

Mahakaushal University, Jabalpur (M.P.)



Scheme & Syllabus

For

B.Sc (Computer Science)

with Research/honors Bachelor of

Computer Applications

2022-23

Duration of Course: 4 Years

Examination Mode: Semester

Examination System: CBCS

**Mahakaushal University
Village-Aithakheda, Mukunwara Road, Post-Tilwara Jabalpur (M.P.) 482003**

CreditDistribution

SEMESTERSYSTEM			CreditsRequired								
			Sem	MJ	MI	DSE	GEC/ OEC	AECC	SEC SB/VB	FW	Total Credit
Level5	Certificate	1 st Year Pass (SemI+SemII)	I	6	6		4	4			20
			II	6	6		4	4			20
Level6	Diploma	2 nd Year Pass (SemIII+SemIV)	III	6	6		4		4		20
			IV	6	6		4		4		20
Level7	Degree	3 rd Year Pass (SemV+SemVI)	V	6		4			4	6	20
			VI	6		8				6	20
Level8	Honors/ Research	4 th Year Pass (SemVII+SemVIII)	VII	6	4	4				6	20
			VIII	6	4					10	20
Total Credit				48	32	16	16	8	12	28	160

SEMESTERSYSTEM			CreditsRequired							
			MJ	MI	DSE	GEC/ OEC	AECC	SEC SB/VB	FW	Total Credit
Level5	Certificate	1 st Year Pass	12	12		8	8			40
Level6	Diploma	2 nd Year Pass	12	12		8		8		40
Level7	Degree	3 rd Year Pass	12		12			4	12	40
Level8	Honors/ Research	4 th Year Pass	12	8	4				16	40
Total Credit			48	32	16	16	8	12	28	160

For Regular Students Course Duration:

Min. Years for Completing UG Degree	3 Years
Min Years for Completing UG (Hons.) Degree	4 Years
Maximum Years for Completing UG Degree	6 Years
Max Years for Completing UG (Hons.) Degree	8 Years

Faculty of Computer Application

Major/Minor: B.s.c AIMI

Sem	Major	Minor	DSE	Open Elective/ Generic Elective	AECC	SEC		FW	Total Credits
						Skill Based	ValueBased		
1	MJ-I(4+2)	MN-I(4+2)		OEC-I (4)	AECC-I(4)				20
2	MJ-II(4+2)	MN-II (4+2)		OEC-II(4)	AECC-II(4)				20
3	MJ-III(4+2)	MN-III (4+2)		OEC-III(4)		SECSB-I(4)			20
4	MJ-IV(4+2)	MN-IV (4+2)		OEC-IV (4)		SECSB-II(4)			20
5	MJ-V(4+2)		DSE-I(4)				SECVB (4)	FieldWork (6)	20
6	MJ-VI(4+2)		DSE-II(4) DSE-III (4)					Internship (6)	20
7	MJ-VII(4+2)	MN-V(4)	DSE-IV(4)					Minor Project(6)	20
8	MJ-VIII(4+2)	MN-VI (4)						Major Research Project(10)	20
Total Credits	48	32	16	16	8	8	4	28	160

Major/Minor

Course Code	Category	Paper	Credits
BBCS101T	MJ	PROGRAMMING IN C	4
BBCS101P	MJ	PROGRAMMING IN C Lab	2
BBCS102T	MN	PC PACKAGES	4
BBCS102P	MN	PC PACKAGES Lab	2
BBCS201T	MJ	OBJECT ORIENTED PROGRAMMING WITH C++	4
BBCS201P	MJ	C++ Lab	2
BBCS202T	MN	DISCRETE MATHEMATICS	6
BBCS301T	MJ	Object Oriented Concepts using JAVA	4
BBCS301P	MJ	JAVA Lab	2
BBCS302T	MN	COMPUTER GRAPHICS	4
BBCS302P	MN	COMPUTER GRAPHICS LAB	2
BBCS401T	MJ	DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LAB	4
BBCS401P	MJ	DIGITAL ELECTRONICS LAB	2
BBCS402T	MN	MACHINE LEARNING TECHNIQUE	4
BBCS402P	MN	MACHINE LEARNING LAB	2
BBCS501T	MJ	PYTHON PROGRAMMING	4
BBCS501P	MJ	PYTHON PROGRAMMING Lab	2
BBCS601T	MJ	WEB PROGRAMMING AND PHP	4
BBCS601P	MJ	WEB PROGRAMMING AND PHP Lab	2
BBCS701T	MJ	ASSEMBLY LANGUAGE PROGRAMMING	4
BBCS701P	MJ	ASSEMBLY LANGUAGE PROGRAMMING Lab	2
BBCS702T	MN	SOFTWARE ENGINEERING	3
BDSC702P	MN	SOFTWARE ENGINEERING Lab	1
BBCS801T	MJ	FREE AND OPEN SOURCE SOFTWARES (FOSS)	4
BBCS801P	MJ	FREE AND OPEN SOURCE SOFTWARES (FOSS) lab	2
BBCS802T	MN	MOBILE APPLICATION DEVELOPMENT	4
BBCS802P	MN	MOBILE APPLICATION DEVELOPMENT Lab	2

Department Specific Elective

BBCS101D-I	DSE	Digital marketing analysis	4
BBCS101D-II	DSE	Mobile Computing	4
BBCS102D-I	DSE	Distribution Computing	4
BBCS102D-II	DSE	Information Security	4
BBCS103D-I	DSE	Soft Computing	4
BBCS103D-II	DSE	Statistics and Data Analytics	4
BBCS104D-I	DSE	Data Mining	4
BBCS104D-II	DSE	Data Compression	4

Skill Enhancement Course (Skill Based) (Any Two)

Course Code	Category	Paper	Credits
BBCS101SB	SEC-SB	Internet of Things	4
BBCS102SB	SEC-SB	Ethical Hacking	4
BBCS103SB	SEC-SB	Embedded system	4
BBCS104SB	SEC-SB	Social Network Analytics	4
BBCS105SB	SEC-SB	Fundamental of PHP	4

Open Elective Compulsory Course / Generic Elective Compulsory Course

Course Code	Category	Paper	Credits
OECC101-I	OEC	Fundamental of Computer	4
OECC101-II	OEC	Environmental Studies	4
OECC102-I	OEC	Entrepreneurship	4
OECC102-II	OEC	Principle of Management	4
OECC103-I	OEC	Nutrition and Fitness	4
OECC103-II	OEC	Current Concerns in Public Health Nutrition	4
OECC104-I	OEC	Travel and Tourism	4
OECC104-II	OEC	Tourism Operation Software Skills	4

Ability Enhancement Compulsory Course

CourseCode	Category	Paper	Credits
AECC101	AECC	EnglishLanguage-I	4
AECC102	AECC	EnglishLanguage-II	4

Skill Enhancement Course (Value Based) (Any One)

CourseCode	Category	Paper	Credits
SECVB101	SEC-VB	Constitution of India	4
SECVB102	SEC-VB	YogainLife	4
SECVB103	SEC-VB	NationalServiceScheme(NSS)	4
SECVB104	SEC-VB	Health&Wellness	4
SECVB105	SEC-VB	Sports	4

Field Work

CourseCode	Category	Paper/Description	Credits
BFWF-501	FW	Field work is the process of observing and collecting data about people, cultures, and natural environments.	6
BFWI-601	FW	The aim of the internship provides a direction to the activities, helps to focus on a result, and to assess the result achieved.	6
BFWP-701	FW	The objective of the minor project is to provide an opportunity for students to undertake short research training outside the classroom to solve real-world issues.	6
BFWR-801	FW	Project objectives describe the desired outcome of a project, which is often a tangible object. It's beneficial to create objectives for your project because creating a specific goal for you helps everyone know what they're supposed to be working toward.	10

BBCS101T:Programming in C

(14 Lectures)

Unit-I

Overview of C: History of C , Importance of C Program, Basic structure of a C-program, Execution of C Program.

C Programming Basic Concepts: Character set, C token, Keywords and identifiers, Constants, Variables, data types, Declaration of variables, assigning values to variables, defining symbolic constants.

Input and output with C: Formatted I/O functions - *printf* and *scanf*, control strings and escape sequences, output specifications with *printf* functions; Unformatted I/O functions to read and display single character and a string - *getchar*, *putchar*, *gets* and *puts* functions.

Unit-II

(12 Lectures)

Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic expressions; Type conversion.

Unit-III

(12 Lectures)

Control Structures: Decision Making and Branching - Decision making with *if* statement, simple *if* statement, the *if-else* statement, nesting of *if...else* statements, the *if-else-if* ladder, the *switch* statement, the *?* operator, the *goto* statement. Decision making and looping - The *while* statement, the *do* statement, *for* statement, nested loops, *exit*, *break*, jumps in loops.

Unit-IV

(10 Lectures)

Derived data types in C: Arrays - declaration, initialization and access of one-dimensional and two-dimensional arrays. Programs using one- and two-dimensional arrays, sorting and searching arrays.

Handling of Strings: Declaring and initializing string variables, reading strings from terminal,

writing strings to screen, Arithmetic operations on characters, String handling functions - strlen, strcmp, strcpy, strstr and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.

Pointers: Understanding pointers, accessing the address of a variable, declaring and initializing pointers, accessing a variable through its pointer, pointer expression, pointer increments and scale factor, pointers and arrays, pointer and strings.

Unit-V

(12 Lectures)

User-defined functions: Need for user-defined functions, Declaring, defining and calling C functions, return values and their types, Categories of functions: With/without arguments, with/without return values. Nesting of functions.

Recursion: Definition, example programs.

Structures and unions: Structure definition, giving values to members, structure initialization, comparison of structure variables, arrays of structures, arrays within structures, Structure and functions, structures within structures. Unions.

Reference Books:

- E. Balagurusamy, Programming in ANSI C, 7th Edition, Tata McGraw Hill.
- Herbert Schildt, C: The Complete Reference, 4th Edition
- Brian W. Kernighan, C Programming Language, 2nd Edition, Prentice Hall Software
- Kernighan & Ritchie: The C Programming Language, 2nd Edition, PHI
- Kamthane, Programming with ANSI and TURBO C, Pearson Education
- V. Rajaraman, Computer Programming in C, 2nd Edition, PHI
- S. Byron Gottfried, Programming with C, 2nd Edition, TMH
- Yashwant Kanitkar, Letus C, 15th Edition, BPB
- P.B. Kottur, Computer Concepts and Programming in C, 23rd Edition, Sapna Book House

BBCS101P: C Programming Lab

List of Programs:

1. Program to read marks of five subjects, calculate percentage of marks and to display appropriate grade declaration message (using else-if ladder)
2. Program to find the greatest of three numbers (using nested if statement)
3. Program to read two integer values & a operator as character and perform basic arithmetic operations on them using switch case (+, -, *, / operations)
4. Program to reverse a number and find the sum of individual digits. Also check for palindrome.
5. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
6. Program to count occurrences of a character in a string.
7. Program to calculate and display the first 'n' Fibonacci numbers
8. Program to find given number is a prime or not.
9. Program to read a string and find a) length b) reverse of it c) check palindrome string d) merge original & reversed string (using built in string library functions)
10. Program to search for a number in a list of numbers using one-dimensional array.
11. Program to find the largest and smallest elements with their position in a one-dimensional array.
12. Program to read 'n' integer values into a single dimension array and arrange them in ascending order using bubble sort method.
13. Program to perform addition and subtraction of two Matrices
14. Program to display factorial of first 'n' integers using recursive function.
15. Program to check a number is a Armstrong by defining isArm() function
16. Program to read a string and count number of letters, digits, vowels, consonants, spaces and special characters present in it.
17. Program to sort a list of strings in ascending order using Pointers
18. Program to add two distances in the inch-feet format using structures (convert inches to feet if greater than 12)
19. Program to enter the information of a student like name, register number, marks in three subjects into a structure and display total, average and grade. Display details in a neat form.

