

Syllabus

Department of Information Technology

**Bachelor of Computer Applications
(2024-2028)**

Bachelor of Computer Applications: Course Structure and Syllabus
Credit Requirement: 96 / 128
Duration: Three Years (6 Semesters) / Four Years (8 Semester)

Eligibility: Class XII (Any stream)

Program Outcomes:

1. Knowledge and comprehension: Students will have a solid grasp of the fundamental ideas and ideas of computer science and its applications, including programming, algorithms, data structures, computer networks, databases, and software engineering.
2. Students' proficiency in a variety of programming languages and tools will allow them to create and develop software applications for a range of platforms.
3. Analytical abilities: Pupils will be able to evaluate complicated issues and create workable solutions using the proper resources and methods.
4. Students will have good oral and written communication skills, which will enable them to work well in teams and as leaders.
5. Professional aptitude: Students will be prepared to work in a variety of computer applications fields, including system analysis, software development, database management, web development, and network administration.
6. Students will acquire a zest for learning and the ability to adapt to new technologies and tools, which will help them stay current in a profession that is continually expanding.

In general, the BCA program aims to develop graduates who have the knowledge, abilities, and characteristics required to thrive in the computer applications business and make major contributions to the discipline.

Program Specific Outcomes:

PSO 1: Graduates will be able to apply their knowledge of mathematics, computer science, and programming to develop software applications that meet user requirements.

PSO 2: Graduates will be able to analyze, design, develop and implement database systems, web applications, and software projects using various technologies and tools.

PSO 3: Graduates will be able to communicate effectively and work collaboratively in a team environment to solve complex problems related to software development.

PSO 4: Graduates will be able to demonstrate proficiency in the use of programming languages, software engineering tools, and techniques to create efficient, scalable, and secure software systems.

PSO 5: Graduates will be able to apply their understanding of ethical and professional responsibilities to make informed decisions in software development projects.

PSO 6: Graduates will be able to identify and analyze emerging trends and technologies in the field of computer applications and use them to solve real-world problems.

PSO 7: Graduates will be able to apply their knowledge of computer applications to diverse fields such as business, healthcare, education, and entertainment.

Programme Objectives:

- a) To facilitate students' acquisition of a comprehensive understanding of computer science fundamentals, including programming languages, algorithms, data structures, database systems, computer networks, and operating systems.
- b) To enable students to enhance their technical skills, focusing on their ability to develop, test, and deploy software applications using various programming languages and software tools.
- c) To cultivate students' problem-solving skills, particularly in analyzing complex problems, identifying relevant solutions, and implementing those solutions effectively.
- d) To foster students' communication and teamwork skills, emphasizing their ability to collaborate effectively in teams, articulate technical concepts clearly and concisely, and engage in cooperative project work.
- e) To prepare students for employment opportunities in the IT sector by providing them with the knowledge, skills, and practical experience necessary to succeed in a range of roles within the industry

Career Prospects:

Graduates with a Bachelor of Computer Applications (BCA) degree have a multitude of career pathways within the IT industry. Some of them are highlighted below:

- Computer Assistant
- Technical Assistant
- Data Entry Operator
- Graphics Designer
- Web Designer
- Web Developer
- Software Developer/Programmer
- Data Manager/ Database Administrator
- System Analyst
- IT Manager
- Network Administrator
- Information Security Analyst
- Researcher or Research Assistant

Programme Structure

Bachelor of Computer Applications is a degree program that focuses on computer applications and other essential skills for the dynamic field of Information Technology. A student may opt for three (3) years and four (4) years program and comprises of two semesters, Autumn Semester and Spring Semester annually. Each student will be required to take a total of 120 or 160 credits depending on duration of the program opted. Credit distribution is highlighted below

Year	Autumn Semester	Spring Semester
1 st Year	1 st Semester	2 nd Semester
2 nd Year	3 rd Semester	4 th Semester
3 rd Year	5 th Semester	6 th Semester
4 th Year	7 th Semester	8 th Semester

Course Outline for Bachelor of Computer Applications

	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
Semester I	BCA 100	Discrete Mathematics and Aptitude	2	-	30
		Discrete Mathematics and Aptitude (Online)	2	-	30
	BCA 101	PC Software and Hardware	1	3	105
	ENGL100	Ability Enhancement Course: English and Communication 1	4	-	60
	SDG 100	Interdisciplinary Course: SDG theme 1: Enhancing Quality of life	1	1	45
	SDG 101	SDG theme 2:	1	1	45

		Towards a Sustainable future			
	FDC 100	Value Added Course:	1	-	15
	FDC 101	Foundation Course 1	1	-	15
		Foundation Course 2			
		Special Event (university week /commemorative days) Introduction to AI		2	60
		Total	13	7	405

	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
Semester II	BCA 150	Concepts of Algorithms & Programming	2	2	90
	BCA 151	Digital Logic	2	-	30
		Digital Logic (Online)	2	-	30
	ENGL150	Ability Enhancement Course: English and Communication 2	4	-	60
	SDG 150	Interdisciplinary Course: SDG theme 3: Equality and Equity (Social, Economics, Gender and Education)	1	1	45
	SDG 151	SDG theme 4: Language, Culture and Music	1	1	45
	FDC 150	Value Added Course: Foundation Course 5: Music and Art Appreciation	-	1	30
	FDC 151	Foundation Course 6: Gender, reproductive health and life skills education	1	-	15
	FDC 152	Foundation Course 7: Human values	1	-	15
	FDC 153	Foundation Course 8: Community Engagement	-	1	30
		Total	14	6	390

	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
	BCA 200	Fundamentals of Operating System	2	-	30

Semester III		Fundamentals of Operating System (Online)	2	-	30
	BCA 201	Object Oriented Programming	1	2	75
		Object Oriented Programming (Online)	1	-	15
	MNR 200 MNR 201 MNR 202 MNR 203	Minor (any One of the following) 1. Education and teaching I 2. Media and communication I 3. Nutrition, sports and fitness I 4. Management and Office administration-I	2	2	90
	MTDP 200	Multidisciplinary 1: Human biology & basic life support	2	2	90
	ENTR 200	Skill enhancement courses: Entrepreneurship	2		30
		Special Event (university week /commemorative days)		2	60
		Total	12	8	420

	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
Semester IV	BCA 250	Programming with Java	1	3	75
		Programming with Java (Online)	2	-	30
	BCA 251	Database Management System	-	3	90
		Database Management System (Online)	1	-	30
	BCA 252	Data structure	2	2	90
	MNR 250 MNR 251 MNR 252 MNR 253	Minor (any One of the following) 1. Education and teaching II 2. Media and communication II 3. Nutrition, sports and fitness II 4. Management and Office administration-II	2	2	90
		Skill enhancement courses: Entrepreneurship	2		30
		Total	10	10	420
		Subject Title	Credits		Hours

Semester V	Subject Code		Theory	Practicum	
	BCA 300	Web Design with Scripting Language	1	3	105
	BCA 301	Computer Networks	2	-	30
		Computer Networks (Online)	2	-	30
	BCA 302	Fundamental of Cloud Computing	2	-	30
	BCA 303	Software Engineering	4	0	60
	MNR 300 MNR 301 MNR 302 MNR 303	Minor (only One of the following) 1. Education and teaching III 2. Media and communication III 3. Nutrition, sports & fitness III 4. Management and Office administration-III	2	2	90
		Special Events (university week, research week ,commemorative days)		2	60
		Total	13	7	405

Semester VI	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
	BCA 351	Python Programming	1	2	75
		Python Programming (Online)	1	-	15
	BCA 352	Internship	-	16	480
		Total	2	18	570
Total credits & hours from Semester I to Semester VI = 120 credits & 2610 hours.					

Semester VII (Honours)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
		1.Web-Technology			
	BCA 400	Web Services	1	3	105
	BCA 401	PHP	2	4	150
		PHP (Online)	2	-	30
	BCA 402	Analysis and Design of Algorithm	2	-	30
		Analysis and Design of Algorithm (Online)	2	-	30
	BCA 403	Advance Database Management System	-	2	60

		Advance Database Management System (Online)	2	-	30
		Total	11	9	435

Semester VII (Honours)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
		2.Hardware and Networking			
	BCA 404	Basic Electronics	1	3	105
	BCA 405	Windows Server Administration	-	4	120
	BCA 406	Wireless Network Security	2	-	30
		Wireless Network Security (Online)	2	-	30
	BCA 402	Analysis and Design of Algorithm	2	-	30
		Analysis and Design of Algorithm (Online)	2	-	30
	BCA 403	Advance Database Management System	-	2	60
		Advance Database Management System (Online)	2	-	30
		Total	11	9	435

Semester VII (Honours)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
		3.Mobile Application			
	BCA 407	Responsive web designing	1	3	105
	BCA 408	Android Programming	2	4	150
		Android Programming (Online)	2	-	30
	BCA 402	Analysis and Design of Algorithm	2	-	30
		Analysis and Design of Algorithm (Online)	2	-	30
	BCA 403	Advance Database Management System	-	2	30
		Advance Database Management System (Online)	2	-	30
		Total	11	9	435

		Subject Title	Credits	Hours
--	--	----------------------	----------------	--------------

Semester VIII (Honours)	Subject Code		Theory	Practicum	
	BCA 450	Project/Internship	-	16	480
	BCA 451	Theory of Computation	4	-	60
		Workshops/Activities	-	-	330
		Total	4	16	840
Total credits & hours from Semester I to Semester VIII = 160 credits & 3880 hours.					

Semester VII (Honours)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
		4.Animation and Game Art			
		Design Fundamentals (Online)	1	3	105
		Graphic Design (Online)	1	3	105
	BCA 409	Classical animation (2D Animation)	-	4	120
	BCA 410	Game Design and Development Technique	-	4	120
	BCA 411	Folklore of North East India	4	-	120
		Total	6	14	570

Semester VIII (Honours)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
		Animation and Game Art			
	BCA 450	Project/Internship	-	16	480
	BCA 452	Research Methodology for Animation and Game Arts	1	-	15
		Research Methodology for Animation and Game Arts (Online)	3	-	45
		Workshops/Activities		-	255
		Total	4	16	795
Total credits & hours from Semester I to Semester VIII = 160 credits & 3975 hours.					

Semester VII (Honours with Research)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
	BCA 412	Research Methodology	1	-	15
		Research Methodology (Online)	3	-	45
	BCA 402	Analysis and Design of Algorithm	2	-	30
		Analysis and Design of Algorithm (Online)	2	-	30
	BCA 403	Advance Database Management System	1	3	105
	BCA 413	Data Analysis using MS-Excel & SPSS Software	1	3	105
	BCA 414	Data Science	2	2	90
		Total	12	8	420

Semester VIII (Honours with Research)	Subject Code	Subject Title	Credits		Hours
			Theory	Practicum	
	BCA 453	Research Work	-	16	480
	BCA 451	Theory of Computation	4	-	60
		Workshops/Activities	-	-	345
		Total	4	16	885
Total credits & hours from Semester I to Semester VIII = 160 credits & 3915 hours.					

Pedagogy

Lectures, case studies, field visits, quiz, article review, focus group discussion, interactive sessions, workshop, seminars, debates, presentation, assignment, article analysis, projects

Semester I

Course: DISCRETE MATHEMATICS & APTITUDE

Course Code: BCA100

Credits: 4

Mode: Offline/Online

About the Course/Subject: Discrete Mathematics is a branch of mathematics that deals with separable and distinct numbers. Combinations, graph theory, and logical statements are included, and numbers can be finite or infinite. It's used in computer science to design the apps and programs we use every day. Math Aptitude Tests help to assess your skills in the following areas: (1) arithmetic, (2) pre-algebra, (3) algebra, (4) word problems, and (5) number series. Passing grades vary

Learning Objectives:

- To make students understand the basic concepts of discrete mathematical structure like set, relations, functions, propositional logics.

Learning Outcomes:

- Analyze logical propositions via truth tables.
- Prove mathematical theorems using mathematical induction.
- Understand sets and perform operations and algebra on sets.
- Determine properties of relations, identify equivalence and partial order relations, sketch relations.
- Identify functions and determine their properties.
- Define graphs, digraphs and trees, and identify their main properties.
- Evaluate combinations and permutations on sets.
- To provide foundation and concepts related to mathematical skills and knowledge for understanding the basic rules of mathematics.
- It is an undeniable advantage that everyone is aware of. It has become a fundamental part of our aptitude and logic that ensures mental discipline.

Course content:

UNIT-I

Sets – Brief review of basics in set theory such as ways of describing a set, Finite and Infinite Set, Set Operation, Union, Intersection of Set, Complement of Sets, Empty Set, Disjoint Set, De Morgan's Law, Power Sets Cartesian Product, Simple Applications

Relations and functions-properties of relations, equivalence relation, partial order relation

function: domain and range, onto, into and one to one functions, composite and inverse functions

UNIT-II

Boolean algebra-definition and properties of boolean algebra, a brief introduction to the application of boolean algebra to switching theory, conversion of complicated switching circuits to simple one, disjunctive and conjunctive normal forms. Functions- characteristic function, composition of

functions, binary and n-ary operations, hashing function, recursive functions. Algebraic systems: semigroups and monoids, groups, subgroups, normal subgroups and quotient groups, cyclic groups, homomorphism and isomorphism.

UNIT-III

Arithmetic Ability – BODMAS rule, Number, LCM and HCF, Fraction, Simplification, Square and cube root, Average, Problem on ages, Surd and indices, Percentage, Profit and loss, Ratio and proportion, partnership, work and wages, Pipe and cistern, time and distance, Simple & compound interest, area, volume and surfaces, clock & calendar. Statistics: Measure of dispersion; mean, median and mode.

