analog Vs Digital Video**: Progressive Vs Interlaced scans, Signal, Bandwidth Characterization of a Digital Video Signal.

Practice:

- Read and play video files
- Extract frames from video files
- Combine frames to create a video file

UNIT 2:

Fourier Analysis of Digital Video Signals: Spatial and Temporal resolution, Fourier Analysis of Digital Video Signals, **Spatial-Temporal Sampling:** Temporal Frequency Response and Flicker Perception. Spatial Frequency Response, Spatiotemporal Frequency Response, Smooth Pursuit Eye Movement

Practice:

- Applying fourier transformation on video
- Time domain analysis
- Frequency domain Analysis

UNIT 3:

Digital Video Formats: Significance of Video Formatting, Data rate and bandwidth trade-off, **File Formats:** MP4, MOV, WMV, AVCHD, FLV, AVI, WebM, MKV

Digital Video Compression Standards: Digital Video Compression Metrics, Digital Video Storage Precisions, Significance of Video compression, **Video Compression Codec's:** Motion JPEG, JPEG 2000, H.264/MPEG-4 AVC, VP8, HEVC, H.265 High Efficiency Video Codec.

Practice:

- Conversion of video files from one format to another.
- Using Motion JPEG Codec

- Using MPEG-4 Codec
- Using H.265 Codec

UNIT 4:

Digital Video Editing Basics: Video Editing Types- Online, Offline, Linear, Non-linear, Assemble, Insert, Rough-cut, Video Shot Transition Effects: Cut, Fade, Wipe, Dissolve, B-roll, Video Shot Boundary Detection Methods: pixel differences, statistical differences, histogram comparisons, edge differences and motion vectors. Video Shot Detection Performance Metrics: ROC Curves, Recall, Precision, F-Measure

Practice:

- Video Shot Detection using pixel Difference
- Video Shot Detection using Histogram based methods
- Video Shot Detection using Edge based methods
- Video Shot Detection using Motion Vectors

Project List

1. Creating a VIDEO object detection system
2. Vehicle detection in Videos using OpenCV and Python
3. Detecting faces in live camera feed with identification of the person.

TEXT BOOK:

1. Rafael C Gonzalez and Richard E Woods, “Digital Image Processing’’, Pearson Education, 3rd Edition, 2009.
2. Handbook of Image and Video processing - Al Bovik (Alan C Bovik), Academic Press,

REFERENCE BOOK:

1. Fundamentals of Digital Image Processing”, Anil K. Jain, PHI, 1995.
2. “Digital Image Processing”, William. K.Pratt, Wiley Interscience, 2nd Ed, 1991.

Project (0-0-4)

Internship (0-0-4)

Android App Development

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

Course Objectives

- 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
- Introduction Android Database, Introduction SQLite, and Content value working with SQLite Databases.
- Got Skill To Create Files, Saving Files And Understanding Database Is Able To Manage Application With Database In order to implement the various process

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Individual after acquiring the knowledge of Android is able to Create Activities, Applications, Network-Based Application With Database Individual	PO1 (3)
CO2	Analyze Broadcast Receiver, Adapters And Internet Are Able To Create Network-Based Application	PO2(3)
CO3	Design and develop Android mobile App	PO3 (3), PO5 (2)

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):

- Android Studio

Information Security

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

Course Objectives

- 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.

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Will gain familiarity with computer network, defences against them, and forensics to investigate the aftermath	PO1 (3)
CO2	Develop a basic understanding of Risk assessment	PO2(3)
CO3	Develop an understanding of security policies as well as protocols to implement such policies	PO2(3)

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:

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

Machine Learning using Python

Code	Course Title	Credit	T-P-PJ
CUTM1019	Machine Learning using Python	4	1-2-1

Course Objectives

- Understand the meaning, purpose, scope, stages, applications, and effects of ML.
- Explore important packages of python, such as numpy, scipy, OpenCV and scikit-learn.