20. Program to input Name of the branches, Total sales of company into an array of structures.

Display branch details in a tabular format. Also display the branch name that recorded the highest sales.

MARKU

BBCS102T: PC Packages (Windows, MS Word, MS Excel, Ms Power Point)

UNIT-I

MS Windows: Introduction to MS Windows: Features of Windows; Various versions of Windows & its use; Working with Windows;

My Computer Use: Opening, Content, Short Cut to Open, Recycle bin : Use, Moving and restoring files to and from Recycle bin. Delete File directly without sending to Recycle Bin. Empty Recycle Bin. Short cut key uses for Recycle bin.

Desktop : Definition, Purpose, Component, Changing Desktop theme, Default Icons, Windows Explorer : Purpose and Using Mechanism.

Screen description & working styles of Windows; Dialog Boxes & Toolbars; Working with Files &

Folders; simple operations like copy, delete, moving of files and folders from one drive to another,

Shortcuts & Autostarts; Accessories and Windows Settings

using Control Panel- setting common devices using control panel, modem, printers, audio, network, fonts, creating users, internet settings,

Start button & Program lists; Installing and Uninstalling new Hardware & Software program on your computer.

UNIT-II

Office Packages: Office activates and their software requirements, Word-processing, Spreadsheet,

Presentation graphics, Database, introduction and comparison of various office suites like MS-Office,

Lotus-Office, Star-Office, Open-Office etc.

MS Word Basics: Introduction to MS Office, Office Button, Introduction to MS Word, Features & area of use. Working with MS Word, Quick Access toolbar, Ribbon, Ruler, Tab, Group, Document Window, Windows Control, Status bar, Scroll Bar, Title Bar, Help Button, Zoom Slide, Menus & Commands, Toolbars & Buttons, Shortcut Menus, and Wizards & Templates.

Creating a New Document, Styles, Different Page Views and layouts, Applying various Text, Finding and Replacing Text, Enhancements, Working with -Styles, Text

Attributes Font, Size Color, Highlight. Paragraph and Page Formatting, Margins, Page Number, Header Footer Text Editing using various features; Bullets, Numbering, Auto formatting, Printing & various print options.

UNIT-III

Advanced Features of MS-Word: Spell Check, Thesaurus, Find & Replace; Headers & Footers, Inserting - Page Numbers, Pictures, Files, Auto texts, Symbols etc., Working with Columns, Tabs & Indents, Creation & Working with Tables including conversion to and from text, Margins & Space management in Document, Adding References and Graphics, Mail Merge Select Document Type, Various methods of selecting Recipients, Creating Document, Merging, Creating Envelops & Mailing Labels. Importing and exporting to and from various formats.

UNIT-IV

MS Excel: Introduction and area of use, Working with MS Excel, concepts of Workbook & Worksheets, Row, Column and Cell, Menus and Dialog box, Sheets, Using Wizards, Various Data Types, Using different features with Data, Cell and Texts, Inserting, Removing & Resizing of Columns & Rows, Working with Data & Ranges, Different Views of Worksheets, Column Freezing, Labels, Hiding, Splitting, Using different features with Data and Text, Use of Formulas, Calculations & Functions: Logical, Database, Date and Time, Information, Mathematical and statistical Functions.

Cell Formatting: Format cell Dialogue Box, Borders & Shading, Decimal Places, Date and Time Formats, Fractions, Currency vs Accounting, Text to Numbers, Numbers to Text, Custom Number Format, Format Painter, Cell Styles, Themes

Working with Different Chart Types: Create chart, Change Chart Type, Switch Row/Column, Chart Title, Legend Position, Data Labels.
Printing of Workbook & Worksheets with various options.

UNIT-V

MS PowerPoint: Introduction & area of use, Working with MS PowerPoint, The Main Screen Component, Creating a New Presentation, Working with Presentation Color, Transition, Drawing tool, Using Wizards; Slides & its different views, Inserting, Deleting and Copying of Slides; Working with Notes, Handouts, Columns & Lists, Adding Graphics, Sounds and Movies to a Slide; Working with PowerPoint Objects, Designing & Presentation of a Slide Show, Custom Animation with Timing, Chart effect, Action.

button and Action Settings, Determining What to Animate, Automate Presentation. Printing Presentations, Notes, and Handouts with print options.

Outlook Express: Features and uses, Configuration and using Outlook Express for accessing e-mails in office.

TEXT & REFERENCE BOOKS:

WINDOWS XP COMPLETE REFERENCE. BPB PUBLICATIONS MS

OFFICE XP COMPLETE BPB PUBLICATION

MS WINDOWS XP HOME EDITION COMPLETE, BPB PUBLICATION.

JOE HABA RAKEN, MICROSOFT OFFICE 2000, 8 IN 1, BY, PRENTICE HALL OF INDIA

I.T. TOOLS AND APPLICATIONS, BY A. MANSOOR, PRAGYA PUBLICATIONS, MATURA

PC Packages LAB:-

Skills to be Covered:

1. Advanced functions and formulas (e.g., VLOOKUP, INDEX-MATCH, IFERROR, nested functions).
2. PivotTables and PivotCharts for dynamic data analysis.
3. Data validation and conditional formatting for improved data quality and visualization.
4. Using Excel's statistical functions for data analysis (e.g., AVERAGEIF, COUNTIFS, SUMIFS).
5. Incorporating external data sources (e.g., SQL databases, web queries) into Excel for analysis.
6. Creating interactive dashboards with slicers, timelines, and dynamic charts.
7. Visual Basic for Applications (VBA) macros for automating repetitive tasks and customizing Excel functionality.
8. Advanced charting techniques (e.g., waterfall charts, Pareto charts, heatmaps) for insightful visualization.
9. Collaboration and sharing features in Excel (e.g., co-authoring, version history).
10. Data analysis add-ins and tools available for Excel (e.g., Solver, Analysis ToolPak).
11. Practical Activities:

Advanced Functions: Perform complex data manipulations using VLOOKUP, INDEX-MATCH, and nested functions.

1. PivotTables and PivotCharts: Analyze large datasets and create dynamic visualizations using PivotTables and PivotCharts.
2. Data Validation and Conditional Formatting: Implement data validation rules and conditional formatting to enhance data integrity and visualization.
3. Statistical Analysis: Apply Excel's statistical functions to analyze data trends and patterns.
4. External Data Integration: Import data from external sources and perform analysis within Excel.
5. Interactive Dashboards: Design interactive dashboards with slicers, timelines, and dynamic charts to explore data visually.

6. VBA Macros: Record and write VBA macros to automate repetitive tasks and customize Excel functionality.
7. Advanced Charting: Create advanced charts like waterfall charts, Pareto charts, and heatmaps for insightful data visualization.
8. Collaboration and Sharing: Explore collaboration features and share workbooks securely with peers.
9. Add-ins and Tools: Install and utilize Excel add-ins for specialized data analysis tasks.

Assessment:

1. Evaluation of the accuracy and efficiency of data analysis performed using Excel functions and tools.
2. Presentation of data analysis results through visually appealing and insightful charts and dashboards.
3. Practical examination where students demonstrate their ability to utilize advanced Excel features to solve complex data-related problems.
4. Additional Resources:
5. Online tutorials and courses on advanced Excel functions and data analysis techniques.
6. Excel forums and communities for troubleshooting and sharing best practices.
7. Official Microsoft Excel documentation and support resources for in-depth guidance.

BBCS201T : OBJECT ORIENTED PROGRAMMING WITH C++

UNIT-I

(12Lectures)

Introduction to object oriented programming, user defined types, structures, unions, polymorphism, encapsulation. Getting started with C++ syntax, data-type, variables, strings, functions, default values in functions, recursion, namespaces, operators, flow control, arrays and pointers.

UNIT- II

(14Lectures)

Abstraction mechanism: Classes, private, public, constructors, destructors, member data, member functions, inline function, friend functions, static members, and references. Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.

UNIT- III

(12Lectures)

Polymorphism: Binding, Static binding, Dynamic binding, Static polymorphism: Function Overloading, Ambiguity in function overloading, Dynamic polymorphism: Base class pointer, object slicing, late binding, method overriding with virtual functions, pure virtual functions, abstract classes.

UNIT- IV

(10Lectures)

Operator Overloading: This pointer, applications of this pointer, Operator function, member and non member operator function, operator overloading, I/O operators. Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration, unexpected exceptions, exception when handling exceptions, resource capture and release.

UNIT- V

(12Lectures)

Dynamic memory management, new and delete operators, object copying, copy constructor, assignment operator, virtual destructor.
Template: template classes, template functions.
Standard Template Library: Fundamental idea about string, iterators, hashes, iostreams and other types.
Namespaces: user defined namespaces, namespaces provided by library.

Object Oriented Design, design and programming, role of classes.

Text Books:

1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India)
2. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education

Reference Books:

1. Big C++ - Wiley India
2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India)
3. C++ and Object Oriented Programming – Jana, PHI Learning.
4. Object Oriented Programming with C++ - Rajiv Sahay, Oxford
5. Mastering C++ - Venugopal, McGraw-Hill Education (India)

LIST OF PRACTICALS OF C++

1. Write a Program That Just Outputs `Hello World
2. Write a Program to Find Maximum and Minimum of Given 3 Numbers.
3. Write a Program That Output Value as Number and as Character.
4. Implementation of the Function That Calculates the Cross Sum of an Integer.(123 as 1+2+3).
5. Determine Number of Characters in a String.
6. Raising a Number N to a Power P is the Same as Multiplying N By Itself P Times. Write a Function Called Power () That Takes a Double Value for N and an INT Value for P and Returns the Result as Double Value. use a Default Argument of 2 for P So That If This Argument is Omitted the Number Will Be Squared. Write a Main () Function That Gets Values from the User to Test This Function.
7. Write a C++ Program to Sort an Array of Integer in Ascending Order Using a Function Called Exchange() Which Accepts Two Integer Arguments By Reference.
8. Write a C++ Program to Implement Function Overloading in Order to Compute.
9. Write a C++ Program to Implement Power(MN) Where
I) M is Double and N is Int II) M and N are Int.
10. Write a Program That Uses a Structure Called Point to Model a Point. Define Three Points and Have the User Input Values to Two of Them. Then Set the Third Point Equal to the Sum of the Other Two and Display the Value of the New Point. Interaction with the Program Might Look Like This:
Enter Coordinates for P1: 3 4
Enter Coordinates for P2: 5 7 Coordinates of P1 + P2 are: 8 11

11. Create the Equivalent of a Four Function Calculator. the Program Should Request the User to Enter a Number an Operator and Another Number. It Should Then Carry Out the Specified Arithmetical Operation: Adding Subtracting Multiplying Or Dividing the Two Numbers. (It Should use a Switch Statement to Select the Operation). Finally It Should Display the Result. When It Finishes the Calculation the Program Should Ask If the User Wants to Do Another Calculation. the Response Can Be Y Or N. Some Sample Interaction with the Program Might Look Like This.