UNIT-IV

Mathematical Reasoning - Mathematically acceptable statements. Connecting words/phrases - consolidating the understanding of “if and only if (necessary and sufficient) condition”, “implies”, “and/or”, “implied by”, “and”, “or”, “there exists” and their use through variety of examples related to real life and Mathematics, Syllogism.

Pedagogy: Lectures, Demonstration, Whiteboard problem solving, Assignemnt

References/Suggested Reading:

- C.L.Liu, ” Elements of Discrete Mathematics”, McGraw Hill
- Trembley, J.P & R. Manohar,” Discrete Mathematical Structure with Application to Computer Science”, TMH
- Doerr Alan &Levasseur Kenneth, “Applied Discrete Structures for Computer Science”, Galgotia Pub. Pvt. Ltd
- SemyourLipschutz& Marc Lipson, “Discrete Mathematics”, Second Edition, Schaum’s Outlines, Tata McGraw-Hill Publishing
- R. D Sharma, “Mathematics Vol. 1 & 2”, Dhanpat Rai Publications; 2017 edition
- R.S Aggarwal, “Quantitative Aptitude, S Chand Publications; 20th edition 3. R.V. Praveen,"Quantitative Aptitude and Reasoning", PHI

Evaluation Criteria:

- Written Test, Assignments, Class Presentation, Calculation

Course Name: PC SOFTWARE AND HARDWARE

Course Code: BCA101

Credits: 4

Mode: Offline

Learning Objective:

This course intends to familiarize the students with MS Office and its applications and PC assembling

Learning Outcomes:

- Students will be able to identify and describe the basic components of a computer system, including the motherboard, CPU, RAM, hard drives, power supply, cooling systems, input/output devices, and peripherals.
- Students will be able to install and configuring hardware parts, like as expansion cards and peripherals, and being familiar with BIOS settings.
- Students will be able to use Microsoft Word to create, edit, and format documents. You can add styles, tables, and graphics to improve the look and functionality of your work.
- Students will be able to use Microsoft PowerPoint, create and deliver engaging presentations that incorporate multimedia components including photos, audio, and video.
- Students will be able to utilise Microsoft Excel to manage and organise data, using formulas and functions for calculations and charts and graphs for visual representation.
- Students will be able to utilise Microsoft Outlook to successfully manage tasks, schedule meetings, and manage emails as well as calendars and contacts.

Course Content:

1. Introduction to MS Word and its area of use
2. Identify the different components of the document window and their functions
3. Creating and saving a document, open an existing file and saving a file using a new name
4. Protecting the document window using a password
5. Document creation- text selection and editing, cut, copy, paste, finding and replacing text.
6. Formatting the text- font and size selection, alignment and spacing of text, paragraph indenting, bullets & numbering, headers & footers and changing case.
7. Working with themes, table of contents, watermark, margins, size and orientation of page
8. Working with hyperlink, columns, drop cap, page setup, print preview and printing of documents
9. Working with tables – insert table, changing cell width & height, alignment of text in cell, insert/delete rows and columns , merging & splitting of cells
10. Working with pictures – picture style, aligning, text wrapping & cropping
11. Working with mail merge
12. Introduction to MS PowerPoint and its area of use
13. Identify the different components of the PowerPoint window and their functions
14. Creating and saving presentation, open an existing presentation and saving it using a new name and protecting the presentation using a password and working with slides – insert, delete and copying of slides
15. Working with themes – color fonts & effects, slide design, background styles, animation and transition effects, setting the slide timer.

16. Working with tables, hyperlinks, insert textbox, slide number, header & footer, wordart
17. Creating a photo album, picture and clipart and working with media clips- insert movie and sound clips.
18. Changing page setup, slide orientation, printing a presentation and running the presentation using the slide show and function key.
19. Introduction to MS Excel and its area use
20. Identify the components of the excel window and their functions
21. Understanding what is a workbook, worksheet, cells, range and auto fill handle Open, save, close and renaming a workbook and protecting the workbook using a password.
22. Inserting worksheets, copying and renaming sheets, deleting sheets, editing text, selecting cells, rearranging & merging of cells contents and working with cell formatting using auto row format, row and column formatting , cell border, hyperlink.
23. Working with formula using addressing method, auto sum and functions, merging from excel workbook to word document.
24. Managing data-sorting data, filtering data, freezing rows & columns, cell contents, working subtotals and data form and working with charts.
25. Understanding the different components of a desktop computer.
26. Understanding the different brands of the components.
27. Assembling a computer
28. Partitioning and Installing operating system and drivers
29. Installing application software
30. Troubleshooting RAM, hard drives, SMPS problems

Pedagogy: Lectures, Demonstration, Assignment, practical work.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Practical work

Semester II

Course Name: CONCEPTS OF ALGORITHMS & PROGRAMMING

Course Code: BCA150

Credits:4

Mode: Offline

Learning Objective:

To provide fundamental computational concepts underlying in computer programming languages using C programming language

Learning Outcomes:

To enable students to implement Programs with pointers and arrays. perform pointer arithmetic" and use the pre-processor.

To enable the students to understand the simple algorithm and flowchart of the program.

To write programs that perform operations using derived data types.

To enable students implementing coding in C programming language to develop an applications.

Course content

UNIT-I

Introduction to Programming-Computer Programming, Programming Technique, Procedural Programming ,Object Oriented Programming; Design of Algorithm- Definition ,Features of Algorithm, Development of Algorithm for simple problems; Flowcharts-Definition, Features of Flowchart, Basic Symbols used in Flowchart, Development of Flowchart for simple problem; Fundamentals of C Programming - character set, Keywords, identifier, Datatypes, statement, Symbolic constant; Input/ output statement- getchar, putchar, scanf, printf, gets, puts; Operators and expression – arithmetic, relational , unary, logical, assignment operator, conditional operator

UNIT-II

Control statement- if statement, if – else statement, nested if – else statement.. Loop control structure: while, do – while, for, switch, break, continue, goto. Array: single and multi-dimensional array, array declaration and initialization; Strings - declaration, initialization, standard library string functions

UNIT-III

Functions-Need and definition, user defined and library function, declaration and prototype, function arguments, return values and nesting of function, calling of function, recursion

UNIT-IV

Structures: Structure declaration, accessing structures elements, nested structures, array of structures, uses of structures. Unions, unions of structures; Pointers- Introduction to pointers, Dynamic memory allocation; Files- fopen(), fclose(), fseek()

UNIT- V

Graphics Programming:Library file- graphics.h, 2-D Coordinate system, Simple Graphics Functions(initgraph(), line(), circle(), arc(), rectangle(), ellipse(), drawpoly(), closegraph(), restorecrtmode(), setfillstyle(), putpixel(), getmaxx(), getmaxy(), outtextxy(), setcolor(), fillcolor(), settextstyle(), moveto(), lineto(), moverel(), linerel()) Pallete and color, Animation functions(imagesize(),getimage(),putimage())

References/Suggested Reading:

- 1. E. Balagurusamy, “Programming in C”, TMH Publications
- 2. Peter Juliff, “Program design”, PPH Publications
- 3. E. Balagurusamy, “Programming in C++”, TMH Publications.
- 4. Yashavant Kanetkar, “Let Us C”, BPB publications
- 5. S.K Basandra, “Computers Today”, Galgotia Publication
- 6. Gottfried, B. S., “Theory and Problems of Programming with C”, New Delhi: Tata McGraw-Hill Publication, 1997

Evaluation Criteria:

- Written Test, Case Study, Assignments, Class Presentation, Practical programming implementation

Course name: DIGITAL LOGIC

Code: BCA151

Credits: 4

Mode: Offline/Online

Course Objectives:

- (i) To provide basic knowledge for design of digital electronic circuits.
- (ii) To provide understanding for the operation of digital computers and design associated with computer hardware.

Course Content:

UNIT- I

Binary Systems – Binary Numbers, Number Base Conversion, Octal and Hexadecimal Numbers, Complements, Signed Binary Numbers, Binary Codes, Binary Storage and Registers

UNIT-II

Logic Gates, Boolean algebra, Map Simplification: Two Map Method, Two and Three Variable Maps, Four-Variable Map, Product of Sums Simplification

UNIT-III

Combinational Circuits (Half -Adder, Full-Adder, Binary Parallel Adder, BCD Adder, Universal Property of NAND and NOR gates, Combinational Circuits using NAND and NOR gates); Flip flops (SR, D, JK, T, Master Slave, Edge-Triggered, Excitation Tables);

UNIT-IV

Sequential Circuits (Latches, Flip-Flop Input Equations, State Table, State Diagram, Design Example, Design Procedure), Integrated Circuits (Digital Logic Families and Integrated Circuits); Decoders (NAND Gate Decoder, Decoder Expansion, Encoders); Multiplexes (4 to 1 Line Multiplexer, Data Selector); Demultiplexer; Code Converter; Registers (Register with Parallel Load); Shift Registers (Bidirectional Shift Registers with Parallel Load, Serial Register); Binary Counters (Binary Counter with Parallel Load, Ripple Counter); Memory Unit (Random-Access Memory, Read-Only Memory, Types of ROMs)

References/Suggested Reading:

1. M. Morris Mano, "Digital Logic and Computer Design", Prentice Hall of India Pvt. Ltd
2. P. Pal Choudhuri, "Computer Organization and Design", Prentice Hall of India Pvt. Ltd
3. M. Morris Mano, "Computer System Architecture", Prentice Hall of India Pvt. Ltd

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignemnt.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion

Semester III

Course name: FUNDAMENTAL OF OPERATING SYSTEM

Course code: BCA200

Credits: 4

Mode: Offline/Online

Learning Objectives

- To describe the major workings of an operating system, their functions and purpose to achieve a knowledge foundation of system software's functioning's and behaviors.

Learning outcomes

- Understanding the basic concepts and functions of an operating system, including process management, memory management, file systems, and input/output operations.
- Ability to analyze and evaluate different types of operating systems, including batch processing, time-sharing, and real-time systems.
- Familiarity with common operating system commands and utilities, and ability to use them effectively.

Course content

UNIT-I

Introduction – Definition, Types of Operating System, Functions of the Operating System, Operating Systems Services, System Calls, Single User, Multi User and Multitasking Operating System

UNIT-II

Process Management – Process, Scheduling, CPU Scheduling Concepts, Process Synchronization, Semaphore, Classical Problems of Synchronization, Deadlocks, Deadlock Detection, Deadlock Recovery.

UNIT-III

Memory Management – Introduction, Logical address V/s Physical address, Swapping, Contiguous Allocation: Partitions, Fragmentation, Paging, Segmentation. Virtual Memory: Page Replacement, Page Replacement Algorithms.

UNIT-IV

File Management – File concepts, Access Methods, File System Mounting, File System Implementation, Partitions and Mounting.

Pedagogy

Lectures, Discussion, Assignments

Suggested readings:

- Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Principles”, Eighth-Tenth Edition, Wiley-India Publication.
- Stalling, W., “Operating system, “Sixth Edition, Prentice Hall (India).

- Sibsankar Halder and Alex A. Aravind, “Operating Systems”, Pearson Education.

Evaluation criteria:

Written Test, Assignments, Presentation.

Subject: OBJECT ORIENTED PROGRAMMING IN C++

Code: BCA201

Credits: 4

Mode: Offline/Online

Learning Objective:

- To provide fundamental concepts in Object Oriented Design. To be able to write and maintain C++ programs and to be able to use the advanced features of C++ programming language.
- To provide fundamental concepts in Object Oriented Design. To be able to write and maintain C++ programs and to be able to use the advanced features of C++ programming language.

Learning Outcome:

- Understanding of core OOP ideas: Students will gain knowledge of core OOP ideas such classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
- Ability to develop and implement OOP programmes: Students will learn how to design and implement object-oriented programmes using C++, including the creation of classes, instantiating objects, and implementing inheritance and polymorphism.
- Understanding of the syntax and features of the C++ programming language: Students will become proficient in the language's syntax as well as its data types, control structures, functions, and pointers.
- Students will gain knowledge of software development best practises, including modular programming, code reuse, testing, debugging, and documentation.
- Students will learn how to use OOP principles and techniques to address complicated programming challenges such as designing software systems, developing algorithms, and creating data structures.

Course content:

UNIT-I

Overview of C++, Object-Oriented programming Paradigm, Basic Concepts of Object-Oriented programming, Benefits of OOPs, Application of OOP, C++ Statement, Class, Structure of C++ Program, Creating the Source file, Compiling and Linking.

UNIT-II

Basic data types, User Defined Data types, Derived Data types, Declaration of Variables, Dynamic Initialization of Variables, Scope Resolution Operator, Member Deferencing Operators, manipulators, Type Cast Operator, Expression and Implicit Conversion, Operators, Loops and Decisions.

Specifying a Class, Defining Member Functions, making an outside Function Inline, Nesting a member function, Array within a Class, memory Allocation for Objects, Arrays of Objects Object as Function Arguments.

UNIT-III

Constructors and Destructors: Introduction, Constructors, parameterized Constructors, Multiple Constructors with Default Arguments, Dynamic Initialization of Objects. Copy Constructors. Dynamic Constructors, Constructing Two-dimensional Arrays, Destructor.

Functions in C++: The Main Function, Function Prototyping, Call by reference, Return by Reference, Inline Functions, Function Overloading, Friend and Virtual Function, Pure Virtual Function.