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to gain knowledge on ML solutions in their respective fields of study	PO1 (3)
CO2	Able to analyze several problems and apply ML techniques to solve it	PO2(3)
CO3	Ability to design prediction and classification models	PO3 (3)

COURSE CONTENT

Module 1 – Application and Environmental-setup (12 hrs)

- Applications of Machine Learning In different fields (Medical science, Agriculture, Automobile, mining and many more).
- Supervised vs Unsupervised Learning based on problem Definition.
- Understanding the problem and its possible solutions using IRIS datasets.
- Python libraries suitable for Machine Learning(numpy, scipy, scikit-learn, opencv)
- Environmental setup and Installation of important libraries.

Module 2 - Regression (8 hrs)

- Linear Regression
- Non-linear Regression
- Model Evaluation in Regression
- Evaluation Metrics in Regression Models
- Multiple Linear Regression
- Feature Reduction using PCA
- Implementation of regression model on IRIS datasets.

Module 3 - Classification (24 hrs)

- Defining Classification Problem with IRIS datasets.
- Mathematical formulation of K-Nearest Neighbour Algorithm for binary classification.
- Implementation of K-Nearest Neighbour Algorithm using sci-kit learn.
- Classification using Decision tree.
- Construction of decision trees based on entropy.
- Implementation of Decision Trees for Iris datasets .
- Classification using Support Vector Machines.
- SVM for Binary classification
- Regulating different functional parameters of SVM using sci-kit learn.
- SVM for multi class classification.
- Implementation of SVM using Iris datasets .
- Implementation of Model Evaluation Metrics using sci-kit learn and IRIS datasets.

Module 4 - Unsupervised Learning (12 hrs)

- Defining clustering and its application in ML .
- Mathematical formulation of K-Means Clustering.
- Defining K value and its importance in K-Means Clustering.
- Finding appropriate K value using elbow technique for a particular problem.
- Implementation of K-Means clustering for IRIS datasets

Projects

- To be defined based on respective study area of student.

References:

Text Book:

1. EthemAlpaydin, Introduction to Machine Learning, Second Edition, http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=12012.

Web Resource:

https://towardsdatascience.com/beginners-guide-to-machine-learning-with-python-b9ff35bc9c51

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
- Acquire skill on algorithm design techniques using a programming language

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Ability to get knowledge on appropriate data type and data structure for a given problem	PO1 (3), PO12(1)
CO2	Analyze to compare algorithms with respect to time and space complexity	PO2(3)
CO3	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	PO2(3)
CO4	Design and develop different algorithms for solving the problems	PO3 (3)

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

DotNET Programming

Course Title	Course Code	Credit	T-P-PJ
DotNET Programming	CUTM2474	4	0-2-2

Course Objectives

- Understand the basics of the .NET Framework and its components.
- Learn the C# programming language.
- Create and manipulate user interfaces using ASP.NET controls.
- Work with data in .NET applications using ADO.NET

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	understanding of the .NET Framework, including its architecture, components, and how to leverage its features for application development	PO1 (3), PO12(1)
CO2	work on various coding exercises and projects, which will enhance problem-solving abilities and logical thinking skills	PO2(3)
CO3	equip you with a solid understanding of C# programming concepts, syntax, and best practices	PO2(3)
CO4	develop applications using the .NET Framework, including building user interfaces, implementing business logic, and working with data storage and retrieval.	PO3 (3), PO5(2)

Module-1: The .NET Technology

Introduction to .NET Framework, Architecture of .NET framework– BCL (Base Class Library), CLR(Common Language Runtime), . NET Languages – introduction, Types of applications supported by .NET Technology, Managed code, compilation to intermediate language, Just-In-Time compilation,garbage collection, assemblies and the GAC

Module-2: C# Basics

Data Types, Variables &amp; Constants, Operators in C#, Arithmetic Operators, Prefix and Postfix notation, Assignment Operators, Relational Operators, Other Operators, Operators precedence, Flow Control and Conditional Statements if-else statement, switch statement, Loops in C#, for loop, do-while loop, Array in C#, foreach Loop.