Enter First Number Operator Second Number: 10/ 3 Answer = 3.333333

Do Another (Y/ N)? Y

Enter First Number Operator Second Number 12 + 100 Answer = 112

Do Another (Y/ N)? N

12. Create a 'Distance' Class with :

- Feet and Inches as Data Members
- Member Function to Input Distance
- Member Function to Output Distance
- Member Function to Add Two Distance Objects
- Write a Main Function to Create Objects of Distance Class. Input Two Distances and Output the Sum.

13. Create a Class Called 'Time' That Has

- Three Integer Data Members for Hours Minutes and Seconds
- Constructor to Initialize the Object to Zero
- Member Function to Add Two Time Objects
- Member Function to Display Time in Hh:Mm:Ss Format
- Write a Main Function to Create Two Time Objects Add Them and Display the Result in Hh:Mm:Ss

Format.

14. Create a Class Called 'Employee' That Has

- Emrcode and Emrname as Data Members
- Member Function Getdata() to Input Data
- Member Function Display() to Output Data
- Write a Main Function to Create Emp an Array of Employee Objects. Accept and
- Display the Details of At Least 6 Employees.

15. Create a Class Rational Which Represents a Numerical Value by Two Double Values- Numerator& Denominator. Include the Following Public Member Functions: Constructor with No Arguments (Default). Constructor with Two Arguments.

- Void Reduce() That Reduces the Rational Number By Eliminating the Highest Common Factor Between the Numerator and Denominator.
 - Overload + Operator to Add Two Rational Number.
 - Overload >> Operator to Enable Input Through Cin.
 - Overload << Operator to Enable Output Through Cout.
 - Write a Main () to Test All the Functions in the Class.
16. Create a Class 'Complex' to Hold a Complex Number. Write a Friend Function to add Two Complex Numbers. Write a Main Function to Add Two Complex Objects.
17. Create a 'Matrix' Class of Size M X N. Overload the '+' Operator to Add Twomatrix Objects. Write a Main Function to Implement It.
18. Create a 'String' Class Which Overloads '==' Operator to Compare Two Stringobjects.
19. Create a Base Class Called 'Shape' Having
- Two Data Members of Type Double.
 - Member Function Get-Data() to Initialize Base Class Data Members.
 - Pure Virtual Member Function Display-Area() to Compute and Display the Area of the Geometrical Object.

BBCS202T: DISCRETE MATHEMATICS

Unit I

(10Lectures)

Logic: Propositional equivalence, predicates and quantifiers, Methods of proofs, proof strategy, sequences and summation, mathematical induction, recursive definitions and structural induction, program correctness.

Unit-2

(10Lectures)

Counting: The basics of counting, the pigeonhole principle, permutations and combinations, recurrence relations, solving recurrence relations, generating functions, inclusion-exclusion principle, application of inclusion-exclusion.

Unit -3

(16Lectures)

Relations: Relations and their properties, n-array relations and their applications, representing relations, closure of relations, equivalence of relations, partial orderings.

Graph theory: Introduction to graphs, graph terminology, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths, planar graphs, graph coloring, introduction to trees, application of trees.

Unit -4

(12Lectures)

Group theory: Groups, subgroups, generators and evaluation of powers, cosets and Lagrange's theorem, permutation groups and Burnside's theorem, isomorphism, automorphisms, homomorphism and normal subgroups, rings, integral domains and fields.

Unit -5

(12Lectures)

Lattice theory: Lattices and algebras systems, principles of duality, basic properties of algebraic systems defined by lattices, distributive and complimented lattices, Boolean lattices and Boolean algebras, uniqueness of finite Boolean expressions, propositional calculus. Coding theory: Coding of binary information and error detection, decoding and error correction.

Text Books:

1) K.H. Rosen: Discrete Mathematics and its application, 5th edition, Tata McGraw Hill. Chapter 1 (1.1-1.5), Chapter 3 (3.1-3.4, 3.6), Chapter 4 (4.1-4.3, 4.5), Chapter

6 (6.1, 6.2, 6.4-6.6) Chapter 7 (7.1-7.6), Chapter 8 (8.1-8.5, 8.7, 8.8)

2. C.L. Liu: Elements of Discrete Mathematics, 2nd edition, TMH 2000. Chapter 11 (11.1-11.10 except 11.7), Chapter 12 (12.1-12.8)

2.3. B. Kalman: Discrete Mathematical Structure, 3rd edition, Chapter 1 (11.1, 11.2)

BBCS301T :Object Oriented Concepts using JAVA

Unit-I

(8Lectures)

Fundamentals of Object-Oriented Programming: Introduction, Object Oriented Paradigm, Basic Concepts of OOP, Benefits and Applications of OOP. Introduction to Java: Java Features, Java Environment, Simple Java Program, Java Program Structure, Java Tokens, Java Statements, Java Virtual Machine.

Unit-II

(14Lectures)

Java Programming Basics: Constants, Variables, Data Types, Declaration of variables, Giving values to the variable, Scope of variables, Symbolic constants, Type casting.

Operators and Expressions: Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operator, Increment and Decrement Operators, Conditional Operator, Special Operators, Mathematical functions.

Using I/O: Byte streams and character streams, predefined streams, reading console input, reading characters, strings, writing console output.

Decision Making & Branching: Simple if statement, if..else statement, nesting of if..else statement, the else..if ladder, the Switch statement.

Unit-III

(12Lectures)

Decision making & Looping- The while statement, the do statement, the for statement. Jumps in loops, Labelled loops. Class & Objects - Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The 'this' keyword, Overloading Methods, Using Objects as Parameters, Returning Objects, Recursion, Understanding 'static', Introducing 'final', Using Command-Line Arguments, Varargs : Variable-Length Arguments
Arrays and Strings: One dimensional arrays, Creating an arrays, Two dimensional arrays, Strings, Vectors, Wrapper classes.

Unit-IV

(12Lectures)

Inheritance- Inheritance Basics, Using 'super', Creating Multilevel hierarchy, Method overriding, Using Abstract Classes, Using final with Inheritance.

Packages & Interfaces - Packages, Access protection in packages, Importing Packages, Interfaces. Exception Handling - Exception Handling Fundamentals –Exception Types, Uncaught

Exceptions, Using try and catch, Multiple catch clauses, Nested try statements, throw, throws, finally, Java's built-in Exceptions.

Unit-V

(14 Lectures)

Multithreaded Programming- Introduction, Creating threads, Extending the thread class, stopping & blocking thread, Life cycle of a thread, Using thread methods, Implementing the Runnable interface.

Event and GUI programming: The Applet Class, Types of Applets, Applet Basics, Applet Architecture, An Applet Skeleton, SimpleApplet Display Methods, Requesting Repaint, The HTMLAPPLET tag. Event Handling - The delegation event model, Event Classes – ActionEvent, KeyEvent & MouseEvent Classes, Event Listener Interfaces – ActionListener,

KeyListener & MouseListener interfaces. Using the Delegation Event Model. Window Fundamentals, Working with Frame Windows, Creating a Frame Window in an Applet. Creating a Windowed Program, Displaying information within a window.

Introducing Swing – two key Swing features, components and containers, the Swing packages, a simple Swing application, event handling. Exploring Swing- JLabel, JTextField, JButton, Check boxes, Radio buttons, JList, JComboBox.

Reference Books:

- E. Balagurusamy, Programming with Java – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.
- Herbert Schildt, Java: The Complete Reference, Seventh Edition, McGraw Hill Publication.
- Herbert Schildt, Java 2 - The Complete Reference, Fifth Edition, McGraw Hill Publication.
- Cay S. Horstmann, Core Java Volume I – Fundamentals, Prentice Hall.
- Somashekara, M. T., Guru, D. S., Manjunatha, K. S., Object Oriented Programming with Java, 3rd Edition, PHI.

BBCS301P:JAVALab

List of Programs:

1. Program to accept student name and marks in three subjects. Find the total marks, average and grade (depending on the average marks).
2. A menu driven program to input two integers & an operator to perform basic arithmetic operations (+, -, *, and /) using switch case structure.
3. Program, which reads two numbers having same number of digits. The program outputs the sum of product of corresponding digits. (Hint Input 327 and 539 output $3 \times 5 + 2 \times 3 + 7 \times 9 = 84$)
4. Program to input Start and End limits and print all Fibonacci numbers between the ranges. (Use for loop)
5. Define a class named Pay with data members String name, double salary, double da, double hra, double pf, double grossSal, double netSal and methods: Pay(String n, double s) - Parameterized constructor to initialize the data members, void calculate() - to calculate the following salary components, and void display() - to display the employee name, salary and all salary components.

Dearness Allowance = 15% of salary HouseRentAllowance = 10% of salary Provident Fund = 12% of salary

 $GrossSalary = Salary + DearnessAllowance + HouseRentAllowance$ Net Salary = Gross Salary - Provident Fund

Write a main method to create object of the class and call the methods to compute and display the salary details.
6. Program to create a class DISTANCE with the data members feet and inches. Use a constructor to read the data and a member function Sum () to add two distances by using objects as method arguments and show the result. (Input and output of inches should be less than 12.)
7. Program to check whether the given array is Mirror Inverse or not.

8. Program to create a class “Matrix” that would contain integer values having varied numbers of columns for each row. Print row-wise sum.

9. Program to extract portion of character string and print extracted string. Assume that ‘n’ characters extracted starting from mth character position.

10. Program to add, remove and display elements of a Vector.

MARKU

BBCS302T:COMPUTER GRAPHICS

Unit 1: Introduction and Line Generation:- Types of computer graphics, Graphic displays: Random scan and raster scan displays, Frame buffer and video controller, Line drawing algorithms (DDA, Bresenham), Circle generating algorithms (Midpoint circle algorithm).

Unit 2: 2D Transformations and Clipping:- Basic transformations: Translation, scaling, rotation, reflection, and shearing, Matrix representation and homogeneous coordinates, Composite transformations, Windowing and clipping: Viewing pipeline, 2D clipping algorithms (Cohen-Sutherland, Liang-Barsky), polygon clipping (Sutherland-Hodgeman).

Unit 3: 3D Transformations and Viewing:- 3D geometric primitives and object representation, 3D transformations: Translation, rotation, scaling, reflection, and shearing, 3D viewing, projections (parallel and perspective), 3D clipping.

Unit 4: Curves and Surfaces:- Curve representation: Spline, B-spline, and Bezier curves, Surface representation: Quadric surfaces (spheres, ellipsoids), polygon meshes, blobby objects.

Unit 5: Hidden Lines and Surface Detection:- Algorithms: Back-face detection, depth buffer (Z-buffer), A-buffer, scan-line method, Basic illumination models: Ambient light, diffuse and specular reflection, Phong model, Color consideration, transparency, shadows.