UNIT-IV

Operator overloading and type conversions: Introduction, Defining Operator Overloading, Overloading Unary Operators, Overloading Binary Operators Using Friends, Manipulation of strings using Operators, Rules for Overloading Operators, Type Conversion.

UNIT- V

Inheritance: extending class – Introduction, Defining Derived Classes, Single Inheritance, making a Private Member Inheritable, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Pointers, Virtual functions and polymorphism Pointers to Objects, This Pointers, Pointers to derived Classes.

LIST OF PROGRAMS

1. Program to demonstrate Basic C++ program
2. Program to implement Functions overloading
3. Program to implement Classes and objects
4. Program to implement Inline functions and Friend functions
5. Program to implement Constructor
6. Program to implement Destructor
7. Program to implement Operator Overloading
8. Program to implement Single inheritance and multilevel inheritance
9. Program to demonstrate Virtual function
10. Program to demonstrate the concepts of Polymorphism

Pedagogy

Lectures, Discussion, Assignments, hands-on practice, Lab work

Suggested readings:

1. E. Balagurusamy, “Programming in C++”, TMH Publications
2. Sarang poornachandra, “Object – Oriented programming with C++”
3. Anirban Das, Goutam Panigrahi, “Object – Oriented programming with C++”
4. Robert Lafore, “Object – Oriented programming in C++”
5. Graham M. Seed, “ An Introduction to Object – Oriented programming in C++”

Evaluation criteria:

Written Test, Assignments, Presentation, Practical work.

Semester IV

Course name: PROGRAMMING WITH JAVA

Course code: BCA250

Credits: 4

Mode: Offline/Online

Learning Objectives

- To identify Java language components and how they work together in applications
- To design and program stand-alone Java applications
- To learn how to implement object-oriented designs with Java
- To learn how to use exception handling in Java applications

Learning Outcome:

- To adapt to changes in environment and to implement advances in the art of programming.
- To use the syntax and semantics of java programming language and basic concepts of object-oriented programming
- To use java programming language for writing well-organized, complex computer programs with both command line and graphical user interface
- To design and develop application based on the concept of java
- To design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and event handling

Course content

UNIT-I

Basic concepts of OOP-Benefits and Applications of OOP; Java Evolution -Java History and Features, Difference of Java from C and C++, Java and Internet, Java and WWW, Web browsers, Hardware and Software requirements, Java Support systems, Java Environment; Overview of Java Language-Simple java Program, An application with two classes, Java Program Structure, Java Tokens, Java Statements, Implementing a Java Program, JVM, Command Line Arguments, JIT, Bytecodes

UNIT-II

Constants, Variables, Arrays and Data Types; Operators and Expressions- Arithmetic, Relational, Logical, Assignment, Increment and Decrement, Conditional, bitwise Operators. Arithmetic Expression and Evaluation, Conversion and Casting Decision Making and Branching - If statement If...Else Statement-Nested If...Else statement- Switch Statement- Conditional Operator; Decision making and Looping-While, do, for Statements, Jumps in loops

UNIT-III

Classes, Objects, Methods; Inheritance; basic swings; Interfaces: Multiple Inheritance; Packages Putting Classes together; Introduction to Multithread Programming, Multi-threading, Thread Life cycle, Multi-threading advantages and issues, Simple thread program, Thread synchronization.

UNIT-IV

Managing Errors and Exceptions; I/O Basics, Reading Console Input, writing Console Output; Networking

UNIT-V

GUI: Introduction to AWT programming, Layout and component managers, Event handling, Applet class; Applet life cycle, Passing parameters, embedding in HTML, Applet Basics and Applet Programming; Graphics programming; Swing components, JApplet, JButton, JFrame, etc. Sample swing programs, The Java Library: Strings

Practical

1. To implement simple program based on operator loop decision statements
2. To implement Program to define Class and instantiate Objects
3. Program to implement constructor and Method overloading and Method overriding
4. Program to create components using Swing
5. Program to implement Wrapper Class and command line argument
6. Program to demonstrate packages and interfaces
7. Program to demonstrate Single level and Multi level inheritance
8. Program to demonstrate Exception Handling
9. Program to demonstrate Multithreading and Synchronization
10. Program to implement Server and client using networking
11. Program s using Applet Class
12. Program to perform String Class and String Buffer Class

Pedagogy

Lectures, Discussion, Assignments, hands-on practice

Suggested readings:

- Herbert Schildt, “The Complete Reference Java 2”, Fifth Edition, Tata McGraw Hills Publishing Company Limited.
- E Balagurusamy, “Programming with Java- A Primer”, Third Edition, Tata McGraw Hill Publishing Company Limited.
- John Hubbard, “Programming with Java”, Schaum’s Outlines, Tata McGraw Hill Publishing Company Limited.
- Cay S. Horstmann, “Core Java, Volume I : Fundamentals (English)”, 9th Edition
- Joyce Farrell, “Java Programming”, Seventh Edition

Evaluation criteria:

Written Test, Assignments, Practical Test

Course name: DATABASE MANAGEMENT SYSTEM

Course code: BCA251

Credits: 4

Mode: Offline/Online

Learning Objectives

- The objective of this Course is to introduce to the students the fundamental concepts necessary for designing, using and implementing database systems and applications. The Course stresses on database modelling and design, physical file storage techniques and language facilities provided by database management systems

Learning Outcomes:

- Understand the fundamental concepts of database management systems, including data models, relational algebra, SQL queries, and normalization.
- Design, create, and manipulate databases using a DBMS software package, such as Oracle, MySQL, or Microsoft SQL Server.
- Apply knowledge of database theory and database design to real-world scenarios, such as creating a database schema for an online store or designing a database for a healthcare organization.
- Develop skills in data manipulation and query optimization, including the ability to write efficient and effective SQL queries.
- Design and implement a database schema: Students should be able to design a database schema using entity-relationship diagrams (ERD) and translate it into a SQL schema definition. They should also be able to create the necessary tables and populate them with sample data.
- Query data from a database: Students should be able to write basic and complex SQL queries to retrieve data from a database. This includes using aggregate functions, grouping, joining tables, and subqueries.
- Modify and update a database: Students should be able to modify the data in a database using SQL statements such as INSERT, UPDATE, and DELETE. They should also be able to create and modify views and indexes.
- Manage database transactions: Students should understand the concept of database transactions and be able to use SQL statements such as COMMIT, ROLLBACK, and SAVEPOINT to manage them. They should also be able to use transactions to ensure data integrity and consistency.

Course content:

UNIT-I

Overview of the database management system [1]- Database systems, Need for Database ,Advantages of using a database, Characteristics of data in a database , Functions of DBMS, Data abstraction, Data independence, Overall Architecture of DBMS, Three level architecture; Data Models[1]- Relational Data Models, ER Model, Hierarchical models, Networking models, Advantages and Disadvantages

of each models

UNIT-II

Entity Relationship model: Components, Symbols, Class and Objects, Attributes; Specialization – Aggregation; Relational Model [1]– Characteristics of Relational Database Model, CODD's rules, Tables, Rows, Columns, Domains, Attributes, Candidate Key, Primary Key, Foreign Key, Super Keys, Unique Keys, Constraints; Normalization[1] -Purpose of Normalization, Functional Dependence, Relational database Design, Normal forms, 1NF, 2NF, 3NF, BCNF, 4NF

UNIT-III

Introducing MySQL [2] –History, Role of MySQL in industry, Version of MySQL, Architecture, Engines; MySQL queries[2]- Data types, operators, functions; Working with Databases and Tables Creating, Copying, Modifying Tables

UNIT-IV

MySQL Advance [2]-Show commands, Working with date and Time data types, Joins like Cross, Inner, Outer, Self, Unions, Subquery, Procedure, Triggers, Views,index, MySQL database export and import

UNIT-V

Database Backup and Recovery: Hardware Protection and Redundancy; Transaction Logs; Importance of Backups; Database recovery; Data storage; Causes of failures; Concurrency Control; Database Security and Integrity

• LIST OF PROGRAMS:

1. Working on MySQL DDL, DML, DTL Basic Data Types
2. Table Constraint definition Commands to create table
3. Commands for table handling Alter table, Drop table, Insert records
4. Commands for record handling Update, Delete Select with operators like arithmetic, comparison, logical Query Expression operators Ordering the records with order by Grouping the records
5. MySQL functions Date, Numeric, Character, conversion Group functions avg, max, min, sum, count
6. Set operations Union, Union all, intersect, minus
7. Join concept Simple, equi, non equi, self, outer join
8. Query & sub queries
9. Working on View Intro, create, update, drop
10. Working with index
11. Primary introduction to User creation, granting privileges (Grant, Revoke, Commit, Rollback, savepoint)
12. Write a query in Mysql to create a table employee and department.

a. Employee (empno, ename,deptno,job,hiredate)

b. Department(deptno,dname,loc)

Include the following constraints on column of emp table.

a. to make the empno as primary key of the table and

b. to ensure that the ename column does not contain NULL values and

c. the job column to have only UPPERCASE entries and

d. to put the current date as default date in hire date column in case data is not supplied for the column.

Include the following constraints on column of dept table.

a. to make deptno as primary key.

b. to ensure dname, loc columns does not contain NULL values

Also enforce REFERENTIAL INTEGRITY, declare deptno field of dept table as primary key and deptno field of emp table as foreign key.

13. Working on MySQL DDL, DML, DTL Basic Data Types

14. Table Constraint definition Commands to create table

15. Commands for table handling Alter table, Drop table, Insert records

16. Commands for record handling Update, Delete Select with operators like arithmetic, comparison, logical Query Expression operators Ordering the records with order by Grouping the records

17. MySQL functions Date, Numeric, Character, conversion Group functions avg, max, min, sum, count

18. Set operations Union, Union all, intersect, minus

19. Join concept Simple, equi, non equi, self, outer join

20. Query & sub queries

21. Working on View Intro, create, update, drop

22. Working with index

23. Primary introduction to User creation, granting privileges (Grant, Revoke, Commit, Rollback, savepoint)

24. Write a query in Mysql to create a table employee and department.

a. Employee (empno, ename,deptno,job,hiredate)

b. Department(deptno,dname,loc)

Include the following constraints on column of emp table.

e. to make the empno as primary key of the table and

f. to ensure that the ename column does not contain NULL values and

g. the job column to have only UPPERCASE entries and

h. to put the current date as default date in hire date column in case data is not supplied for the column.

Include the following constraints on column of dept table.

c. to make deptno as primary key.

d. to ensure dname, loc columns does not contain NULL values

Also enforce REFERENTIAL INTEGRITY, declare deptno field of dept table as primary key and deptno field of emp table as foreign key.

Pedagogy

Lectures, Discussion, Assignments

Suggested readings:

- Abraham Silberschatz- Henry K. Korth- S. Sudarshan, “Database System Concepts”, 4th edition, McGraw Hill International Edition
- VikramVaswani,”MySQL (TM): The Complete Reference”, McGraw Hill Education Publication
- Madhilika Jain- Vineeta Pillai- Shashi Singh- Satish Jain, “A Level- Introduction to Database Management Systems”, BPB Publications
- R S Gill, “Database Management System”, I K International
- R Elmasri and S B Navathe,” Fundamentals of Database Systems”, Pearson Publication
- 6. G. K. Gupta, “Database Management System”, Tata McGraw Hill Publication

Evaluation criteria:

Written Test, Case Study, Assignments, Class Presentation

Course name: Data Structure

Course code: BCA252

Credits: 4

Mode: Offline

Learning Objectives

- To understand the implementations of algorithms, their efficiencies and to learn the fundamental components of problem solving by designing a method of organizing large amounts of data in an effectively solvable manner

Learning outcomes

- To enhance knowledge about implementation of abstract data types using arrays and a linked list.
- To apply the different linear data structures like stack and queue to various computing problems
- To learn different types of trees and apply them to problem solutions
- To discuss graph structure and understand various operations on graph and their applicability
- To analyze the various sorting and searching algorithms
- To understand the hashing technique and hash functions
- To implement the different abstract data type using Arrays and Linked list

- To implement different sorting algorithms

Course content

UNIT-I

Introduction – The concept of data structure, Abstract data type, Concept of list & array, Recursion Functions and its implementation; Introduction to Stack – Stack as an abstract data type, primitive operation on stack, Stacks application: Infix, post fix, prefix and recursion, multiple stacks.

UNIT-II

LINKED LIST – Basic operations on linked list, Stacks and queues linked list, Header nodes, Doubly Linked List, Circular Linked List, Application of linked list.

UNIT-III

TREES – Basic Terminology, Binary Trees, Basic operation on Binary tree; Traversal of binary trees – In-order, Pre-order & Post-order, Binary Search Tree and its Applications. GRAPHS – Definition, Terminology, Directed, Undirected & Weighted graph, Representation of graphs, Graph Traversal Depth first & Breadth first search, Spanning Trees, Minimum Spanning Tree, Shortest path algorithm;

UNIT-IV

SORTING – Insertion sort, Selection sort, Bubble sort, Quick sort, Merge Sort, Heap sort, Comparison of sorting methods, Hash Table; Collision resolution Techniques.