Module-3: Object and Classes

Concept of a class, Objects, Fields, Methods, Instantiating the class, Accessing the members of a class, Access modifiers, Properties, Static members of the class, Constructors, Desturctors, Overloading Constructors, Value types (out &amp; ref keywords) ,sealed keyword,Inheritance,

Module-4: Object and Classes -II

Polymorphism, Overriding the methods, the new keywords, Type casting, is and as keywords, Boxingand Un-boxing, Structures, Enumeration, Nested Classes, Abstract classes, Exception handling, Delegates &amp; Events

Module-5: ASP.NET - I

Overview of ASP.NET framework, ASP.NET Page – layout, lifecycle, Stages in Web Forms Processing, Introduction to Server Controls, HTML Controls, Validation Controls, User control, Data Binding Controls, Configuration, Session State, Adding controls to a web form, Buttons, Text Box , Labels, Checkbox, Radio Buttons, List Box, etc, Master Pages, themes and skins

Module-6: ASP.NET – 2

Overview of ADO.NET, Benefits of ADO.NET, ADD.NET compared to classic ADO, ADO.NET

architecture (Connected and Disconnected), State Management, Using Database connection.

Working with DataSets, Managed Providers, Data Binding, Typed DataSets, Working with Data

Reader, Transactions

Module-7: Packaging and Deploying

Reporting, Web Services – overview, creation and calling, Packaging and Deploying ASP.NET

Applications

Reference Books

1.Andrew Troelsen, Philip Japikse, : C# 6.0 and the .NET 4.6 Framework, Apress, 2017.

2.Black Book: .NET 4.5 Programming (6-in-1) covers .NET 4.5 Framework, Visual Studio 2012, C#2012, ASP.NET 4.5, VB 2012, and F# 3.0, Dreamtech Press, 2013.

3. Matthew MacDonald: Beginning ASP.NET 4.5 in C#, Apress, 2013.

4. G. Andrew Duthie, “ASP.NET programming with Microsoft Visual C#.NET Step by Step”, version 2003, Prentice-Hall of India.

NodeJS

Course Title	Course Code	Credit	T-P-PJ
NodeJS	CUTM2475	2	0-2-0

Course Objectives

- Understand nodejs, module, to create webserver, debugging.

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Will gain nodejs	PO1 (3), PO12(2)
CO2	Identify and fix defects the common safety issues in code	PO2 (3)
CO3	Students will able to Create and debugging application using nodejs	PO3(3), PO5(2)

Module 1: Introduction and setup

Introduction to NodeJS, Advantages of NodeJS, Installation of nodejs

Module 2: Node JS Modules

Functions, Buffer, Module, Modules Types, Core Modules, Local Modules, Modules Exports

Module 3: Node Package Manager

What is NPM, Installing Packages Locally, Installing package globally, Adding dependency in package Json, Updating packages,

Module 4: Creating Web Server

Creating Web Server, Sending Requests, Handling HTTP requests,

Module 5: File System

Read File, Writing a File, Opening a File, Deleting a File, Writing a file asynchronously, Other I/O Operations

Module 6: Debugging Node JS Application

Core Node JS Debugger

Module 7: Events

Event Emitter class, Inheriting Events, Returning event emitter

CLOUD SECURITY ASSOCIATE

Course Outline

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM2477	Cloud Security Associate	0-2-0	

Course Objectives

- To familiar with Azure Security fundamentals and to gain necessary skills to become certified for/with Microsoft Azure Security Technologies

Learning Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	To Secure cloud applications in Azure	PO1 (3)
CO2	Implement resource management security & network security in Azure	PO2 (3)
CO3	Implement virtual machine host security in Azure	PO3(3)

Module 1: Azure Security Technologies (AZ 500 Part 1) : Secure your cloud apps (2 Hours)

Introduction, Defense in depth, Identity management, Infrastructure protection, Encryption, Network security, Application security

Module 2: Azure Security Technologies (AZ 500 Part 2) : Implement Resource Management Security (3 Hours)

Introduction, Protect against security threats by using Azure Security Center, Detect and respond to security threats by using Azure Sentine, Store and manage secrets by using Azure Key Vault, Exercise - Manage a password in Azure Key Vault, Host your Azure virtual machines on dedicated physical servers by using Azure Dedicated Host, Knowledge check

Module 3: Azure Security Technologies (AZ 500 Part 3) : Implement Network Security (4 Hours)