Textbooks:

"Computer Graphics: Principles and Practice" by John F. Hughes, Andries van Dam, Morgan McGuire, et al.

"Computer Graphics with OpenGL" by Donald Hearn and M. Pauline Baker

Reference Books:

"Interactive Computer Graphics: A Top-Down Approach with WebGL" by Edward Angel and Dave Shreiner

"Fundamentals of Computer Graphics" by Peter Shirley, Michael Ashikhmin, and Steve Marschn

COMPUTER GRAPHICS LAB:-

1. Line Drawing Algorithms
2. Implement the Digital Differential Analyzer (DDA) and Bresenham's line drawing algorithms to draw lines on the screen.
3. Circle and Ellipse Drawing Algorithms
4. Implement the Midpoint circle algorithm and Midpoint ellipse algorithm to draw circles and ellipses.
5. 2D Transformations
6. Write programs to perform basic 2D transformations including translation, rotation, scaling, reflection, and shearing on a given set of points.
7. Clipping Algorithms
8. Implement Cohen-Sutherland and Liang-Barsky line clipping algorithms for clipping lines against a rectangular window.
9. Implement the Sutherland-Hodgman algorithm for polygon clipping.
10. 3D Transformations
11. Implement 3D transformations including translation, rotation, and scaling on a given set of 3D points or objects.
12. Bezier and B-Spline Curves
13. Write programs to create and manipulate Bezier curves and B-spline curves for given control points.
14. Hidden Surface Removal
15. Implement the Z-buffer algorithm and Back-face culling for hidden surface removal in 3D scenes.
16. Illumination and Shading
17. Implement basic illumination models including ambient, diffuse, and specular reflections.
18. Implement shading techniques such as flat shading, Gouraud shading, and Phong shading.
19. Texture Mapping
20. Implement basic texture mapping techniques to map a 2D image onto a 3D object's surface.
21. Introduction to OpenGL.
22. Write simple OpenGL programs to render basic geometric shapes and apply transformations and lighting.

BBCS401T : DIGITAL ELECTRONIC AND COMPUT ORGANIZATION

UNIT-I

(12Lectures)

Data Representation Data Types and Number Systems Binary Number System Octal & Hexa-Decimal Number System Fixed Point Representation 1's & 2's Complement Binary Fixed- Point Representation Arithmetic Operation On Binary Numbers Overflow & Underflow Floating Point Representation Codes ASCII EBCDIC Codes Gray Code Excess-3 & BCD Error Detection & Correcting Codes Binary Storage and Registers.

UNIT- II

(14Lectures)

Boolean Algebra and Digital Logic Circuits -Logic Gates and OR NOT Gates and Their Truth Tables NOR, NAND & XOR Gates Boolean Algebra Basic Definition and Properties Basic Boolean Law's Demorgan's Theorem Map Simplification Minimization Techniques K Map – Two Three and More Variables Maps Sum of Product & Product of Sums Don't Care Conditions Combination Circuits Half Adder & Full Adder, Full Subtractor, and Decimal Adder Code Conversion Multilevel NAND and NOR Circuits Multiplexers and Demultiplexers, RAM and ROM Working & Circuit

UNIT- III

(12Lectures)

Sequential Circuits Flip-Flops - RS D JK & T Flip-Flop Triggering in Flip Flops Analysis of Clocked Sequential Circuits State Reduction and Assignment Flip Flop Excitation Tables

Registers Counters and the Memory UNIT Shift Registers Ripple Counters and Synchronous Counters Design of Counters. Common Bus System Computer Instructions Timing and Control Instruction Cycle Memory-Reference Instructions Complete Computer Description

UNIT- IV

(10Lectures)

Design of Basis Computer Control Logic Gates Control of Registers and Memory Design of

Accumulator Logic Control of Ac Register Adder and Logic Circuit Multiple Bus Organization of Computer Memory Addressing Micro Program Programming the Basis Computer Machine

Languages Assembly Language.

UNIT- V

(12Lectures)

Input-Output Organizations- I/O Interface I/O Devices Isolated Vs Memory-Mapped I/O Synchronous & Asynchronous Data Transfer. Memory Organization - Auxiliary Memory Magnetic Drum Disk & Tape Semi-Conductor Memories Direct Memory Access (DMA) Memory Hierarchy Main Memory Auxiliary Memory Associative Memory Cache Memory Virtual Memory Address Space & Memory Space Address Mapping Page Table Page Replacement Segmentation Cache Memory Hit Ratio Mapping Techniques Memory Management Hardware.

Reference Books:

- 1.M. Morris Mano-Computer System Architecture Revised 3rd Edition Pearson Publication.
- 2.Morris Mano-Digital Logic and Computer Design.

DIGITAL ELECTRONIC Lab:-

1. Instruction Set Architecture (ISA) Design:
 2. Define a simple instruction set architecture, including instruction formats, opcode definitions, and addressing modes.
 3. Specify the instruction set's functionality, such as arithmetic and logic operations, memory access, and control flow instructions.
4. Processor Architecture Design:
 5. Design the architecture of the processor, including the datapath and control unit.
 6. Select appropriate components such as registers, ALU (Arithmetic Logic Unit), program counter, and instruction register.
7. Instruction Execution:
 8. Implement the instruction execution cycle, including instruction fetch, decode, execute, and write-back stages.
 9. Develop control logic to decode instructions and generate control signals for the datapath

10. components.

11. Memory Interface:

12. Interface the processor with memory modules for instruction and data storage.

13. Implement memory access instructions to load and store data from/to memory.

14. Testing and Debugging:

15. Test the processor design using a set of assembly language programs or machine code instructions.

16. Debug any errors or inconsistencies in the processor's behavior through simulation or hardware testing.

17. Performance Evaluation:

18. Evaluate the performance of the processor in terms of execution speed, throughput, and resource utilization.

19. Compare the performance metrics with theoretical expectations and optimize the design as needed.

20. Documentation and Presentation:

21. Document the design specifications, implementation details, and test results in a comprehensive report.

22. Prepare a presentation summarizing the project objectives, methodology, design decisions, and outcomes.

BBCS402T: Machine learning technique

UNIT-I INTRODUCTION TO ARTIFICIAL NEURAL NETWORKS (10Lectures)

Fundamentals Of Neural Networks – Model of Artificial Neuron – Neural Network Architectures – Learning Methods – Taxonomy Of Neural Network Architectures – Applications.

UNIT II FEED FORWARD NEURAL NETWORKS (12Lectures)

Perceptron Models: Discrete, Continuous and Multi-Category – Training Algorithms: Discrete and Continuous Perceptron Networks – Limitations of the Perceptron – Model. Credit Assignment Problem – Generalized Delta Rule, Derivation of Back propagation (BP) Training, and Summary of Back propagation Algorithm – Kolmogorov Theorem.

UNIT III: MACHINE LEARNING (14Lectures)

Machine Learning Fundamentals – Types of Machine Learning - Supervised, Unsupervised, Reinforcement- The Machine Learning process. Terminologies in ML- Testing ML algorithms: Overfitting, Training, Testing and Validation Sets- Confusion matrix -Accuracy metrics- ROC Curve- Basic Statistics: Averages, Variance and Covariance, The Gaussian- The Bias-Variance trade off- Applications of Machine Learning.

UNIT IV: SUPERVISED LEARNING (12Lectures)

Regression: Linear Regression – Multivariate Regression- Classification: Linear Discriminant Analysis, Logistic Regression- K-Nearest Neighbor classifier. Decision Tree based methods for classification and Regression- Ensemble methods.

UNIT V: UNSUPERVISED LEARNING**(12Lectures)**

Clustering- K-Means clustering, Hierarchical clustering - The Curse of Dimensionality - Dimensionality Reduction - Principal Component Analysis - Probabilistic PCA- Independent Components analysis.

TEXT BOOKS

1. CharuC.Aggarwal “Neural Networks and Deep learning” Springer International Publishing,2018
2. Satish Kumar, “Neural Networks, A Classroom Approach”, Tata McGraw -Hill, 2007.
3. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
4. Stephen Marsland, “Machine Learning –An Algorithmic Perspective”, CRC Press, 2009.
5. Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, “Machine Learning”, Pearson Education, 2018.
6. Christopher Bishop, “Pattern Recognition and Machine Learning” Springer, 2011.

REFERENCE BOOKS

1. Andreas C. Muller, “Introduction to Machine Learning with Python: A Guide for DataScientists”, O'Reilly,2016.
2. Sebastian Raschka, “Python Machine Learning”, Packt Publishing, 2015.
3. Hastie, Tibshirani, Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd Edition, Springer, 2017.
4. Ethem Alpaydin, “Introduction to Machine Learning”, 2nd Revised edition, MIT Press,2010.

LIST OF EXPERIMENTS

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a

given set of training data samples. Read the training data from a .CSV file.

2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

REFERENCES

1. Christopher Bishop, "Pattern Recognition and Machine Learning" Springer, 2007.

2. Stephen Marsland, “Machine Learning – An Algorithmic Perspective”, Chapman and Hall, CRC Press, Second Edition, 2014.
3. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
4. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, Third Edition, 2014.
5. Tom Mitchell, “Machine Learning”, McGraw-Hill, 1997.

MARKU

BDSC501T:Python Programming

Unit-I

(10Lectures)

PythonBasics,Objects-PythonObjects,StandardTypes,OtherBuilt-inTypes,InternalTypes, StandardTypeOperators,StandardTypeBuilt-inFunctions,CategorizingtheStandardTypes, Unsupported Types Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types.

Unit-II

(16Lectures)

FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-lineArguments, File System, File Execution, Persistent StorageModules,RelatedModulesExceptions:ExceptionsinPython,DetectingandHandling Exceptions, Context Management, *Exceptions as Strings, Raising Exceptions, Assertions, StandardExceptions,*CreatingExceptions,WhyExceptions(Now)?,WhyExceptionsatAll?, Exceptions and the sys Module, Related Modules Modules: Modules and Files, Namespaces, ImportingModules,ImportingModuleAttributes,ModuleBuilt-inFunctions,Packages,Other Features of Modules.

Unit-III

(12Lectures)

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules.

Unit-IV

(12Lectures)

GUIProgramming:Introduction,TkinterandPythonProgramming,BriefTourofOtherGUIs, RelatedModulesandOtherGUIsWEBProgramming:Introduction,WedSurfingwithPython, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGIApplicationAdvanced CGI, Web (HTTP) Servers.

Unit-V

(10Lectures)

Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules.

ReferenceBooks:

- CorePythonProgramming, WesleyJ.Chun, SecondEdition, Pearson.
- ThinkPython, AllenDowney, GreenTeaPress
- IntroductiontoPython, KennethA.Lambert, Cengage
- PythonProgramming:AModernApproach, VamsiKurama, Pearson

- LearningPython, MarkLutz, O'Really.