LIST OF PROGRAMS:

1. Implementation of Concatenation & length using for
2. Implementation of Comparison & length using for
3. WAP to Access substring
4. WAP to find the Factorial using recursion
5. WAP to find the GCD of a number using recursion
6. WAP to find the Tower of Hanoi using recursion
7. WAP to find the Fibonacci Series using recursion
8. WAP to implement Insertion in an Array
9. WAP to implement Deletion in an Array
10. WAP to perform Binary output
11. WAP to implement Linear Binary & Sort
12. WAP to implement Bubble sort
13. WAP to implement Insertion
14. WAP to implement Select
15. WAP to implement Merge
16. WAP to implement Quick
17. WAP to implement BST & Tracing
18. WAP to Create a Linked list
19. WAP to implement Insertion in a linked list
20. WAP to implement Deletion in a linked list

21. WAP to implement Searching in a linked list
22. WAP to implement Double Linked list
23. WAP to implement Circular Linked list
24. WAP to implement Stack push and pop array
25. WAP to implement Stack Linked list
26. WAP to implement Queue Array and Linked List
27. WAP to implement Double and circular Queue
28. WAP to implement Circular Stack

Pedagogy

Demonstration, hands-on Practice, problem solving, iterative learning, Group projects

Suggested readings:

- Seymour Lipschutz, “Data Structures”, TATA McGraw-Hill
- A. A Puntambekar, “Data structures Using 'C' “, Technical publications
- E. Balagurusamy, “Data Structures Using C “, TATA McGraw-Hill
- Yashavant Kanetka, “Data Structures Through C”, BPB Publication

Evaluation criteria:

Written Test, Case Study, Assignments, Class Presentation, Practical

Semester V

Course name: WEB DESIGNING WITH SCRIPTING LANGUAGE

Course code: BCA300

Credits: 4

Mode: Offline

Learning Objectives

1. To principles of web design, including layout, typography, color theory, and visual hierarchy.

Learning Outcomes:

2. Understanding of the basic principles of web design, including layout, typography, color theory, and visual hierarchy.
3. Ability to create responsive and accessible websites that work on multiple devices and browsers.
4. Knowledge of HTML, CSS, and JavaScript, and ability to use these technologies to create and style web pages.

Course content

1. Creating web page using basic formatting tags: heading, paragraph, underline break, bold, italic, underline, superscript, subscript, font and image; different attributes like align, color, bgcolor, font face, border, size
2. Write HTML code to develop a Web page having the background in red and title "My First Page" in any other color
3. Create an HTML document giving details of your name, age, telephone number, address, TLC code & enrolment number aligned in proper order
4. Write an HTML code to design a page containing text, in form of paragraphs giving suitable heading style
5. Create a page to show different attributes of Font tag
6. Create a page to show different attributes: italics, bold, underline
7. Creating web page having navigation links using anchor tag, internal, external, mail and image links; lists-ordered, unordered
8. Creating web page having table tag; HTML Form controls-form, text, password, textarea, button, checkbox, radio button, select box, hidden controls, Frameset and frame
9. Write an HTML code to create a Web page of blue color and displaylinks in red colour
10. Create a Web page with appropriate content and insert an image towards the left hand side of the page. When user clicks on the image, it should open another Web page
11. Create a Web page, which should contain a table having two rows and two columns.
12. Write an HTML code to develop a Web page having two frames that divide the Web page into two equal rows.
13. Write an HTML code to develop a Web page having two frames that divide the Web page into two equal rows and then divide the second row into two equal columns.

14. Write an HTML code to develop a Web page having frames as described in the above question and then fill each frame with a different background color
15. Design a page with a text box called 'name' and a button with label 'Enter. When you click on the button another page should open with the message "Hello < name >", where name should be equal to the name entered in the first page
16. Writing programs implementing cascading style Sheet (CSS), CSS syntax, comments, id and class, background color, background image- text - text color, text alignment, text decoration, text transformation, text indentation; CSS font - font families, font style, font size - setting text size , using pixels and em; CSS lists - different list item markers, unordered list, ordered list, an image as the list item marker
17. Writing programs implementing CSS tables - table borders, collapse borders, table width and height, table text alignment, table padding, table color; CSS positioning - static positioning, fixed positioning, relative positioning, absolute positioning, overlapping elements, float, horizontal align, image gallery, image opacity/transparency
18. Writing program using Javascript tag, comments, variables, document methods-write and writeln methods, alert; operators-arithmetic, assignment, relational, logical, javascript functions, conditional Statements, loops, break and continue; events familiarization-onLoad, onClick, onBlur, onSubmit, onChange
19. Write a JavaScript code to create a pull down menu box.
20. Write a program to move a text with mouse pointer and to change colour of text randomly
21. Create a Web page using two image files, which switch b/w one another as the mouse pointer moves over the image. Use the On Mouse over and On Mouse out event handler
22. Create an HTML form that has a number of text boxes. The user fills the textboxes with data. Write a script that verifies that all textboxes have been filled. If a text box has been left empty pop up an alert message indicating the box that has been left empty. When OK button is clicked, set focus to that specific textbox. If all the textboxes are filled, display thank you.
23. Working HTML 5 events using javascript-offline, onabort, onafterprint, onbeforeunload, onbeforeprint, onblur, oncanplay, oncanplaythrough, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragcenter, ondragleave, ondragover, ondragstart, ondrop, ondurationchange, onemptied, onended, onerror, onfocus, oninput, oninvalid, onload, onmouseover, onmouseup, onmouse wheel, onpagehide, onpageshow, onplaying, onprogress, onratechange, onredo, onresize, onscroll, onseeked, onseeking, onselect, onsubmit, onsuspend, onundo, onunload, onvolumechange, onwaiting
24. Working with scalable vector graphics-embedding SVG, SVG line, circle, rectangle, ellipse, polygon, gradients; Canvas element-using canvas to draw polygon, path, text, transformation
25. Working with web storage-session storage, local storage, delete web storage; web socket events open, message, error, close; web socket methods-socket.send(), socket.close()
26. Working with Joomla 3.4 CMS-installation, work areas, control panel, -toolbar; menu-content, component, extensions, help menu

27. Creating menus, adding menus items, modifying menu items, submenus
28. Working with Joomla modules-create module, breadcrumb module, feed display module, footer module, search module, random image module, whos is online module, syndicate module
29. Working with Joomla global setting-system setting, media setting, language manager, private messages, mass emailing, cache management, users setting
30. Working with Joomla template-template manager, customize template, adding template, creating, adding,customize logo, category management, adding content, formatting content, article metadata, adding banners, contacts adding news feed, adding forum, web links
31. Working with joomla plugins-plugin mangers, authentication plugins, content plugins, editor plugins, search plugins, users plugins, extension, system plugins
32. Working on Site Management-global configuration- site online and offline, metadata setting, change site url ,updating web site, updating extension, disabling and uninstalling extensions, back up site
33. Web hosting-www, web server, internet service provider, web hosting providers,domain names, web hosting email servers,web hosting technologies and types
34. Working with Cpanel-using file section tools, mange domains, manage email, manage security section, manage databases, manage software section tools

Pedagogy:

Demonstration, Handson, practical implementation, Assignment, Discussion.

Suggested readings:

1. HTML5 and CSS3: Develop with Tomorrow's Standards Today, Hogan Brian P, Springer India Private Limited
2. HTML 5 Foundations, Matt West, Wiley India Pvt Ltd
3. Responsive Web Design with HTML5 and CSS3, Hogan Brian P., Shroff Publishers & Distributers Private Limited – Mumbai
4. HTML 5 and CSS 3 Made Simple, Ivan Bayross, BPB
5. Joomla Accessibility, Joshue O Conner, Shroff Publications

Evaluation criteria:

Written Test, Case Study, Assignments, Class Presentation, Lab Work

Course name: Computer Networks

Course code: BCA301

Credits: 4

Mode: Offline/Online

Learning Objectives

- To understand state-of-the-art in network protocols, architectures and applications, process of networking

Learning Outcomes:

- To understand the different layers of Networking
- To understand the different protocols in the different layers of Network

Course content

UNIT-I

Introduction – Basic Communication Model, Data Communications, Computer Network Criteria, Types of connections; Network topology types – Bus Topology, Ring Topology, Star Topology, Mesh Topology and Tree Topology; Man, Wan, LAN; Goals and Applications of computer networks, Network Functions, Network Hardware, Designs Issues for layers, Interfaces and Services, Connection oriented and Connectionless Services; Introduction to OSI Model – Functions of each layers, The TCP/IP Reference Model, Comparison of OSI and TCP/IP Models, Physical Layer, Digital Signals, Data Transmission Concept; Types of transmission – Wired and Wireless Media and its types, Satellite Networks and its types; Packet Switching; Message Switching; Broadband ISDN;

UNIT-II

Introduction to Data Link Layer ;Services Provided by the Data Link Layer to the Network Layer; Framing; Framing Methods – Character Count and Bit Stuffing; Error Control; Introduction to Error Detection and Correction; Error Detection – Content Error and Flow Integrity Errors; Two Dimensional Parity Check; Cyclic Redundancy Check – CRC generator and CRC checker; Check Sum; Hamming Codes, Flow Control; Sliding Window Protocol; Automatic Repeat Request (ARQ) ARQ techniques – Stop and Wait ARQ, Go BACK –n ARQ and Selective Repeat Request;

UNIT-III

Medium Access Layer – CSMA , CSMA/CD, Collision – Free Protocols; IEEE 802 Standards; Token Bus – IEEE 802.4;Token Ring – IEEE 802.5;Introduction to Network Layer; Routing algorithm – Static Algorithm - Dijkstra's Algorithm, Bellman-Ford routing algorithm, Flooding, Flow Based Routing; Dynamic Algorithm - Distance Vector Routing Algorithm and , Count to Infinity Problem, Link State Routing Algorithm; Congestion - Open and Close Loop Control; Congestion control in Datagram Subnets; Traffic Shaping - Leaky bucket and Token Bucket; Fragmentation; Firewall; Tunneling; IP address and its classes; Unicast and Multicast Routing;

UNIT-IV

Transport Layer – Transport layer Services; Sockets and its types; Addressing in Transport Layer; Crash Recovery; TCP and UDP; TCP Protocols; TCP Segment Header; Check Sum; TCP transmission Policy – Silly Window Syndrome; TCP Congestion Protocol; Session Layer and Presentation Layer; Domain Name System; Electronic Mail; MIME; SMTP; Email- Gateways; FTP; TFMP; Caching; Mail Server

Pedagogy

PPT, Lectures, Case studies

Suggested readings:

- A.S. Tannenbaum, “Computer networks”, Second Ed., Prentice Hall India.
- Halsall, “Data Communication, Computer Networks”, Pearson Education.

Evaluation criteria:

- Written Test, Case Study, Assignments, Class Presentation

Subject: SOFTWARE ENGINEERING

Code: BCA303

Credits: 4

Mode: Offline

OBJECTIVES:

- To provide the students with the concept of software engineering fundamentals, principles and skills needed to develop and maintain high quality software products.
- To make the students to learn the processes and techniques of software engineering which include requirements specification, design, implementation, testing and management of software projects.

LEARNING OUTCOMES:

- Understand the principles and techniques of software engineering.
- Apply software development methodologies to design and develop software applications.
- Analyze, design and develop software requirements specifications and software designs.
- Understand the role of testing in software development and apply software testing techniques to verify software.
- Apply software project management concepts to manage software development projects effectively.
- Understand the concepts of software maintenance and apply maintenance techniques to manage software maintenance.

- Develop skills to work in a team and communicate effectively with team members and clients.
- Develop problem-solving skills and apply them to solve real-world software engineering problems.
- Understand the ethical and professional responsibilities of software engineers and apply them in software development projects.
- Gain knowledge and experience in emerging software engineering technologies and tools.

UNIT-I

Introduction –Evolution, software definition, S/W types, S/W characteristics, Software failures; Software engineering: definition, Terminology, Components, Application, Myths, Software Engineering Process and Product;

UNIT-II

Software Development Life Cycle (SDLC) Models; SRS and S/W Design – Role of SRS, IEEE Standards for SRS Documents, Requirement Engineering; Structured Information – DFD and Data Dictionary ; Requirements specification design fundamentals – characteristic of SRS, SRS Validation , Components of SRS, Entity-Relationship Diagram;

UNIT-III

Software Project Planning, management and Metrics–Project management process, Measuring software, LOC and function point metrics, metrics for software quality; Estimation – Scope, resources, estimation technique, COCOMO model; Decomposition Technique – Empirical Models , automated tools; Design specification, design objectives and principles, structured design, Modularity, Coupling, Cohesion, Structured design Methodology, Most Abstract Input(MAI); OO design; verification;

UNIT-IV

Coding and Testing – Program Development, Verification, Monitoring and Control; Testing fundamentals; testing principles and objectives, Functional Testing; Structural testing; Testing Strategies, level of testing, test plan, test case design
SQA and Software Maintenance – SQA Plans; Formal technique reviews; Metrics; Corrective Maintenance; Adaptive Maintenance and Preventive Maintenance;

UNIT-V

IEEE-CS/ACM - Software Engineering Code of Ethics-Introduction, purpose, preamble, principles-public, client and employer, product, judgment, management, profession, colleagues, self; Plagiarism – What is it, types, Five levels or degrees of plagiarism, prevention, MLCU policy, IEEE plagiarism guidelines, citation-APA, IEEE;

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignment.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion

Semester VI

Subject: PYTHON PROGRAMMING

Code: BCA351

Credits: 4

Mode: Offline/Online

OBJECTIVE:

To enable students understand python programming paradigm and develop applications using it.

LEARNING OUTCOMES:

- Understand the fundamental concepts of programming and data structures using Python.
- Write basic Python programs using conditional statements, loops, functions, and file handling.
- Design and implement Python programs for solving computational problems.
- Use Python libraries such as NumPy and Pandas for data analysis and processing.
- Implement various algorithms such as sorting and searching using Python.
- Understand and use object-oriented programming concepts in Python.
- Apply knowledge of Python programming in solving real-world problems.