Introduction, What is defense in depth?, Protect virtual networks by using Azure Firewall, Protect from DDoS attacks by using Azure DDoS Protection, Filter network traffic by using network security groups, Exercise - Configure network access to a VM by using a network security group, Combine Azure services to create a complete network security solution, Knowledge check

Module 4: Azure Security Technologies (AZ 500 Part 4) : Implement Host Security (2 Hours)

Introduction, Defense in depth, Identity management, Infrastructure protection, Encryption, Network security, Application security

Module 5: Azure Security Technologies (AZ 500 Part 5) : Manage Identity and Access (4 Hours)

Introduction, Protect against security threats by using Azure Security Center, Detect and respond to security threats by using Azure Sentinel, Store and manage secrets by using Azure Key Vault, Exercise - Manage a password in Azure Key Vault, Host your Azure virtual machines on dedicated physical servers by using Azure Dedicated Host, Knowledge check

Module 6: Azure Security Technologies (AZ 500 Part 6) : Manage Security Operations (4 Hours)

Introduction, Protect against security threats by using Azure Security Center, Detect and respond to security threats by using Azure Sentinel, Store and manage secrets by using Azure Key Vault, Exercise - Manage a password in Azure Key Vault, Host your Azure virtual machines on dedicated physical servers by using Azure Dedicated Host, Knowledge check

CLOUD SECURITY FOUNDATION

Course Outline

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM2476	Cloud Security Foundation	0-2-0	

Course Objectives

- This course is designed to help students gain a foundational knowledge of cybersecurity principles and services for cloud computing through a guided hands-on approach.

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Design and deploy a secure application on AWS	PO1 (3)

Module 1: Welcome to AWS Academy Cloud Security Foundations (2 Hours)

Course prerequisites and objectives, Course overview, AWS Certified Security – Specialty certification, Activity: AWS Documentation Scavenger Hunt

Module 2: Introduction to Security on AWS (2 Hours)

Security in the AWS Cloud, Security design principles, shared responsibility model, Activity: Shared Responsibility Model, Module 2 Wrap-up Video, Assignment, Module 2 Knowledge Check

Module 3: Securing Access to Cloud Resources (6 Hours)

Introduction to Securing Access to Cloud Resources, IAM fundamentals, authenticating with IAM, Authorizing with IAM, Examples of authorizing with IAM, Demo: Amazon S3 Cross-Account Resource-Based Policy, Additional authentication and access management services, Using AWS Organizations, Lab 3.1: Using Resource-Based Policies to Secure an S3 Bucket, Introduction to Securing Your Infrastructure, Structure of a three-tier web application, Using a VPC, Setting up public and private subnets and internet protocols, Using AWS security groups, Using AWS network ACLs, Using AWS load balancers, Pulling it all together, Protecting your compute resources, Module 3 Wrap-up video, Assignment, Lab 3.2: Securing VPC Resources by Using Security Groups, Module 3 Knowledge Check

Module 4: Protecting Data in Your Application (3 Hours)

Protect data at rest, Amazon S3 protection features, Protection through encryption, Protect data in transit, Best practices for protecting data in Amazon S3, Additional data protection services, Module 4 Wrap-up video, Assignment, Lab 4.1: Encrypting Data at Rest by Using AWS KMS, Assignment, Module 4 Knowledge Check

Module 5: Logging and Monitoring (3 Hours)

Introduction to Logging and Monitoring, Importance of logging and monitoring, Capture and collect, AWS services with built-in logs, Monitor and report, Best practices for logging and monitoring, Additional AWS services for logging and monitoring, Demo: Security Hub, Module 5 Wrap-up video, Assignment, Lab 5.1: Monitoring and Alerting with CloudTrail and CloudWatch, Module 5 Knowledge Check

Module 6: Responding to and Managing an Incident (3 Hours)

Introduction to Responding to and Managing an Incident, Identifying an incident, AWS services that support the discovery and recognition phase, AWS services that support the resolution and recovery phase, Best practices for handling an incident, Module 6 Wrap-up video, Assignment, Lab 6.1: Remediating an Incident by Using AWS Config and Lambda, Assignment, Module 6 Knowledge Check