BDSC501P:PythonProgrammingLab

List ofPrograms:

1. Writeaprogramtodemonstratedifferentnumberdatatypesin Python.
2. WriteaprogramtoperformdifferentArithmeticOperationsonnumbersin Python.
3. Writeaprogramtcreate,concatenateandprintastringandaccessingsub-stringfrom a given string.
4. Write a python script toprint the current date inthe following format “Sun May 2902:
5. 26:23 IST 2017”
6. Writeaprogramtcreate,append,andremovelistsin python.
7. Writeaprogramtodemonstrateworkingwithtuplesinpython.
8. Writeaprogramtodemonstrateworkingwithdictionariesinpython.
9. Writeapythonprogramtofindlargestofthreenumbers.
10. WriteaPythonprogramtoconverttemperaturestoandfromCelsius,Fahrenheit. [Formula :
 $c/5 = f-32/9$]
11. WriteaPythonprogramtoconstructthefollowingpattern,usinganested forloop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
* *
*
```

12. Write a Python script that prints prime numbers less than 20.
13. Write a Python program to find factorial of a number using Recursion.
14. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
15. Write a Python program to define a module to find Fibonacci Numbers and import the module to another program.
16. Write a Python program to define a module and import a specific function in that module to another program.
17. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
18. Write a program that inputs a text file. The program should print all of the unique words in the
19. File in alphabetical order.
20. etc.
21. Write a Python class to convert an integer to a Roman numeral.
22. Write a Python class to implement $\text{pow}(x, n)$.
23. Write a Python class to reverse a string word by word.

Reference Books:

- Core Python Programming, Wesley J. Chun, Second Edition, Pearson.
- Think Python, Allen Downey, Green Tea Press
- Introduction to Python, Kenneth A. Lambert, Cengage
- Python Programming: A Modern Approach, Vamsi Kurama, Pearson
- Learning Python, Mark Lutz, O'Really.

BDSC601T: Web Programming and PHP

UNIT I: HTML – (12Lectures)
General Introduction to Internet and WWW; **HTML:** Structured language, Document types, Rules of html, Html tags, Head tags, Body tags, Headings, Divisions and Centering, Quotations, Preformatted text, Lists, Horizontal Rules, Block level elements, Text level elements, Character entities, Comments, Fonts, Tables: Table tags, Colors, Color names, Color values, Marquee tag.

UNIT II Advanced HTML- (10Lectures)
Linking in html: Anchor tags, Layer tags, Link relationships, URL: Relative, Absolute, Image, Image maps, Frames: Layouts, Targeting, No frame tag, Floating frames, Audio, Embed tag, Forms: form tag and its attributes.

UNIT III: CSS: (12Lectures)
Style sheet Basics, Adding Style to a Document, CSS (Cascading Style Sheet) and HTML Elements, Selectors, Document Structure and Inheritance. **Javascript** - Introduction to Javascript, Variables and data types, Declaring Variables, Operators, Control Structures, Conditional Statements, Loop Statements, Functions, Objects, Dialog Boxes, Alert Boxes, Confirm Boxes, Prompt Boxes, Javascript with HTML, Events, Introduction to DHTML-Components, Uses, Features.

UNIT III: PHP: (10Lectures)
Overview of PHP, The building blocks of PHP: variables, globals & superglobals
Data types: Set type, type casting, test type, Operators & Expressions, Flow control functions in PHP, Functions: Defining a function variable scope, calling a function returning values, setting default values for arguments, passing variable reference, built in functions

UNIT IV: Arrays: (16Lectures)
Creating arrays (associative & multidimensional), Array related functions. Working with Objects, Working with string functions: Formatting strings, Using Date and Time functions. Forms in PHP: Form elements, adding elements to a form, creating a simple input form, combining HTML & PHP code on a single page, redirecting the user, creating a send mail form, File upload form, working with files and directories. **Cookies:** Introduction, different types of cookies, setting a cookie with PHP, deleting a cookie, session function overview: starting a session, working with session variables.

CORE TEXTBOOK:

1. V.K.Jain, Advanced Programming in Web Design, CyberTech Publications
2. Julie C. Meloni, PHP, MySQL and Apache, Pearson Education

ADDITIONAL REFERENCES:

1. Ivan Bayross "Web Enabled Commercial Application Development: HTML, D

BDSC601P: Web Programming and PHP lab

(HTML/ JAVASCRIPT)

1. Practicing basic HTML tags, text tags, text styles, paragraph styles, headings, lists
2. Tables in HTML, Frames in HTML, nested frames, Link and Anchor Tags
3. Including graphics, video and sound in web pages, including Java applets
4. Layers & Image Maps
5. Creating animated Gifs
6. Cascading Style sheets
7. DHTML
8. HTML forms and Fields
9. Exercises covering basic introduction to JavaScript
10. Development of a website involving a variety of tools practiced above

(PHP) Sample Questions

1. *Setup WAMP/XAMPP Server or Setup Apache, MySQL and PHP separately in your PHP Lab.*
2. Write a PHP program to generate a random number between 1 and 100.
3. Modify above program to accept range of the random number from HTML interface.
4. Programs involving various control structures like if, else, elseif/elseif", Alter native Syntax for 'if, else, elseif/else if'
5. Programs involving various control structures like while, do-while, for, foreach, switch, break, continue. Try alternative syntax for while, do-while, for, foreach, switch.
6. Programs involving the declare, return, require, include, require-once, include_once and goto.
7. Program to demonstrate PHP Array functions, PHP Array Sorting, PHP Key Sorting, PHP Value Sorting, PHP Multi Array Sorting, PHP Array Random Sorting,
8. Program to demonstrate PHP Array functions. PHP Array Reverse Sort.
9. ng, Array to String Conversion, Implode() function, String to Array, Array Count, Remove Duplicate Values
10. Program to demonstrate PHP Array functions. arraySearch, Array Replace, Array Replace Recursive, Array Sub String Search
11. Demonstrate the use of regular expression to compare two strings.
12. Extract Domain name from URL
13. Find the number of rows from a MySQL database for your query.
14. Generate a Guestbook which will allow your website visitor to enter some simple data about your website.
15. Write a PHP script to find whether the given number is a prime or not.
16. Write a PHP script to demonstrate the use of break, continue statements using nested loops.
17. Write a PHP script to display the Fibonacci sequence with HTML page.
18. Write a PHP script to create a chessboard.
19. Write a PHP script using built-in string function like strpos(), strpos(), substr_count(), etc
20. Write a PHP script to transform a string to uppercase, lowercase letters, make a string's first character uppercase.
21. Write a PHP function to check whether all array values are strings or not.
22. Write a PHP script to count number of elements in an array and display a range of array elements.

BDSC701T: Assembly language Programming

Unit-I (10Lectures)

Computer Architecture: Processor to memory communication, processor to I/O device communication, Instruction Format; **Instruction Cycle:** Fetch Cycle, Execution Cycle; Instruction Set: CISC Architecture, RISC Architecture, Comparison; Memory Chips; Pipelining and Parallel Processing;

Unit-II (10Lectures)

Input / Output Organization: Asynchronous Data Transfer, Programmed I/O (concepts only); DMA: DMA Controller, DMA Transfer Modes; I/O Processor.

Unit-III (18Lectures)

The 8086 Microprocessor: 8086 Internal Architecture: Execution Unit, Bus Interface Unit, Multiplexing of address/ data bus, 8086 registers, **Memory bank:** Even bank, Odd Bank, Pins and Signals, Bus cycles, Direct Memory Access, DMA Transfer modes.

Unit-IV (12Lectures)

Instruction set of 8086: Data transfer, Arithmetic, Bit manipulation, string, Branch control, Iteration control and Processor control, Addressing modes, **Interrupts:** Interrupt Vector Table ,Response, Types, Priority.

Unit-V (12Lectures)

Assembly Language Programming: Program development tools, variables and constants used in assembler,Assembler directives, Procedures and macros, Interrupts of personal computers, Hand coding of assembly language programs, examples.

TEXTBOOKS

1. NMathivanan, Microprocessors, PCHardwareandInterfacing,PHI Edition.
2. IntroductiontoInformationTechnology,2ndEdition, ITLEducationolutionslimited,Pearson.

REFERENCES

1. ANagoorKani,8086Microprocessorandit'sapplications,McGrawhill,secondedition.
2. JohnDCarpinelli,Computersystemorganizationandarchitecture,PearsonEducation.

BDSC701P: Assembly language Programming lab

1. Arithmetic operations
 2. Data movement operations
 3. Control operations
 4. Input-output operations
 5. String operations
 6. Logical operations
1. Program to move a block of data from one location to another
 2. Program to find the presence of a given value in a block of data
 3. Program for reverse a string
 4. Program to find the number of alphabets in a word

MARKU

BBCS702T: Software Engineering

Unit-I

(12Lectures)

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models. Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

Unit-II

(10Lectures)

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. System models: Context models, behavioral models, data models, object models, structured methods.

Unit-III

(18Lectures)

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagram.

Unit-IV

(1Lecture)

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of

debugging. Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

Unit-IV

(12Lectures)

Metrics for Process and Products: Software measurement, metrics for software quality. Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

ReferenceBooks:

- Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
- Software Engineering- Sommerville, 7th edition, Pearson Education.
- The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.
- Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
- Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
- Fundamentals of object-oriented design using UML Meilerpage- Jones: Pearson Education.

Software Engineering lab:-

1. Online Voting System:
2. Develop an online voting system for conducting elections securely over the internet.
3. Implement features such as user authentication, ballot creation, voting, and result tabulation.
4. Ensure security, integrity, and anonymity of votes through encryption and authentication mechanisms.
5. E-commerce Platform:
6. Build an e-commerce platform for buying and selling products online.
7. Include features such as user registration, product catalog, shopping cart, checkout, and payment processing.
8. Implement search and recommendation algorithms to personalize the shopping experience.
9. Hospital Management System:
10. Develop a hospital management system to manage patient records, appointments, and medical services.
11. Include modules for patient registration, appointment scheduling, billing, and inventory management.
12. Ensure compliance with healthcare regulations and standards for data security and privacy.
13. Learning Management System (LMS):
14. Create a learning management system for delivering online courses and educational content.
15. Implement features such as course creation, enrollment, progress tracking, and assessment.
16. Support multimedia content, discussion forums, and collaboration tools for interactive learning experiences.
17. Social Networking Platform:
18. Build a social networking platform for connecting users, sharing content, and fostering online communities.
19. Include features such as user profiles, friend connections, news feed, messaging, and notifications.
20. Implement privacy settings and content moderation tools to ensure a safe and inclusive environment.
21. Task Management Application:
22. Develop a task management application for organizing and tracking tasks within teams or projects.
23. Implement features such as task creation, assignment, priority setting, and status tracking.
24. Support collaboration and communication among team members through comments, file attachments, and notifications.
25. Customer Relationship Management (CRM) System:
26. Create a CRM system for managing customer interactions, sales leads, and marketing campaigns.