UNIT-I

Introduction to Python-Installation and Working with Python, variables, Operators understanding python blocks; Data types- Declaring and using Numeric data types: int, float, complex, Using string data type and string operations Defining list and list slicing, Use of Tuple data type, implement necessary program for the topics

UNIT- II

Python program flow control-Conditional blocks using if, else and elif, Simple for loops in python, For loop using ranges, string, list and dictionaries, Use of while loops in python Loop manipulation using pass, continue, break and else Programming using Python conditional and loops block, Functions, modules and packages-Organizing python codes using functions Organizing python projects into modules Importing own module as well as external modules Understanding Packages Powerful Lamda function in python Programming using functions, modules and external packages, implement necessary program for the topics

UNIT- III

Python String, List and dictionary manipulations, building blocks of python programs Understanding string in build methods List manipulation using in build methods Dictionary manipulation Programming using string,list and dictionary in build functions, python fileoperation, Reading config files in python Writing log files in python

Understanding read functions, read(), readline() and readlines()Understanding write functions, write() and writelines()Manipulating file pointer using seek Programming using file operations, implement necessary program for the topics

UNIT-IV

Python object oriented programming–OOps Concept of class, object and instances Constructor, class attributes and destructors Real time use of class in live projects Inheritance , overlapping and overloading operators Adding and retrieving dynamic attributes of classes Programming using OOps support8 : Python Regular Expression Powerful pattern matching and searching Power of pattern searching using regex in python Real time parsing of networking or system data using regex Password, email, url validation using regular expression Pattern finding programs using regular expression, Python Exception Handling Avoiding code break using exception handling safe guarding file operation using exception handling, handling and helping developer with error code, programming using Exception handling, implement necessary program for the topics

UNIT- V

Python database interaction-SQL Database connection using python, creating and searching tables Reading and storing config information on database Programming using database connections, implement necessary program for the topics

REFERENCES:

1. Martin C. Brown,“Python: The Complete Reference”,McGraw Hills
2. YashavantKanetkar,”Let us Python”, BPB Publications
3. R. NageswaraRao,“Core Python Programming”, Dreamtech Press
4. Bill Lubanovic,“Introducing Python”,Shroff Publishers

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Semester VII

Honours: Web-technology

Subject: WEB SERVICES

Code: BCA400

Credits: 4

Mode: Offline

OBJECTIVE: To provide knowledge on application-to-application interactions on the Web and integrate the existing network computer infrastructure into the Web.

UNIT-I

Introduction to web services- Fundamentals of XML, XML Syntax, XML Document Structure, Schema Languages; DTD, XML Schema; Presentation technologies –XSL, XFORMS, XHTML–Transformation –XSLT, XLINK, XPATH, Xquery; Developing Web services-Objectives, Web service standards, SOAP-The Processing model, Faults, Data representation and RPC, Protocol binding, WSDL-Interface Descriptions, Binding description, service description, UDDI-Descriptions, Discovery

UNIT-II

Business motivations for web services – B2B, B2C, Technical motivations, limitations of CORBA and DCOM, Service oriented Architecture (SOA), Architecting web services, Implementation view, web services technology stack, logical view, composition of web service, deployment view, process view

UNIT-III

Transport protocols for web services, messaging with web services protocols, SOAP, describing web services, WSDL – Anatomy of WSDL, manipulating WSDL; web service policy – Discovering web services, UDDI, Anatomy of UDDI, Web service inspection, Ad-Hoc Discovery, Securing web services

UNIT-IV

Implementing XML in E-business-B2B, B2C Application; Different types of B2B interaction, Components of e-business, XML systems – ebXML, Rosetta Net Applied XML in vertical industry, Web services for mobile devices

UNIT-V

XML and Content Management-Semantic Web, Role of Meta data in web content, Resource Description Framework, RDF schema, Architecture of semantic web, content management workflow, XLANG WSFL

REFERENCES:

1. Ron schmelzer et al, “XML and Web Services”, Pearson Education, 2002
2. Sandeep Chatterjee and James Webber, “Developing Enterprise Web Services: An Architect’s

Guide”, Prentice Hall, 2004

3. Frank P. Coyle, “XML, Web Services and the Data Revolution”, Pearson Education, 2002

4. Keith Ballinger, “.NET Web Services Architecture and Implementation”, Pearson Education, 2003

Subject: PHP

Code: BCA401

Credits: 4

Mode: Offline/Online

OBJECTIVE:

To provide students with basic PHP technology with emphasis on program structure, language syntax and its implementation.

LEARNING OUTCOMES:

- Understand the fundamental concepts and principles of PHP programming language, including its syntax, data types, control structures, and functions.
- Analyse and debug PHP code using various debugging tools and techniques.
- Evaluate and compare different PHP frameworks.
- Keep up-to-date with emerging trends and technologies in the field of PHP development.

UNIT-I

Essential PHP - Getting PHP, Creating your development Environment, Creating a first PHP Page, Mixing HTML and PHP, Printing some text, Printing some HTML, Echo power, Working with variables, Storing Data in variables, Interpolating Strings, Creating variable variables, Creating constant, Understanding PHP internal Data Types

UNIT-II

Operators and Flow Controls – PHP Math operators, Working with the assignment operators, Incrementing and decrementing values, String operators, Operator precedence, Using If statement, PHP Comparison operators, PHP Logical operators, Else statement, Elseif statement, Switch statement, Using For loops, Using While loops, Using Do...While loops, Using foreach loop, Terminating loops early, PHP alternate syntax

UNIT-III

Strings and Arrays – String functions, Converting to and from strings, Formatting Text strings, Building yourself some arrays, Modifying the Data in arrays, Deleting arrays with loops (for loop, print_r function, for each loop, while loop), PHP array Functions, Extracting Data from arrays, Sorting arrays, Using PHP array operators, Comparing array with each other, Handling Multidimensional arrays in loops, Splitting and Merging arrays, other array functions

UNIT-IV

Creating Functions – Creating functions in PHP, Passing functions some Data, Passing arrays to functions, Passing by reference, Passing variable numbers to arguments, Returning Data from arrays,

Returning arrays, Returning List, returning reference, Introducing variable Scope in PHP, Accessing Global Data, Working with Static variables, PHP conditional functions, PHP variable functions

UNIT-V

Reading Data in Web Pages – Setting up web pages to communicate with PHP, Handling Text fields, Handling Text areas, Handling Check boxes, Handling radio buttons, Handling List boxes, Handling Password controls, Handling Hidden controls, Handling image Maps, Handling Buttons(Making Button data Persist, using Submit Buttons as HTML buttons)

SUGGESTED READINGS:

1. Peter MacIntyre , RasmusLerdorf , Kevin, "Programming PHP", O'Reilly
2. GSteven Holzner, "Php: The Complete Reference", McGraw Hill Education
3. VikramVaswani, "PHP 5.3: A Beginner's Guide : A Beginner's Guide", McGraw Hill Education
4. Janet Valade, "PHP and MySQL For Dummies, 4th Edition", John Wiley & Sons Inc

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignment.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion

Subject: PHP-PRACTICAL

Code:

Credits: 4

Mode: Offline

OBJECTIVE:

To practice writing program using PHP on notepad++ and other IDE

LEARNING OUTCOMES:

- Develop dynamic web applications using PHP and integrate them with a backend database.
- Apply the concepts of object-oriented programming (OOP) in PHP to create reusable and maintainable code.
- Apply security best practices in PHP programming to prevent common vulnerabilities and attacks.
- Collaborate effectively in a team to design, develop, and deploy PHP-based web applications.

Course Content:

1. WAP in PHP to print some text
2. WAP in PHP to store data in variables, Interpolating Strings, Creating variable variables, creating constant.
3. WAP in PHP using math operators, for Incrementing and decrementing values, String operators, Operator precedence.

4. WAP in PHP using If statement, PHP Comparison operators, PHP Logical operators, Else statement, Elseif statement, Switch statement, Using For loops, Using While loops, Using Do...While loops, Using foreach loop, Terminating loops early, PHP alternate syntax.
5. WAP in PHP using String functions, Modifying the Data in arrays, Deleting arrays with loops (for loop, print_r function, foreach loop, while loop), PHP array Functions, Extracting Data from arrays, Sorting arrays, Using PHP array operators, Comparing array with each other, Handling Multidimensional arrays in loops, Splitting and Merging arrays, other array functions
6. WAP in PHP using Creating functions in PHP, Passing functions some Data, Passing arrays to functions, Passing by reference, Passing variable numbers to arguments, Returning Data from arrays, Returning arrays, Returning List, returning reference, Introducing variable Scope in PHP, Accessing Global Data, Working with Static variables, PHP conditional functions, PHP variable functions.
7. WAP in PHP using Handling Text fields, Handling Text areas, Handling Check boxes, Handling radio buttons, Handling List boxes, Handling Password controls, Handling Hidden controls,

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Coding, Class Presentation, Group discussion

Honours: Hardware and Networking

Subject: BASIC ELECTRONICS

Code: BCA404

Credits: 4

OBJECTIVE:

To make the students understand the efficacy of electronic principles which are pervasive in engineering applications

LEARNING OUTCOMES:

- Analyze and apply the principles of basic electronics theory to solve problems related to electrical circuits.
- Understand the functioning of electronic components such as resistors, capacitors, and diodes, and apply them in various circuit configurations.
- Develop a basic understanding of semiconductors, their properties and their applications in electronic circuits.
- Apply the knowledge of electronic circuit analysis in designing and analyzing basic electronic circuits.
- Understand the basic concepts of digital electronics and Boolean algebra.

- Design simple logic circuits using basic gates.

UNIT-I

Basic Electricity and conducting Material: Introduction, Current, Voltage, emf, Power generation system, Switch- plug wiring, Analyzing Conductivity of elements, Types of Conductors, Semiconductors - Silicon, Germanium.

UNIT-II

Electronics Components: Resistors, Capacitors, Inductors, Transformers, Types, working and Properties, Voltage and current sources, Diode, Zener diode, Photo diode, Light emitting diode(LED), Transistors (NPN,PNP), their characteristics and uses, Field effect transistor, Phototransistor.

UNIT-III

Electronics Circuits: AC Fundamentals, Ohm's law, Series and Parallel connection of Registers and Capacitors, Half wave rectifier, Full wave rectifier and Bridge rectifier.

UNIT-IV

Regulated Power Supply: Basic regulated power supply using Zenerdiode;Block diagram of IC based Power supply;Basic Switch Mode Power Supply (SMPS); Basic uninterrupted Power Supply (UPS)

UNIT-V

Basic Measuring Instruments: Multimeters – Electronics and Digital, Cathode Ray Oscilloscope (CRO), Block diagram and basic working; Different uses of CRO, LCR – Q meter. Different tools used for practicals; Soldering and desoldering practice

SUGGESTED READINGS:

1. B.L Theraja, "Basic Electronics", S.Chand
2. Albert Paul Malvino,"Digital computer Electronics, and Code", Tata McGraw-Hill Public
3. Malvino,"Electronics Principles", McGraw-Hill Publication

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: WINDOWS SERVER ADMINISTRATION

Code: BCA405

Credits: 4

Mode: Offline

OBJECTIVE: The goal is to equip the students with the skills to Configure, administer and manage a Windows Server.

LEARNING OUTCOMES:

- Install and configure Windows Server operating system on a network.
- Manage user accounts and group policies to provide security and access control on a network.
- Manage network services such as DHCP, DNS, and file sharing to ensure smooth functioning of the network.
- Monitor and maintain server performance to ensure optimal functioning of the network.
- Plan and implement backup and disaster recovery strategies to safeguard network data and minimize data loss.
- Implement and manage Active Directory services to provide centralized authentication and authorization for network resources.