27. Include modules for contact management, sales pipeline tracking, lead generation, and customer support.
28. Integrate with email marketing tools, analytics platforms, and customer service channels for comprehensive CRM functionality.
29. Inventory Management System:
30. Build an inventory management system for tracking stock levels, orders, and shipments.
31. Implement features such as inventory tracking, reorder management, stock alerts, and reporting.
32. Integrate with barcode scanners, RFID technology, and ERP systems for seamless inventory management operations.
33. Project Management Tool:

34. Develop a project management tool for planning, tracking, and coordinating project activities.
35. Include features such as task scheduling, resource allocation, progress monitoring, and milestone tracking.
36. Support different project management methodologies (e.g., Agile, Waterfall) and collaboration workflows.
37. Real Estate Management System.

MARKU

BBCS801T: FREE AND OPEN SOURCE SOFTWARES (FOSS)

Unit-I

(12Lectures)

Open source software: concepts, features, benefits over proprietary software, examples, Free software: concepts, features, advantages, Free software Vs Open Source software, Free and Open Source Software (FOSS), Four essential freedoms, Free software movements, free software foundation(FSF), history, policies, GPL, free operating systems

Unit-II

(12Lectures)

Linux: features, history, various Linux distributions, Linux architecture, kernel and shell, Linux desktop environments- GNOME and KDE, Linux File System and Directories, types of files, Installing and Configuring Linux, File access permissions, understanding system administration

Unit III:

(16Lectures)

Linux Commands: date, time, who, echo, man, info, cal, pwd, more, less, head, tail, mkdir, cd, cp, mv, rm, touch, sort, wc, cut, cat with options, ls with options ,grep with options, chmod mounting the file system, command line processing etc. Types of editors in Linux, Introduction to vi editor, modes in vi editor, common vi editor commands for open a file, save a file, delete a file, quit a file etc

Unit IV:

(10Lectures)

Shell Programming: Introduction to Linux Shell and Shell Scripting, types of shells in Linux, steps in creating a shell script, write and run shell scripts, shell variables and printing.

Unit-V

(10Lectures)

Shell programming arrays: deleting shell variables, script permissions and names, scripting guidelines, adding shell comments, shell variables, constants, operators, conditional statements, looping statements, using arrays, shell functions, files, shell built-in functions

CORE TEXTS

1. Christopher Negus, *Red Hat Linux 9 Bible*, WILEY-Dreamtech, New Delhi,
2. Thomas Schenk, *Red Hat Linux System Administration*, Techmedia, New Delhi, 2003

ADDITIONAL REFERENCES

- https://www.tutorialspoint.com/unix/shell_scripting.htm

FREE AND OPEN SOURCE LAB:

1. Content Management System (CMS) Customization:
2. Customize a popular FOSS CMS (e.g., WordPress, Joomla, Drupal) to meet specific requirements for a website or web application.
3. Modify themes, plugins, and functionality to tailor the CMS to the needs of a particular organization or project.
4. E-commerce Platform Extension:
5. Extend an existing FOSS e-commerce platform (e.g., WooCommerce, Magento, OpenCart) with custom features or integrations.
6. Develop new payment gateways, shipping methods, product types, or marketing tools to enhance the platform's functionality.
7. Educational Software Development:
8. Develop FOSS educational software for teaching and learning purposes, such as interactive simulations, programming environments, or digital textbooks.
9. Collaborate with educators and subject matter experts to design and implement educational tools that address specific learning objectives.
10. Healthcare Information System Enhancement:
11. Contribute to the development and improvement of FOSS healthcare information systems (e.g., OpenMRS, GNU Health) by implementing new modules, features, or integrations.
12. Address usability, interoperability, and security issues to enhance the usability and effectiveness of the healthcare software.
13. GIS (Geographic Information System) Application Development:
14. Develop FOSS GIS applications or plugins for spatial data analysis, visualization, and mapping.
15. Implement features such as geocoding, routing, spatial analysis algorithms, and custom map layers to support geographic data processing tasks.
16. Scientific Computing Library Expansion:
17. Extend existing FOSS scientific computing libraries (e.g., NumPy, SciPy, matplotlib) with new algorithms, data structures, or optimization techniques.
18. Contribute to community-driven projects to improve performance, reliability, and usability of scientific computing tools.
19. Collaborative Text Editor Integration:
20. Integrate a FOSS collaborative text editor (e.g., Etherpad, OnlyOffice, CKEditor) with other tools or platforms for document collaboration and editing.
21. Develop plugins, connectors, or APIs to enable seamless integration with project management systems, content .
22. repositories, or communication tools.
23. Community Engagement Platform Development:
24. Build a FOSS community engagement platform for managing discussions, contributions, and interactions within open source projects or organizations.
25. Implement features such as forums, mailing lists, issue trackers, and contributor recognition tools to foster collaboration and communication.
26. Accessibility Tool Enhancement:
27. Contribute to FOSS accessibility tools (e.g., screen readers, magnifiers, voice recognition software) by improving usability, compatibility, and functionality.
28. Address accessibility barriers and compliance requirements to make software applications more inclusive and accessible to users with disabilities.
29. Data Visualization Toolkit Expansion:

30. Extend FOSS data visualization libraries (e.g., D3.js, Plotly, Vega) with new chart types, interactive features, or data processing capabilities.
31. Develop plugins, extensions, or templates to simplify the creation of data-driven visualizations for web applications, dashboards, and reports.

MARKU

BBCS802T: MOBILE APPLICATION DEVELOPMENT

Unit 1: Introduction to Mobile Application Development:- Overview of mobile platforms: iOS, Android, cross-platform development, Introduction to mobile development frameworks: Swift for iOS, Kotlin/Java for Android, React Native, Flutter, Setting up development environments: Xcode for iOS, Android Studio for Android, IDEs for cross-platform development.

Unit 2: User Interface Design for Mobile Applications:- Principles of mobile UI/UX design: simplicity, consistency, responsiveness, Mobile UI components: layouts, views, controls, navigation patterns, Designing adaptive layouts for different screen sizes and orientations, Implementing UI design using Interface Builder (iOS) and XML layout files (Android).

Unit 3: Mobile Application Development Basics:- Introduction to mobile development concepts: activities (Android), view controllers (iOS), state management, Working with user input: handling touch events, gestures, text input, Data persistence: storing data locally (SQLite, SharedPreferences, Core Data), accessing remote data (RESTful APIs).

Unit 4: Advanced Mobile Application Development Topics:- Multi-screen navigation: navigation controllers (iOS), fragments (Android), routing (React Native, Flutter), Integrating device features: camera, GPS, accelerometer, sensors, Authentication and security: user authentication, authorization, encryption, secure communication.

Unit 5: Mobile Application Deployment and Distribution:- App deployment process: app store submission guidelines, provisioning profiles (iOS), signing keys (Android), Testing and debugging mobile applications: unit testing, UI testing, device testing, beta testing, App distribution channels: app stores (Apple App Store, Google Play Store), enterprise ,distribution, over-the-air (OTA) updates.

Recommended Book:

"Programming Mobile Applications for Android and iOS" by O'Reilly Media. This book provides a comprehensive introduction to mobile application development for both Android and iOS platforms. It covers essential concepts, best practices, and practical examples to help students learn mobile development techniques effectively. Additionally, "Flutter in Action" by Eric Windmill is a practical guide to building mobile applications with Flutter, Google's UI toolkit for

building natively compiled applications for mobile, web, and desktop from a single codebase.

MOBILE APPLICATION LAB:-

1. Fitness Tracking App:
2. Develop a mobile app that tracks users' fitness activities such as running, cycling, and workouts.
3. Implement features such as GPS tracking, activity logging, calorie counting, and goal setting.

4. Provide visualizations and progress tracking to motivate users and help them achieve their fitness goals.
5. Recipe Sharing and Meal Planning App:
 6. Create a mobile app for sharing recipes, meal planning, and grocery shopping.
 7. Allow users to browse and search for recipes, create meal plans, and generate shopping lists based on selected recipes.
 8. Include features for saving favorite recipes, customizing meal plans, and sharing recipes with friends and family.
9. Language Learning App:
 10. Develop a mobile app for learning languages through interactive lessons, quizzes, and vocabulary exercises.
 11. Offer features such as flashcards, audio pronunciation, grammar explanations, and progress tracking.
 12. Support multiple languages and customizable learning paths based on users' proficiency levels and learning goals.
13. Event Planning and Management App:
 14. Build a mobile app for organizing and managing events such as parties, meetings, and conferences.
 15. Allow users to create event invitations, send RSVPs, manage guest lists, and track event details and schedules.
 16. Provide features for event discovery, location mapping, reminders, and real-time communication among attendees.
17. Task Management and Productivity App:
 18. Create a mobile app for organizing tasks, setting reminders, and improving productivity.
 19. Implement features such as task lists, deadlines, priority levels, and task categorization.
 20. Offer integration with calendar apps, notification reminders, and synchronization across multiple devices.

Department Specific Elective

BBCS101D-I : Digital marketing analysis

UNIT 1: DIGITAL MEDIA AND ANAL

Digital media types – Owned and earned social metrics – Paid searches and Organic Searches - Aligning Digital and Traditional Analytics – Identifying social media listening tools – Understanding social media engagement software – Social media engagement tools.

UNIT 2: TOOLS FOR DIGITAL ANALYTICS

Social Media Listening Tools - Evolution, Social analytics life cycle, Social media monitoring software: Sysomos, Radian6, Visible Technologies, Zoho social and others. Search Analytics Tools – Basics of search, Search analytics use cases, Search data, Google trends, YouTube trends, Google Adwords keyword, Yahoo clues, Collecting insights through search data.

Audience Analysis Tools – Audience Analysis Use Cases, Audience analysis tool types –

Audience analysis Techniques, Event Triggers. Content Analysis Tools - Content Audits- Optimizing

Content Distribution, Analysing Content Consumption. Engagement Analysis Tools–Social Media

Engagement Software (SMES), using SMES, study of different SMES in the market.

UNIT 3: DIGITAL INFLUENCE AND LISTENING

Reality of Digital Influence - Media List - Klout, PeerIndex - Online Versus Offline Influence - Using

the Influencer List - Developing Social Media Listening Program - Using Listening Data for Program

Planning - Implementing Listening Program - Conversation Audit - Online Influencers - Conducting

Social brand benchmarking - Use of Online data for crisis anticipation-Identifying known issues - Crisis

day monitoring and ongoing reporting - Corrections after crisis - Improving customer service – Social

customer service conflict - Social customer service models.

UNIT 4: RESEARCH PLAN AND SEARCH ANALYSIS

Launching new product – Product life cycle – Introduction Phase – Growth Phase – Maturity Phase.

Formulating research plan – Developing source list – Research methods – Constructing reports –

Delivering reports – Report use cases – Building central repository of information – Search analytics for digital strategy – Search analytics for content strategy and planning – Search analytics for paid advertising.

UNIT 5: ROI, MOBILE ANALYTICS AND BUSINESS INTELLIGENCE

Return on Investment (ROI) – Return on Engagement, Influence, Experience – Tracking ROI

Understanding measurement fundamentals – Measurement reporting cadence - Mobile Analytics –

Mobile market landscape – Mobile marketing measurement – Marketing activities – Audience/visitor

metric – Mobile app performance - Social CRM – Social CRM initiative – Social CRM Initiative –

Future of Digital Data – Business Intelligence.

TEXT BOOKS:

1.Chuck Hemann and Ken Burbary, “Digital Marketing Analytics: Making Sense of Consumer Data in a Digital World”, Que Publishing, 2nd edition, 2018.

REFERENCE BOOKS:

1.Simon Kingsnorth, “Digital Marketing Strategy: An Integrated Approach Marketing”, Kogan Page Publisher, First edition, ISBN-13: 978-0749474706, 2016.

2.Dave Chaffey, Fiona Ellis-Chadwick, “Digital Marketing – Strategy, Implementation and Practice”, Pearson Education, Sixth edition, ISBN-13: 978-1292077611, 2016.

E BOOKS

1.Eric Enge, Andy Crestodina, Larry Kim, Steve Rayson and Chad White, “How the Pros Turn Marketing Analytics Into Effective Marketing Strategies”, Alexa, An Amazon Company.
<https://blog.alexa.com/wp-content/uploads/2016/12/How-to-Pros-Turn-Marketing-Analytics-into-Effective-Marketing-Strategies-ebook.pdf>

MOOC

1.<https://www.coursera.org/learn/marketing-analytic>

BBCS101D-II: Mobile Computing

Unit 1: Introduction to Mobile Computing:- Overview of mobile computing: definition, history, evolution, and significance, Mobile computing architecture: mobile devices, communication networks, middleware, and applications, Mobile operating systems: iOS, Android, Windows Mobile, and others, Mobile application development frameworks: native, web-based, and hybrid approaches.

Unit 2: Mobile Networking and Communication:- Wireless communication technologies: GSM, CDMA, LTE, Wi-Fi, Bluetooth, NFC, Mobile network architectures: cellular networks, ad hoc networks, satellite networks, Mobile IP (IPv4 and IPv6): addressing, routing, and mobility management, Mobile communication protocols: TCP, UDP, HTTP, SMS, MMS.

Unit 3: Mobile Application Development:- Principles of mobile application development: user interface design, responsiveness, power efficiency, Mobile application development frameworks: Android SDK, iOS SDK, Xamarin, React Native, User interface design patterns: navigation patterns, layout design, responsive design, Mobile application testing and debugging: unit testing, UI testing, device testing, emulation, and simulation.

Unit 4: Location-Based Services and Mobile Commerce:- Location-based services (LBS): GPS, A-GPS, location sensing techniques, Mobile commerce (m-commerce): mobile payment systems, mobile banking, mobile shopping, Mobile advertising and location-based marketing: targeted advertising, proximity marketing, Security and privacy issues in location-based services and mobile commerce.

Unit 5: Mobile Security and Privacy:- Mobile device security: device encryption, screen lock, biometric authentication, Mobile application security: secure coding practices, data encryption, secure communication, Mobile malware and vulnerabilities: types of mobile threats, prevention techniques, Privacy concerns in mobile computing: data collection, tracking, consent, and regulation.

Recommended Book:

"Mobile Computing" by Raj Kamal. This book provides a comprehensive overview of mobile computing concepts, technologies, and applications. It covers topics such as mobile communication systems, mobile computing architecture, mobile application development, location-based services, mobile security, and privacy. With clear explanations and real-world examples, it serves as a valuable resource for students and practitioners interested in mobile computing.

BBCS102D-I: Distribution Computing

Unit 1: Introduction to Distribution Computing:- Overview of distributed systems, Motivation for distributed computing, Characteristics and challenges of distributed computing, Types of distributed systems: client-server, peer-to-peer, distributed databases, etc. Introduction to parallelism and concurrency.

Unit 2: Communication in Distributed Systems:- Communication models: synchronous vs. asynchronous, message passing, remote procedure calls (RPC), etc. Network protocols and middleware, Socket programming, Marshalling and unmarshalling.

Unit 3: Distributed Computing Models:- Overview of distributed computing models: MapReduce, actor model, tuple spaces, etc. Principles of distributed algorithms, Consistency models: eventual consistency, strong consistency, etc. Fault tolerance and replication strategies.

Unit 4: Distributed Systems Design:- Design principles for distributed systems, Scalability, availability, and reliability, Distributed system architectures: layered architectures, microservices, etc. Case studies of distributed systems design.

Unit 5: Advanced Topics in Distribution Computing:- Distributed storage systems: distributed file systems, key-value stores, etc. Distributed computing platforms: Apache Hadoop, Apache Spark, etc. Security and privacy in distributed systems, Emerging trends and future directions in distributed computing.

Reference Book:

"Distributed Systems: Principles and Paradigms" by Andrew S. Tanenbaum and Maarten Van Steen

"Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall, "Distributed Algorithms" by Nancy A. Lynch, "Designing Data-Intensive Applications" by Martin Kleppmann, "Distributed Computing: Principles, Algorithms, and Systems" by Ajay D. Kshemkalyani and Mukesh Singhal

BBCS102D-II: Information Security

Unit 1: Introduction to Information Security:- Definition and scope of information security, Fundamental principles: confidentiality, integrity, availability (CIA triad), Threats, vulnerabilities, and attacks, Risk management and risk assessment.

Unit 2: Cryptography and Cryptographic Protocols:- Basics of cryptography: encryption, decryption, symmetric and asymmetric encryption, Cryptographic algorithms: DES, AES, RSA, etc. Hash functions and digital signatures, Public key infrastructure (PKI), Cryptographic protocols: SSL/TLS, SSH, IPsec, etc.

Unit 3: Network Security:- Network threats and attacks: malware, phishing, DDoS, etc. Firewalls and intrusion detection/prevention systems (IDS/IPS), Secure communication protocols: HTTPS, VPN, etc. Wireless network security: WPA, WPA2, WPA3, etc. Network security best practices.

Unit 4: Security Management and Governance:- Security policies, standards, and procedures, Security governance frameworks: ISO 27001, NIST Cybersecurity Framework, etc. Compliance and regulatory requirements, Security awareness and training, Incident response and disaster recovery planning.

Unit 5: Emerging Trends in Information Security:- Cloud security: challenges and best practices, Mobile security: BYOD, mobile device management (MDM), etc. Internet of Things (IoT) security, Big data security and analytics. Artificial Intelligence (AI) and Machine Learning (ML) in cybersecurity.

Reference Book:

"Cryptography and Network Security: Principles and Practices" by William Stallings,

"Network Security Essentials: Applications and Standards" by William Stallings, "Information Security Management Principles" by David Alexander, Maira Sutton, and Andy Taylor, "Cybersecurity: Managing Risk in the Information Age" by Domenic Antonucci, J.R. Simpson, and Anthony Vance

MARKU

BBCS103D-I: Soft Computing

Unit 1: Introduction to Soft Computing:- Overview of soft computing paradigms: fuzzy logic, neural networks, genetic algorithms, and evolutionary computation, Characteristics and advantages of soft computing approaches, Applications of soft computing in various domains.

Unit 2: Fuzzy Logic:- Basics of fuzzy set theory: membership functions, fuzzy operators Fuzzy inference systems: Mamdani and Sugeno models, Fuzzy logic applications in control systems, decision-making, and pattern recognition

Unit 3: Neural Networks:- Introduction to artificial neural networks (ANNs), Types of neural networks: feedforward, recurrent, convolutional, etc. Training algorithms: backpropagation, gradient descent, etc. Neural network applications in classification, regression, and optimization.

Unit 4: Genetic Algorithms:- Basics of genetic algorithms (GAs): representation, selection, crossover, mutation, Genetic programming (GP) and evolutionary strategies (ES), Optimization problems and solution techniques using genetic algorithms, Genetic algorithm applications in function optimization, scheduling, and machine learning.

Unit 5: Hybrid and Emerging Soft Computing Techniques:- Hybridization of soft computing techniques, Integration of soft computing with other computational intelligence methods, Swarm intelligence: particle swarm optimization (PSO), ant colony optimization (ACO), etc.
Bio-inspired computing: artificial immune systems, memetic algorithms, etc.

Reference Book:

"Soft Computing: Techniques and its Applications in Electrical Engineering" by Sivanandam S.N. and Deepa S.N. "Fuzzy Logic with Engineering Applications" by Timothy J. Ross , "Neural Networks: A Comprehensive Foundation" by Simon Haykin, "Introduction to Genetic Algorithms" by Melanie Mitchell, "Hybrid Intelligent Systems" by Ajith Abraham, Crina Grosan, and Witold Pedrycz.

BBCS103D-II: Statistics and Data Analytics

Unit 1: Introduction to Statistics and Data Analysis:- Basic concepts in statistics: population vs. sample, descriptive vs. inferential statistics, Data types: qualitative vs. quantitative data, Data visualization techniques: histograms, scatter plots, box plots, etc. Summary statistics: mean, median, mode, variance, standard deviation.

Unit 2: Probability Theory:- Fundamentals of probability: sample spaces, events, probability axioms, Conditional probability and independence, Random variables and probability distributions: discrete and continuous distributions, Expected value and variance.

Unit 3: Statistical Inference:- Estimation: point estimation, interval estimation, Hypothesis testing: null and alternative hypotheses, Type I and Type II errors, significance level, p-value Parametric and non-parametric tests, Confidence intervals and hypothesis tests for means, proportions, variances, etc.

Unit 4: Regression Analysis:- Simple linear regression: model, estimation, inference, diagnostics, Multiple linear regression: model, estimation, inference, diagnostics Regression assumptions and violations, Model selection techniques: forward selection, backward elimination, etc.

Unit 5: Data Analytics and Machine Learning:- Introduction to data analytics process: data preparation, exploratory data analysis (EDA), modeling, evaluation, deployment,

Supervised learning techniques: classification, regression, Unsupervised learning techniques: clustering, dimensionality reduction, Introduction to popular machine learning algorithms: decision trees, random forests, support vector machines (SVM), k-nearest neighbors (k-NN), etc.

Reference Book:

"Statistics for Business and Economics" by Paul Newbold, William L. Carlson, and Betty Thorne,
"Probability and Statistics for Engineering and the Sciences" by Jay L. Devore, "Statistical Inference" by George Casella and Roger L. Berger, "An Introduction to Statistical Learning: with Applications in R" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, "Python for Data Analysis" by Wes McKinney

BBCS104D-I: Data Mining

Unit 1: Introduction to Data Mining:- Overview of data mining concepts, process, and applications, Data preprocessing: cleaning, integration, transformation, reduction, Data mining tasks: classification, regression, clustering, association rule mining, anomaly detection, Evaluation of data mining models: accuracy, precision, recall, F1-score, ROC curve.

Unit 2: Classification and Regression:- Classification algorithms: decision trees, k-nearest neighbors (k-NN), support vector machines (SVM), Naive Bayes, etc. Regression analysis techniques: linear regression, logistic regression, Model evaluation and validation techniques: cross-validation, confusion matrix, ROC analysis