CONTENTS:

1. Installing and Configuring Windows Server 2008
2. Install Server Core; optimize resource utilization by using Features on Demand; migrate roles from previous versions of Windows Server
3. Configure Server Core; delegate administration; add and remove features in offline images; deploy roles on remote servers; convert Server Core to/from full GUI; configure services; configure NIC teaming
4. Configure local storage ;Design storage spaces; configure basic and dynamic disks; configure MBR and GPT disks; manage volumes; create and mount virtual hard disks (VHDs); configure storage; pools and disk pools
5. Configure server roles and features ;Configure file and share access; Create and configure shares; configure share permissions; configure offline files; configure NTFS permissions; configure access-based enumeration (ABE); configure Volume Shadow Copy Service (VSS); configure NTFS quotas ;Configure print and document services
6. Configure the Easy Print print driver; configure Enterprise Print Management; configure drivers; configure printer pooling; configure print priorities; configure printer; permissions
7. Configure servers for remote management; Configure WinRM; configure down-level server management; configure servers for day-to-day management tasks; configure multi-server management; configure Server Core;
8. Configure Windows Firewall ;Configure Hyper-V
9. Create and configure virtual machine settings ;Configure dynamic memory; configure smart paging; configure Resource Metering; Configure guest integration services
10. Create and configure virtual machine storage ;Create VHDs and VHDX; configure differencing drives; modify VHDs; configure pass-through disks; manage snapshots; implement a virtual Fibre Channel adapter

11. Create and configure virtual networks; Implement Hyper-V Network Virtualization; configure Hyper-V virtual switches; optimize network performance; configure MAC addresses; configure network isolation; configure synthetic and legacy virtual network adapters;
12. Deploy and configure core network services ;Configure IPv4 and IPv6 addressing Configure IP address options; configure subnetting; configure supernetting; configure interoperability between IPv4 and IPv6; configure ISATAP; configure Teredo
13. Deploy and configure Dynamic Host Configuration Protocol (DHCP) service; Create and configure scopes; configure a DHCP reservation; configure DHCP options; Configure client and server for PXE boot; configure DHCP relay agent; authorize DHCPserver
14. Deploy and configure DNS service; Configure Active Directory integration of primary zones; configure forwarders; configure Root Hints; manage DNS cache; create A and PTR resource records
15. Install and administer Active Directory; Install domain controllers; Add or remove a domain controller from a domain; upgrade a domain controller; install Active Directory Domain Services (AD DS) on a Server Core installation; install a domain controller from Install from Media (IFM); resolve DNS SRV record registration issues; configure a global catalog server
16. Create and manage Active Directory users and computers; Automate the creation of Active Directory accounts; create, copy, configure, and delete users and computers; configure templates; perform bulk Active Directory operations; configure user rights; offline domain join; manage inactive and disabled accounts
17. Create and manage Active Directory groups and organizational units (OUs) ;Configure group nesting; convert groups including security, distribution, universal, domain local, and domain global; manage group membership using Group Policy; enumerate group membership; delegate the creation and management of Active Directory objects; manage default Active Directory containers; create, copy, configure, and delete groups and OUs
18. Create and manage Group Policy ;Configure a Central Store; manage starter GPOs; configure GPO links; configure multiple local group policies; configure security filtering
19. Configure security policies; Configure User Rights Assignment; configure Security Options settings; configure ;Security templates; configure Audit Policy; configure Local Users and Groups; configure
20. User Account Control (UAC) ;Configure application restriction policies; Configure rule enforcement; configure Applocker rules; configure Software Restriction Policies
21. Configure Windows Firewall ;Configure rules for multiple profiles using Group Policy; configure connection security rules; configure Windows Firewall to allow or deny applications, scopes, ports, and users; configure authenticated firewall exceptions; import and export

Suggested Readings:

1. Hassell J.,” Learning Windows Server 2008”, O’Reilly Media.
2. Hassell J., “Windows Server 2008: the definitive guide”, O’Reilly Media
3. Tom Carpenter, ”Microsoft Windows Server Administration Essentials Courseback”, Pearson
4. MTA Windows Server Administration Fundamentals (Microsoft Official Academic Course) Courseback ;Microsoft Official Academic Course
5. Mark Minasi, Kevin Greene , Christian Booth, Robert Butler, John McCabe, “Mastering Windows Server 2012 R2”

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: WIRELESS NETWORK SECURITY

Code:

Credits: 4

Mode: Offline/Online

OBJECTIVES:

- To enable students to acquire basic knowledge of wireless cellular, ad hoc and sensor networks.
- To enable students to acquire knowledge about wireless communication fundamentals.
- To understand the impact of mobility on routing protocols,

LEARNING OUTCOMES:

- Understand the fundamental concepts of wireless networks and their security mechanisms.
- Analyze the security risks and vulnerabilities associated with wireless networks.
- Evaluate and select appropriate security mechanisms for different wireless network scenarios.
- Apply ethical and legal principles while working with wireless network security.
- Conduct research on emerging wireless network security technologies and trends.

UNIT-I

Introduction to wireless network architectures: cellular networks, wireless local area networks, multi-hop networks. Cellular systems- Frequency Management and Channel Assignment- types of handoff and their characteristics, dropped call rates & their evaluation - MAC – SDMA – FDMA –TDMA – CDMA – Cellular Wireless Networks.

UNIT-II

Wireless LAN and Wireless Wans-IEEE 802.11 Standards – Architecture – Services, Physical Layer- MAC sublayer- MAC Management Sublayer, Other IEEE 802.11 standards, HIPERLAN, WiMax standard; Wireless wans-First Generation Analog, Second Generation TDMA – GSM, Short Messaging Service in GSM, Second Generation CDMA – IS-95, GPRS - Third Generation Systems (WCDMA/CDMA 2000).

UNIT-III

Wireless MANS AND PANS-Wireless MANs – Physical and MAC layer details, Wireless PANs – Architecture of Bluetooth Systems, Physical and MAC layer details, Standards.

UNIT-IV

Adhoc and sensor networks-Characteristics of MANETs, Table-driven and Source- initiated On Demand routing, protocols, Hybrid protocols, Wireless Sensor networks- Classification, MAC and Routing protocols.

UNIT-V

Services, mechanisms and attacks; Security architecture – security services, authentication, data confidentiality, data integrity, nonrepudiation, availability; Security Mechanisms-attacks; Security network model.; Classical Encryption techniques-Symmetric cipher model, Cryptography, Cryptanalysis; Substitution techniques – Caesar Cipher, Monoalphabetic Cipher, Playfair Cipher, Transposition techniques. Authentication and key establishment ,Buffer overflow attacks ,Websecurity, Internet worms, viruses, spyware, Spam, phishing, botnets, denial of service ,TCP/IP and DNS security ,Firewalls and intrusion detection systems Wireless security.

Suggested Readings:

1. William Stallings, "Wireless Communications and networks", Pearson Education
2. Dharma Prakash Agrawal & Qing-An Zeng, "Introduction to Wireless and Mobile Systems", Thomson India Edition
3. Kaufman, Perlman, and Speciner," Network Security", Pearson Education

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Elective: Mobile Applications

Subject: RESPONSIVE WEB DESIGNING

Code: BCA407

Credits: 4

Mode: Offline

OBJECTIVES:

To enable students designing responsive sites using a combination of fluid layouts, media queries, and fluid media; adopt a responsive workflow from the very start of a project.

LEARNING OUTCOMES:

- Demonstrate a comprehensive understanding of the principles and technologies used in responsive web design.
- Develop responsive web layouts using HTML, CSS and JavaScript.
- Use responsive web design tools and frameworks to create effective and user-friendly web designs.
- Implement responsive web design techniques to improve website performance and accessibility.
- Apply knowledge of responsive web design best practices and ethical considerations when creating web designs.

UNIT-I

Foundation of responsive design-what is responsive design, why responsive design; Responsive content-content strategy, managing content, content governance, adaptive content

UNIT-II

HTML for responsive sites- working with HTML, basic page structure, viewport, structural elements, creating page, clean and semantic HTML; CSS for responsive sites-how CSS works, CSS version, using cascade, organizing your stylesheet, the box model, display, positioning, float and clear, basic styles; media queries-what is media query-structure-using media queries in stylesheet links, what we can query, browser support, breakpoints, design ranges; Images-way to display images, alt text, imagefile formats, optimizing images, content images, background images, responsive images

UNIT-III

Working responsively-responsive workflow, strategy and planning, content before layout, thinking about layout, prototypes, visual design, responsive design tools; Mobile and beyond-user experience, device agnostic design, focusing on mobile first, types of devices, touch, screen size, accessibility, deciding which devices to support, testing

UNIT-IV

Designing responsive websites-typography, start with HTML, typefaces, using fonts, sizing text, line length, whitespace, margins and padding, changing typeface for screen size; Navigation and header layout-responsive navigation, branding, navigation links, navigation patterns

UNIT-V

Performance-why performance matters, performance as design, how web pages are loaded and rendered, measuring performance, cleaning up code, minimizing HTTP requests, server stuff, Javascript, CSS, hosting, conditionally loading content, reflows and repaints, RESS

Suggested Readings:

1. Clarissa Peterson, "Learning Responsive Web Design: A beginner's guide", O'Reilly Media, Inc
2. Brett Romero, "Responsive Web Design Overview : For Beginners", Createspace Publication
3. Benjamin LaGrone, "HTML5 and CSS3 Responsive Web Design Cookbook", Shroff / Packt Publication

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: ANDROID PROGRAMMING

Code: BCA408

Credits: 4

Mode: Offline/Online

OBJECTIVE:

- To enable student, understand the android application life cycle,
- To identify,analyze and acquire skills for developing android applications

LEARNING OUTCOMES:

- Understand the basics of mobile app development and design user-friendly interfaces.
- Implement core Android components such as activities, intents, and services in their applications.
- Use different data storage methods such as SQLite, Shared Preferences and Cloud storage to persist and retrieve data.
- Apply best practices for security, performance optimization, and app publication to the Google Play Store.

UNIT-I

Introduction to Android, Smartphone features, Installing the SDK, Creating Android Emulator, Installing Android development tools, Choosing which Android version to use, Android Life cycle, Android applications structure.

UNIT-II

Creating a project, Working with android manifest.XML, Various controls, Layouts, Text controls, Button controls Images, Supporting Multiple Screen, Android Activities, Application context, Intent WebView.

UNIT-III

List View, Spinner, AutoComplete Textview, MultiAutoComplete extview, Toast, Dialogue Notification, Statusbar Notification, Option Menu, Context Menu, contextual action mode, Popup menu, menu from xml, Linkify, Match Filter & Transform Filter

UNIT-IV

Saving and Loading Files, SQLite Databases, Android Database Design, Exposing Access to a Data Source through a Content Provider, Content Provider Registration, Native Content Providers, Intent Overview, Implicit Intents, Creating the Implicit Intent Example Project, Explicit Intents, Creating the Explicit Intent Example Application, Intents with Activities, Intents with Broadcast Receivers, An Overview of Threads, The Application Main Thread, Thread Handlers, A Basic Threading Example, Creating a New Thread, Implementing a Thread Handler, Passing a Message to the Handler

UNIT-V

Sending SMS Messages Programmatically, Getting Feedback after Sending the Message Sending SMS Messages Using Intent Receiving, sending email, Introduction to location-based service, configuring the Android Emulator for Location-Based Services, Geocoding and Map-Based Activities, Multimedia: Audio, Video, Camera, Playing Audio and Video, Recording Audio and Video, Using the Camera to take and Process Pictures

Suggested Readings:

1. Warren Tim, "Android Programming For Beginners", Ingram Publishing
2. Zigurd Mednieks, Laird Dornin, G. Blake Meike, "Programming Android", O'Reilly
3. Jason Wei, "Android Database Programming", Packt Publishers

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: ANDROID PROGRAMMING-PRACTICAL

Code:

Credits: 4

Mode: Offline

OBJECTIVE:

- To provide students with an understanding of the principles and concepts of Android application development.,
- To enable students to create, develop and test Android applications using Android Studio.
- To teach students how to publish their Android applications on Google Play Store.
- To familiarize students with various tools and techniques for debugging and testing Android applications.

LEARNING OUTCOMES:

- Develop a basic understanding of the Android application development framework and environment.

- Design and develop Android applications using Java and Android Studio.
- Implement various Android components such as activities, fragments, intents, and services.
- Implement Android UI components such as layouts, views, and widgets.
- Test and debug Android applications using Android Studio and other debugging tools.
- Publish and distribute Android applications to the Google Play Store.

CONTENTS:

1. Setting up Android Studio
2. Creating a new project
3. Understanding project structure
4. User Interface Design in Android: Introduction to Layouts: Linear Layout, Relative Layout , Table Layout, Creating custom views.
5. Working with Views and Widgets: Understanding Views and View Groups, Text Views, Buttons and Checkboxes, Spinners, Date Pickers and Time Pickers.
6. Working with Activities and Intents: Understanding Activities, Creating new activities, Passing data between activities using Intents.
7. Working with Fragments: Introduction to Fragments, Creating Fragments, Adding Fragments to Activities.
8. Working with Resources: Understanding Resources in Android, Strings, Colors, and Styles, Drawable resources, Working with images.
9. Networking and Web Services: Working with HTTP requests, Parsing JSON data, Displaying images from the internet.
10. Saving Data: Understanding Data Storage in Android, Shared Preferences, Internal Storage, External Storage.
11. Android Debugging and Testing: Debugging Android Applications, Testing Android Applications.
12. Publishing Android Application: Preparing the application for release, Signing the application, Publishing application on Google Play Store.

Suggested Readings:

1. Warren Tim, "Android Programming For Beginners", Ingram Publishing
2. ZigurdMednieks, Laird Dornin, G. Blake Meike, "Programming Android", O'Reilly
3. Jason Wei, "Android Database Programming", Packt Publishers

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: ANALYSIS AND DESIGN OF ALGORITHMS

Code: BCA402

Credits: 4

Mode: Offline/Online

OBJECTIVE:

- To analyze algorithm, its efficiency and the fundamental component of problem solving.
- To understand the importance of algorithm and finding its time and space complexity both theoretically and practically.

LEARNING OUTCOMES:

- Develop an understanding of fundamental concepts in algorithm design and analysis, including sorting, searching, and graph algorithms.
- Analyze the time and space complexity of an algorithm using Big-O notation and other techniques.
- Develop the ability to evaluate the efficiency and effectiveness of algorithms and select the appropriate algorithm for a given problem.
- Develop problem-solving skills, critical thinking, and logical reasoning ability through the course.

UNIT-I

Introduction – Fundamentals of Algorithmic Problem Solving, Statement of the Problem, Design of an Algorithm, Correctness of an Algorithm, Analyzing an Algorithm, Implementation of Algorithm; Problem Types – Searching, Sorting, Graph Related Problems; Graph Representations – Adjacency matrix, Adjacency List, Path Matrix, Spanning Tree; Graph Properties – Bipartite Graph; Analysis of Algorithm Efficiency – Space Complexity, Analysis of Space Complexity, How to calculate Space Complexity, time complexity; Asymptotic Notations – Big Oh Notation, Omega Notation, Theta Notation, Little Oh Notation, Comparison of Asymptotic Notations.

UNIT-II

Mathematical Analysis Recursive – Backward Substitution Method, Important Recurrence Type, Fibonacci Numbers, Recursion Tree; Brute Force Method – Bubble Sort, Implementation of bubble sort, Selection Sort, Implementation of Selection Sort; Exhaustive Search – Travelling Salesman Problem, Knapsack Problem, and Assignment Problem; Divide and Conquer – Merge sort, Analysis and Implementation of Merge Sort, Quick Sort, Analysis and Implementation, Binary Search, Analysis and Implementation; Multiplication of Large Integers; Strassen's Matrix Multiplication.

UNIT-III

Decrease and Conquer – Insertion sort: analysis of Insertion Sort and its implementation; DFS and BFS –Depth First Search, Breadth First Search; Topological Sort; Transform and Conquer – Balanced Search Tree, AVL Trees; Heap sort – Heaps, Initial Heap Construction, Inserting a Key into a Max Heap, Deleting a key from Max Heap

UNIT-IV

Space and Time Tradeoffs – Sorting by Counting; String Matching – Horspool Algorithm and its implementation; Hashing – hash Functions, Collision Resolution Techniques; Dynamic Programming – Warshall's Algorithm; Floyd's Algorithm; Knapsack problem.

UNIT-V

Greedy Technique – Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Codes and Tree; Backtracking & branch and bound, n – Queens Problem, Assignment Problem, Knapsack Problem, Travelling Salesman Problem; Limitation of Algorithm Power – P, NP and NP – Complete Problems – NP – Completeness, Polynomial Time, NP – completeness and reducibility.

Suggested Readings:

1. Puntambekar, "Analysis and Design Of Algorithms", Technical Publications
2. Anany Levitin, "Introduction to Design of Analysis and Algorithm", Addison Welsey Edition
3. Robert Sedgewick and Phillippe Flajolet, "An Introduction to the Analysis of Algorithm (2nd Edition)", Welsey Publication.
4. Sara Baase and Allen Van Gelder, "Computer Algorithm: Introduction to Design and Analysis of Algorithm (3rd Edition)".

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignment.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion

Subject: ADVANCED DATABASE MANAGEMENT SYSTEM (ADBMS)

Code: BCA403

Credits: 4

Mode: Offline/Online

OBJECTIVE:

To provide students with an understanding of the design, creation, maintenance and management of the open source relational database management system, MySQL.

LEARNING OUTCOMES:

- Understand the concept and importance of advanced database management systems
- Demonstrate knowledge of the basic and advanced concepts of database management systems.
- Analyze and design database systems using advanced concepts of normalization and data modeling.

UNIT-I

Overview of the database management system[1]- Database systems, Need for Database, Advantages of using a database, Characteristics of data in a database, Functions of DBMS, Data abstraction, Data independence, Overall Architecture of DBMS, Three level architecture; Data Models[1]- Relational Data Models, ER Model, Hierarchical models, Networking models, Advantages and Disadvantages of each models.

UNIT-II

Relational Model[1]– Characteristics of Relational Database Model, CODD's rules, Tables, Rows, Columns, Domains, Attributes, Candidate Key, Primary Key, Foreign Key, Super Keys, Unique Keys, Constraints; Normalization[1] -Purpose of Normalization, Functional Dependence, Relational database Design, Normal forms, 1NF, 2NF, 3NF, BCNF, 4NF and 5 NF.

UNIT-III

Introducing MySQL[2] –History, Role of MySQL in industry, Version of MySQL, Architecture, Engines; MySQL queries[2]- Data types, operators, functions; Working with Databases and Tables- Creating, Copying, Modifying Tables; MySQL Advance [2]-Show commands, Working with date and Time data types, Joins like Cross, Inner, Outer, Self, Unions, Subquery, Procedure, Triggers, Views, MySQL SQL Injection, MySQL database export and import.

UNIT-IV

MySQL Administration[2]- MySQL Access Control System, Create User, Grant Privileges to Account, Revoking Privileges from Users, Maintaining MySQL Database Tables, Backup ,Restoring Databases Using mysqldump Tool.

UNIT-IV

Introduction to Object-Oriented Databases, Distributed databases, Client-Server Architecture, Data Mining, Data Warehousing, Deductive databases, Databases on the World Wide Web, Multimedia Databases, Geographical Information Systems.

Suggested Readings:

1. Sai Sumathi, "Fundamentals of Relational Database Management Systems", Springer (India) Pvt.Ltd
2. Vikram Vaswani, "MySQL(TM): The Complete Reference", Mc Graw Hill Education Publication
3. Ramez Elmasri, SHamkant B.Navatha, "Fundamentals of Database Systems", Pearson Education
4. Ivan Bayross, "Mastering Database Technologies", BPB Publications
5. Baron Schwartz, "High Performance MySQL", O'Reilly Publication
6. Luke Welling, Laura Thomson "MySQL Tutorial", Pearson Education Publication

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussio

Elective: Animation and Game Art

Subject: Design Fundamentals

Code:

Credits: 4

Mode: Online

OBJECTIVE:

To implement design elements, qualities, and principles to create a visually appealing design

LEARNING OUTCOMES: Students will be able to

- (a) Comprehend the fundamentals of art, shades and shadows
- (b) Classify the different perspective views of objects
- (c) Acquire the concept art skills and comprehend the concept art
- (d) Evaluate the importance of history of architecture and level Design

UNIT I

Visual Culture

Introduction to Visual Culture- Influences of Visual Art, Architecture; History of print media, illustrations, comics etc., and influences on our senses and impact on Visual Culture; Study of photography, Cinema, Television and media impact on our visual consciousness contributing to Visual Culture.

UNIT II

Visual Elements

Conceptual Elements - Point, Line, Plane, Volume; Visual Elements - Shape, Form, Size, Color, Texture- Direction, Position, Space and Gravity, Representation; Perception of Color –meaning, emotion and communication from everyday experiences. Color in Art and Design; Visual communication- through Image & Text - Meaning and associations

UNIT III

Study of Shapes& Forms

Types of Shapes- Study of Organic & Inorganic shapes -To study basic- 3Dimensional Forms- Study of Organic & Inorganic forms; Basic Geometrical forms such as Cube, Pyramid, Sphere, Cone, Cylinder using different materials such as Clay, Plaster and paper boards, Wire, straw, sticks; Study of Textures in 2D and 3 D

Unit IV

Design Principles in 2D and 3D

Balance, Harmony, Rhythm, Proportion, Scale, Unity, Dominance, Emphasis, Contrast, Movement and Space using Design Elements – such as Point, Line, Plane, Volume, Shape, Form, Size, Color & Texture; composition of Shapes and Forms; using Tessellation, Units and their shapes, transformations, and metamorphosis; Design using Black & White pigments, Repetition Structure, Similarity, Gradation, Radiation, Anomaly, Contrast, Concentration, Texture, Space.

Suggested Readings:

Aaris Sherin, "Design Elements, Colour Fundamentals: A Graphic Style Manual for Understanding How Colour Affects Design", Rockport Publishers

Steven Bradley, "Design Fundamentals—Elements, Attributes, & Principles: A Beginner's Guide to Graphic Communication", Kindle Edition

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: Graphic Design

Code:

Credits: 4

Mode: Online

OBJECTIVE:

To apply graphic design principles in the ideation, development, and production of visual message

LEARNING OUTCOMES: Students will be able to

1. Implement sketching techniques in graphic design
2. Identify and apply color wheel and color harmonies in design
3. Create logos and branding using the golden ratio and grid systems
4. Recognize the elements of good design and apply them to their work

UNIT I**Introduction to Graphic Design**

The world's most famous graphic designer story; How to use their knowledge of graphic design into today's world; Use of sketching/doodling in graphic design UNIT Introduction; Tools what we are going to cover; Future of Graphic Design; Use of graphic design in UI/UX, Motion graphic etc.

UNIT II**Color Theory**

History of color; Introduction to Pencil Colors; How different brand will be useful to achieve a beautiful result; Use of different paper; Introduction to color wheel; Different types of Color Harmonies, Cool and Warm colors; Color Psychology; Positivity and Negativity about different colors.

UNIT III**Graphics**

Raster Graphics: Working with Adobe Photoshop; Vector Graphics: Difference between vector and raster graphics; How to create vector art using reference; Create an illustrator; How to create a mandala; Understanding of Typography; Masking and another blend effect; How to make isometric design; How to create a logo using golden ratio and grid system; How to create a branding

Unit IV

Visual Communication

Elements of Good Design: Line, Shape, Space, Value, Typography; Principal of Good Design: Balance, Contrast, Figure and Ground, Alignment; Grid System in logo Design and Other layout Techniques; How to create personal branding and client branding

Suggested Readings:

- Richard Stone, "Graphic Design: A Beginner's Guide To Mastering The Art Of Graphic Design", BPB Publications
- Dabner David, "Graphic Design School", Thames & Hudson Ltd,
- Timothy Samara, "Drawing for Graphic Design: Understanding Conceptual Principles and Practical Techniques to Create Unique, Effective Design Solutions", Rockport Publishers

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: Classical animation (2D Animation)

Code: BCA409

Credits: 4

Mode: Offline

OBJECTIVE:

To create 2D Digital animations by implementing the animation principles and by using 2D software authoring environment

LEARNING OUTCOMES: Students will be able to

- Describe different types of animation such as stop motion photo animation, zoetrope, thaumatrope, cell and paper animation
- Apply tweening techniques such as shape tween, shape hint tween, object, and merge drawings
- Create characters and background scenes using bone setup tools and lip movements
- Implement Control sound and video and handle components and apply behavior using ActionScript

UNIT I

Introduction

What is Animation - Early examples of Animation - Stop Motion Photo Animation - Zoetrope – Thaumatrope – Cell and Paper Animation - Types of Animation - Facial expressions - Flash Overview – About Adobe Animate Interface - Menu Bar, Tools – Layers - Property Inspector – Timeline – Stage - Scene - File Formats: .fla, .swf – Library - Import to the stage - Import to Library - Color Swatches - Grid, Guide & Rulers - Creating New Document - Working on Stage

UNIT II

Using Layers

About Layers - Create - Delete Layers, Rename Layers - Lock & Unlock Layers - Type of Frames: Key

fame - Blank Frame - Onion Skin – FPS – Symbols – Graphics - Movie Clip & Button – Tweening - Types of Tweens - Object & Merge Drawings - Gradient Color - Color Effects

UNIT III

Animation Techniques

Filters, Shape Tween, Shape Hint Tween, About Mask, Layer Mask, Ease in and Ease out Animation, Working Frame - by - Frame Animations, Working on Tween Animation, Character creation - Animation - Creating Storyboard, Creating any Cartoon Character, Creating Background Scenes, Using Bone Setup Tool, Lip Movements, Adding Audio to Scene, Publish Movie, Publish Setting

Unit IV

Action Script

Interface of Action Script in Flash, Uses of Action Script, Event Handling, Using Code Snippets, Drag & Drop, Moving with Keyboard Arrows, Go to Frame and Stop, Play & Stop Sound Layout Creation - Timeline Actions- Creating Frame- Action Controlling Sound -Sound Controls–Video Controls- Handling Components- Applying Behavior

Suggested Readings:

- i. Robert R, Snow D, “Flash CS4 Professional Bible”, Wiley Publishing
- ii. Richard Williams, "The Animator's Survival Kit: A Manual of Methods, Principles, and Formulas for Classical, Computer, Games, Stop Motion and Internet Animators", Faber and Faber Ltd.
- iii. Harold Whitaker and John Halas , "Timing for Animation" ,Focal Press
- iv. Preston Blair, "Cartoon Animation", Walter Foster Publishing

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: Game Design and Development Technique

Code: BCA410

Credits: 4

Mode: Offline

OBJECTIVE:

To enable students creating game objects, components and develop compelling games using Unity tools and techniques

LEARNING OUTCOMES: Students will be able to

- Comprehend the basics of scripting in Unity and how it interacts with the Unity API
- Create and use materials for 2D games, including light-immune shaders and textures with transparency
- Analyze the implementation of player and enemy movement, weapons, and collision in the "2D Alien Invasion" project

- Create and test the atlas texture prefab in Unity

UNIT I

Unity Basics for 2D Games

Unity Projects, Assets, and Scenes, Project Wizard and Project Panel, Assets and Project Files Scenes Navigating Scenes and View ports, Game Objects, Transforms, and Components, Cameras, Meshes and Geometry, Scripting and the Unity API, Mono Develop, Components, Performance, Profiling.

UNIT II

Materials and Textures

Using Materials and Textures, Getting Started with Materials, Mesh Renderers, Shaders, Working with Textures, Materials for 2D Games, Use White Ambient Light, Use Light-Immune Shaders, Creating Textures, Power-2 Dimensions, Retain Quality, Expand Alpha Channels for Transparency

UNIT III

Quick 2D Workflow

Getting Started at Making —2D Alien Invasion, Adding the Player and Enemies to the Scene, Implementing Player Movement, Implementing Player Weapons with Prefabs , Creating an Ammo Prefab Object, Implementing the Ammo Trajectory, Implementing Moving Enemies and Collision, The Enemy Controller.cs Script ,Setting the Box Collider as a Trigger Volume; Editor Classes; The Create Wizard Function

Unit IV

Quad and Mesh

Procedural Geometry and Textured Quads, Getting Started with the Create Quad Feature, Setting the Quad's Anchor Point, Specifying the Asset Path, Generating the Quad Mesh ,Create Vertices ,Create Quad as an Asset, Instantiate Quad in Scene, Testing the Quad Mesh Generator; Generating Atlas Textures, Getting Started with Atlas Textures, Configuring Texture Inputs, Atlas Textures and UVs, Generating an Atlas ,Optimizing Texture Inputs ,Generating an Atlas ,Atlas Generation Generating an Atlas , Saving the Atlas Prefab, Testing the Atlas Texture

Suggested Readings:

- Patrick Felicia , "Getting Started with Unity 2D Game Development", Packt
- Claudio Scolastic,"Unity 2D Game Development Cookbook", Packt

SUGGESTED READINGSS

- Alan Thorn, "Learn Unity for 2D Game Development", Apress
- Sagar Ganatra, Sushant Kafle,"Unity 2D Game Development Projects", Packt

Pedagogy: Lectures, Power point, problem-solving, hands-on demonstration, Assignment.

Evaluation Criteria:

Lab Assignments, Lab Assessment, Class Presentation, Group discussion

Subject: Folklore of North East India

Code: BCA411

Credits: 4

Mode: Offline

OBJECTIVE:

To enable students creating original ideas for characters, setting, and plotlines in their design and development of games and animations by analyzing and synthesizing unique cultures, arts and crafts, mythical creatures and various folklore genres from North East India.

LEARNING OUTCOMES: Students will be able to

- Identify and analyse the various mythical creatures, characters of the diverse cultural conveyed through the folktales
- Create original ideas and develop characters, setting, and plotlines in their design and development of games and animations

UNIT I

Folktales of North East India

The Legend of the Iei Tree; The Blessing of the Mendicant; Houdon Lamboiba and Pabet, Tapta and folktales from Nagaland, Mizoram and other north east states of India

UNIT II

Mythical Creatures

Folktales related to the mythical creatures: Pheiccham; Lasi; U Thlen\ Thlen; Ban Jhakri; Yeti/Grape-Frut; Keibu Keioiba; Uchek Langmeidong; Taoroinai; Laikhutsangbi; Tual sumsu; Aonglemlatsu and other mythical creatures from other north east states of India

UNIT III

Legends

The Leap of Ka Likai, Legends of Lepchas, Apong; Pensam: Myth and Reality, Legends of Chawngunga

Suggested Readings:

- Dr. Rajesh Bhowmik, "Tribal Arts and Crafts of North East India", Supriya Books;
- Mrs Raby, "Folktales of the Khasis", Kessinger Publishing
- Dr Chirom Rajketan, B. Jayantakumar, "Folktales of Manipur", Culture Research Center, Manipur

Pedagogy: Lectures, Power point, Assignment.

Evaluation Criteria:

Class Presentation, Group discussion

Subject: RESEARCH METHODOLOGY

Code: BCAR412

Credits: 4

OBJECTIVE:

To enable students to use appropriate research methods and tools to solve problems in animation and game arts

UNIT I

Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable. Research Process; Research Question – Investigation Question – Measurement Issues – Hypothesis – Qualities of a good Hypothesis –Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance

UNIT II

Concept and Importance in Research – Features of a good research design – Exploratory Research Design – concept, types and uses, Descriptive Research Designs – concept, types and uses. Experimental Design: Concept of Independent & Dependent variables; Qualitative and Quantitative Research; Measurement: Concept of measurement– what is measured? Problems in measurement in research – Validity and Reliability. Levels of measurement – Nominal, Ordinal, Interval, Ratio

UNIT III

Sampling: Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non Response. Characteristics of a good sample. Probability Sample – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample – Practical considerations in sampling and sample size; Data Analysis: Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association

UNIT IV

Interpretation of Data and Paper Writing – Layout of a Research Paper, Journals in Computer Science, Impact factor of Journals, When and where to publish? Ethical issues related to publishing, Plagiarism, Self-Plagiarism, Plagiarism detection software. Citation, Citation tools, Reference Management Software-Zotero/Mendeley, Software for paper formatting- LaTeX/MS Office.

Suggested Readings:

- i. Adam Watkins, "Maya A Professional Guide", Dreamtech
- ii. Tom Meade and Shinsaka Anima, "The Complete Reference Maya", Tata MC.Graw – Hill

Subject: Data Analysis using MS-Excel and SPSS software

Code: BCA413

Credits: 4

Mode: Offline

OBJECTIVE:

1. To understand the fundamentals of MS Excel, such as reading data using various data formats, organizing and manipulating data, to some of the more advanced features.
2. To explore the use of MS Excel and SPSS for analysing descriptive and inferential statistical tests.

Learning Outcome:

1. Understanding the concept related to Ms-Excel, SPSS and its operations. Representing data diagrammatically and graphically using MS-EXCEL and SPSS.
- 2: Ability to analyse descriptive statistics test of measures of central tendency and dispersion, correlation analysis using MS-Excel and SPSS.
- 3: Ability to analyse inferential statistical tests for comparing mean and association using MS-Excel and SPSS

UNIT 1:

Introduction to MS Excel - MS Excel Options – Ribbon - Sheets - Saving Excel File as PDF, CSV and Older versions - Using Excel Shortcuts - Copy, Cut, Paste, Hide, Unhide, and Link the Data in Rows, Columns and Sheet - Using Paste Special Options - Formatting Cells, Rows, Columns and Sheets - Protecting & Unprotecting Cells, Rows, Columns and Sheets with or without Password - Page Layout and Printer Properties, VBA Macro - Introduction to VBA Macro -Recording Macro & Understanding Code Behind - Editing, Writing VBA Code and Saving as Macro

UNIT 2:

Functions: - Logical Functions - Date and Time Functions - Information Functions -Math and Trigonometry Functions - Statistical Functions - Text Functions - Charts:- Simple Bar Chart – Multiple Bar Chart – Pie Chart – Donut Chart - Line Chart – Histogram – Scatter Plot - Radar Chart – Bubble Chart

UNIT 3:

Introduction SPSS Environment: data editor, output viewer, syntax editor – Data view window – SPSS Syntax – Data creation – Importing data – Variable types in SPSS and Defining variables – Creating a Codebook in SPSS. Computing Variables - Recoding (Transforming) Variables: Recoding Categorical String Variables using Automatic Recode - Rank Cases - Sorting Data - Grouping or Splitting Data.

UNIT 4:

Analysis Descriptive Statistics for Continuous Variables - The Explore procedure - Frequencies Procedure – Descriptives - Compare Means - Frequencies for Categorical Data.

Statistical technique for Association: Pearson Correlation, Chi-square Test of Independence – Statistical technique for Compare Means: One Sample t Test, Paired Samples T Test, Independent Samples T Test, One-Way ANOVA.

Suggested Readings:

1. Wallace Wang, “Microsoft Office 2019 For Dummies”, Wiley
2. M L Humphrey, “Excel 2019 Formulas & Functions: 3 (Excel Essentials 2019)”
3. SPSS for Intermediate Statistics: Use and Interpretation, Nancy L. Leech et. al., Second edition published in 2005 by Lawrence Erlbaum Associates, Inc

Subject: DATA SCIENCE

Code: BCA414

Credits: 4

Mode: Offline/Online

OBJECTIVE:

To understand the principles of data analysis and statistics

To learn how to manipulate and process large data sets

To develop skills in data visualization

LEARNING OUTCOMES:

- Identify the different stages of the data analytics lifecycle and their significance
- Perform data analysis and visualization using various tools such as Ranges, Tables, Conditional Formatting, Sorting, and Filtering
- Describe the process of data preprocessing, including data types, possible errors, and preprocessing operations
- Describe the importance of ethical considerations in data science

UNIT I

Fundamentals of Data Science

Introduction, Why data science? Data Analytics life cycle, Types of Data analysis, Data Science tools, Areas of study in data science, Role of SQL in data science, Data Preprocessing: Introduction, data types and forms, possible data error types, various data pre-processing operations

UNIT II

Statistical Data Analysis, AI and Machine Learning

Role of statistics in data science, Descriptive statistics, Inferential statistics, Overview of AI and machine learning, Supervised machine learning, Regression methods, Classification methods, Unsupervised machine learning, Clustering methods, Association analysis; Machine Learning Algorithms: Linear Regression, K-nearest Neighbors(k-NN), K-mean, Naive Bayes, Scraping the Web: APIs and Other Tools

Unit III

Data plotting and Visualization

Introduction, Visual encoding, Data Visualization software and libraries, Basic, specialized and advanced data visualization tools, Visualization of geospatial data, Data visualization types. Data Analysis with worksheet-Ranges and Tables-Data Cleaning with Text Functions, Containing Date Values and Containing Time Values; Conditional Formatting; Sorting and Filtering; Subtotals with Ranges; PivotTable, Quick Analysis; Lookup Functions; Data Visualization-Band Chart, Thermometer Chart, Gantt chart, Waterfall Chart, Sparkline and Pivot Charts.

Unit IV

Ethics and Data Science

The Five Cs: Consent, Clarity, Consistency and Trust, Control and Transparency, Consequences, Implementing Five Cs, Data's Day of Reckoning, Ethics and Security Training, Developing Guiding Principles, Building Ethics into Data-drive Culture, Regulation

Suggested Readings:

1. Gypsy Nandi and Rupam Kumar Sharma, Data Science fundamentals and Practical Approaches, BPB Publication, First Edition, 2020
2. Bernd Held, Excel Functions and Formulas, BPB Publications
3. Ethics and Data Science by Mike Loukides, Hilary Mason, & D J Patil, O'Reilly
4. Introducing Data Science by Davy Cielen, Arno D. B. Meysman, Mohamed Ali, Latest Edition, Manning

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignment.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion

Semester VIII

Subject: THEORY OF COMPUTATION

Code: BCA 451

Credits: 4

OBJECTIVE:

To introduce students the basic concepts in theoretical computer science and the formal relationship among machines, languages and grammars

LEARNING OUTCOMES:

- Understand the formal languages and regular expressions that are used to specify the syntax of programming languages.
- Analyze the complexities of algorithms and problems using various theoretical models like Turing Machines and Finite Automata.
- Apply the concepts of Context-free grammars, Pushdown Automata and Turing Machines in designing and analyzing algorithms.
- Demonstrate proficiency in solving decision problems using Turing machines and other formal methods.
- Understand the significance and applications of formal languages and automata theory in programming languages, compilers and software engineering.

UNIT-I

Strings, Alphabet, Language, Operations, Finite state machine, definitions, Finite automation model, Acceptance of strings and languages, Non Deterministic Finite Automation, Deterministic Finite Automation, Equivalence between NFA and DFA. Conversion of NFA into DFA, Minimization of FSM, Equivalence between two FSM's. Moore and Melay machines

UNIT-II

Regular sets, Regular expressions, Identify rules, Manipulation of regular expressions. Equivalence between RE and FA, Inter conversion, Pumping lemma, Closure properties of regular sets (proofs not required), Regular grammars, Right linear and left linear grammars, Equivalence between regular linear grammar and F. A. inter conversion between RE and RG.

UNIT-III

Context free grammar, Derivation trees, Chomsky Normal Form, Greibach Normal Form, Push Down Automata, Definition, Model, acceptance of CFL, Equivalence of CFL and PDA, Interconversion, enumeration of properties of CFL (proofs omitted).

UNIT-IV

Turing Machine, Definition, Model, Design of TM, Computable functions, Recursive enumerable language, Church's hypothesis, Counter machine, Types of TM's (Proofs not required). Chomsky hierarchy of languages, Linear bounded automata and context sensitive language, Introduction of DCFL and DPDA, LR(0), grammar, Desidability of problems.

UNIT-V

Undecidability: Properties of recursive & non-recursive enumerable languages, Universal Turing Machine, Post-correspondence problem, Introduction to recursive function theory.

Suggested Readings:

1. Hopcraft H. E. & Ulman J, “Introduction to Automata Theory, Languages and Computation”
2. Peter Linz,” An Introduction to formal Languages and Automata”
3. John C. Martin, “Introduction to Languages and the Theory of Automata”
4. Lewis H. P. and Papadimiton C. H, “Elements of Theory of Computation“
5. Mishra and Chandrashekharan, “Theory of Computation”

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignment.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion

Subject: Research Methodology for Animation and Game Arts

Code: BCA 452

Credits: 4

Mode: Offline/Online

OBJECTIVE:

To enable students to use appropriate research methods and tools to solve problems in animation and game arts

LEARNING OUTCOMES: Students will be able to

- Explain the concept of research and its process, types, terminologies by giving examples
- Analyze the potential impact of the current trends on research in animation and game arts

UNIT I

Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable. Research Process; Research Question – Investigation Question – Measurement Issues – Hypothesis – Qualities of a good Hypothesis – Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance

UNIT II

Concept and Importance in Research – Features of a good research design – Exploratory Research Design – concept, types and uses, Descriptive Research Designs – concept, types and uses. Experimental Design: Concept of Independent & Dependent variables; Qualitative and Quantitative Research; Measurement: Concept of measurement– what is measured? Problems in measurement in research – Validity and Reliability. Levels of measurement – Nominal, Ordinal, Interval, Ratio

UNIT III

Sampling: Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non Response. Characteristics of a good sample. Probability Sample – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample – Practical considerations in sampling and sample size; Data Analysis: Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association

UNIT IV

Current Trends in animation and game arts; Trends for the future

Suggested Readings:

1. Adam Watkins, "Maya A Professional Guide", Dreamtech
2. Tom Meade and Shinsaka Anima, "The Complete Reference Maya", Tata MC.Graw – Hill

Pedagogy: Lectures, Power point, problem solving, Demonstration, Assignment.

Evaluation Criteria:

Written Test, Assignments, Class Presentation, Group discussion