Unit 3: Clustering and Association Rule Mining:- Clustering algorithms: k-means, hierarchical clustering, DBSCAN, etc. Association rule mining: Apriori algorithm, FP-growth algorithm, Applications of clustering and association rule mining in real-world datasets.

Unit 4: Dimensionality Reduction and Feature Selection:- Techniques for dimensionality reduction: principal component analysis (PCA), singular value decomposition (SVD), t-distributed stochastic neighbor embedding (t-SNE), etc. Feature selection methods: filter, wrapper, embedded methods, Importance of dimensionality reduction and feature selection in data mining tasks.

Unit 5: Advanced Topics in Data Mining:- Ensemble learning methods: bagging, boosting, random forests, Deep learning for data mining: neural network architectures, convolutional neural networks (CNNs), recurrent neural networks (RNNs), etc. Big data analytics and scalability issues in

data mining, Ethical considerations and challenges in data mining and predictive analytics

Reference Book:

"Introduction to Data Mining" by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, "Pattern Recognition and Machine Learning" by Christopher M. Bishop, "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei, "Feature Engineering for Machine Learning" by Alice Zheng and Amanda Casari, "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

BBCS104D-II: Data Compression

Unit 1: Introduction to Data Compression:- Overview of data compression: definitions, goals, and applications, Lossless vs. lossy compression techniques, Information theory basics: entropy, entropy coding, Shannon's source coding theorem, Performance metrics for compression algorithms: compression ratio, throughput, quality loss.

Unit 2: Lossless Compression Techniques:- Huffman coding: principles and implementation, Arithmetic coding: principles and implementation, Dictionary-based compression: LZ77, LZ78, LZW, Burrows-Wheeler Transform (BWT) and Move-to-Front (MTF) coding.

Unit 3: Lossy Compression Techniques:- Transform-based compression: Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Quantization: scalar quantization, vector quantization, JPEG compression for images, MPEG compression for audio and video.

Unit 4: Compression Standards and Algorithms:- Overview of compression standards: JPEG, MPEG, MP3, ZIP, etc. Adaptive compression techniques, Error resilience and error concealment in compressed data, Multimedia compression challenges and solutions.

Unit 5: Advanced Topics in Data Compression;- Wavelet-based compression techniques, Context-based and prediction-based compression methods, Compression in specialized domains: genomic data, sensor data, etc. Emerging trends and future directions in data compression.

Reference Book:

"Introduction to Data Compression" by Khalid Sayood, "Data Compression: The Complete Reference" by David Salomon, "Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards" by Yun Q. Shi and Huifang Sun, "Multimedia Compression: Algorithms and Systems" by Vasudev Bhaskaran and Konstantinos Konstantinides, "Wavelets and Subband Coding" by Martin Vetterli and Jelena Kovacevic

**Skill Based: Skill
Enhancement Courses**

BDSC101SB: Internet of Things

Unit-I

(10Lectures)

Introduction – Definition and Characteristics of IoT, Physical Design of IoT; Things in IOT, Logical Design of IoT; IoT Functional Blocks, IoT Communication APIs, IoT Enabling Technologies; WSN, Cloud Computing, Big Data Analysis, Communication Protocols, Embedded Systems.

Unit-II

(10Lectures)

(14Lectures) IoT Hardware, Devices and Platforms – Basics of Arduino Hardware, The Arduino IDE, Basic Arduino Programming, Basics of Raspberry pi; Introduction to Raspberry pi, Programming with Raspberry pi, CDA IoT devices: Ubimote, WiFimote, BLE Mote, WINGZ gateway, Introduction to IoT Platforms, IoT Sensors and actuators.

Unit-III

(14Lectures)

IoT Protocols – IoT Data link Protocols, Network Layer Routing Protocols, Network Layer Encapsulation Protocols, Session Layer Protocols, IoT Security Protocols, Service Discovery Protocols, Infrastructure Protocols.

Unit-IV

(12Lectures)

IoT Programming – Arduino Programming: Serial Communications – Getting Input from Sensors, Visual, Physical and Audio Outputs, Remotely Controlling External Devices, Wireless Communication, Programming with Raspberry pi: Basics of python Programming, Python Packages of IoT, IoT Programming with CADC IoT devices.

Unit-V

(10 Lectures) Domain Specific

IoT – Home automation, smart cities, Smart Environment, IoT in Energy,

Logistics, Agriculture, industry and Health & Lifestyle sensors, Case Studies: A Case Study of Internet of Things U

sing Wireless Sensor Networks and Smart Phones, Security Analysis of Internet-of-Things: A Case Study

of August Smart Lock, Open IoT Platform

Reference Books:

- Vijay Madiseti and Arshdeep Bahga, —Internet of Things (A Hands-on Approach) II, 1st Edition, VPT, 2014.
- Margolis, Michael. —Arduino Cookbook: Recipes to begin, Expand and Enhance Your Projects I. O'Reilly Media Inc. 2011.
- Monk, Simon. Raspberry Pi Cookbook: Software and hardware problems and Solutions. O'Reilly Media, Inc. 2016.
- https://onlinecourses.swayam2.ac.in/aic20_sp06/preview
- https://onlinecourses.swayam2.ac.in/arp19_ap79/preview

BDSC102SB: Ethical Hacking

Unit-I

(14Lectures)

Introduction to Hacking – Importance of Security – Elements of Security – Phases of an Attack – Types of Hacker Attacks – Hactivism – Vulnerability Research – Introduction to Footprinting – Information Gathering Methodology – Footprinting Tools – WHOIS Tools – DNS Information Tools – Locating the Network Range – Meta Search Engines.

Unit-II

(10Lectures)

Introduction to Scanning – Objectives – Scanning Methodology – Tools Introduction to Enumeration – Enumeration Techniques – Enumeration Procedure – Tools.

Unit-III

(14Lectures)

Introduction – Cracking Passwords – Password Cracking Websites – Password Guessing – Password Cracking Tools – Password Cracking Countermeasures – Escalating Privileges – Executing Applications – Keyloggers and Spyware.

Unit-IV

(12Lectures)

Programming Fundamentals – C language – HTML – Perl – Windows OS Vulnerabilities – Tools for Identifying Vulnerabilities – Countermeasures – Linux OS Vulnerabilities – Tools for Identifying Vulnerabilities – Countermeasures.

Unit-V

(10Lectures)

Introduction – Security Assessments – Types of Penetration Testing – Phases of Penetration Testing – Tools – Choosing Different Types of Pen-Test Tools – Penetration Testing Tools.

Reference Books:

- ECCouncil, — Ethical Hacking and Countermeasures: Attack Phases, Cengage Learning, 2010.
- Jon Erickson, — Hacking, 2nd Edition: The Art of Exploitation, No Starch Press Inc., 2008.
- Michael T. Simpson, Kent Backman, James E. Corley, — Hands-On Ethical Hacking and Network Defense, Cengage Learning, 2013.

- Patrick Engebretson, — The Basics of Hacking and Penetration Testing—Ethical Hacking and Penetration Testing Made Easy!, Second Edition, Elsevier, 2013.
- Rafay Boloch, — Ethical Hacking and Penetration Testing Guide!, CRC Press, 2014
- https://onlinecourses.swayam2.ac.in/aic20_sp06/preview
https://onlinecourses.swayam2.ac.in/arp19_ap79/preview

MAKRU

BDSC103SB: Embedded system

Unit 1: Introduction to Embedded Systems and Data Analytics:- Overview of embedded systems: definition, characteristics, and applications, Introduction to data analytics in embedded systems, Importance of data-driven decision-making in embedded system design, Challenges and constraints of data analytics in embedded systems.

Unit 2: Data Acquisition and Preprocessing in Embedded Systems:- Sensors and actuators in embedded systems, Analog and digital signal processing techniques, Data acquisition methods: sampling, quantization, noise reduction, Preprocessing techniques: filtering, feature extraction, normalization.

Unit 3: Statistical Analysis for Embedded Systems:- Descriptive statistics: mean, median, mode, variance, standard deviation, Probability distributions: discrete and continuous distributions, Hypothesis testing and confidence intervals, Regression analysis for modeling relationships between variables.

Unit 4: Machine Learning for Embedded Systems:- Introduction to machine learning concepts, Supervised learning techniques: classification, regression, Unsupervised learning techniques: clustering, dimensionality reduction, Optimization for embedded systems: model size, inference time, power consumption.

Unit 5: Applications of Data Analytics in Embedded Systems:- Case studies and examples of data analytics applications in embedded systems, Real-time analytics for monitoring and control, Predictive maintenance and fault detection,, Embedded vision and image processing applications,

Reference Book:

"Embedded Systems: Architecture, Programming and Design" by Raj Kamal, "Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers" by Jonathan W. Valvano, "Statistics for Engineers and Scientists" by William Navidi, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers" by Pete Warden and Daniel Situnayake, "Practical Machine Learning for Computer Vision" by Valliappa Lakshmanan

BDSC104SB: Social Network analytics

UNIT I SOCIAL NETWORK ANALYSIS

(11Lectures)

Network analysis- Development of Social network analysis- Key concepts and measures in network analysis
-The global structure of networks - The macro-structure of social networks - Personal networks..

UNIT II WEB SEMANTICS IN SOCIAL NETWORK APPLICATIONS

(11Lectures)

Electronic sources for network analysis - Electronic discussion networks - Blogs and online communities
- Web-based networks - Knowledge Representation on the Semantic Web - Ontologies and their role in the

Semantic Web Ontology languages for the Semantic Web - The Resource Description Framework.

(RDF)and RDF Schema - The Web Ontology Language (OWL) - Comparison to the Unified Modelling

Language (UML) - Comparison to the Entity/Relationship (E/R) model and the relational model -

Comparison to the Extensible Markup Language (XML) and XML Schema.

UNIT III MODELLING AND AGGREGATING SOCIAL NETWORK DATA

(11Lectures)

State-of-the-art in network data representation - Ontological representation of social individuals -

Ontological representation of social relationships - Aggregating and reasoning with social network data -

Representing identity - On the notion of equality - Determining equality - Reasoning with instance equality -

Evaluating smushing

UNIT IV DEVELOPING SOCIAL-SEMANTIC APPLICATIONS

(11Lectures)

Building Semantic Web applications with social network features - The generic architecture of Semantic

Web applications -Sesame – Elmo – GraphUtil - The features of Flink - System design – open academia:

distributed, semantic-based publication management - The features of open academia - System design.

UNIT V EVALUATION OF SOCIAL NETWORK ANALYSIS

(11Lectures)

Evaluation of web-based social network extraction - Data collection - Preparing the data – Optimizing goodness of fit - Comparison across methods and networks - Predicting the goodness of fit – Evaluation through analysis - Semantic-based Social Network Analysis in the sciences - Data acquisition - Representation, storage and reasoning- Visualization and Analysis – Results - Descriptive analysis - Structural and cognitive effects on scientific performance .

TEXT BOOKS

1. Peter Mika , Social Networks and the Semantics Web”, Springer, 2007

REFERENCE BOOKS

1. Borko Furht, “Handbook of Social Network Technologies and Applications”, 1st Edition, Springer,2010.

E-BOOK

1. [http://www.asecib.ase.ro/mps/Social%20Networks%20and%20the%20Semantic%20Web%20\[2007\].pdf](http://www.asecib.ase.ro/mps/Social%20Networks%20and%20the%20Semantic%20Web%20[2007].pdf)

MOOC

1. <https://www.coursera.org/learn/social-network-analysis>.

BDSC105D-II: Fundamental of PHP

Unit 1: Introduction to PHP and Web Development Basics:- Overview of PHP: history, features, and applications, Setting up a PHP development environment, Basic syntax and structure of PHP scripts, Introduction to HTML and CSS for web development, Creating dynamic web pages with PHP.

Unit 2: Working with Variables, Data Types, and Control Structures in PHP:- Declaring and using variables in PHP, Data types and type conversion in PHP, Conditional statements: if, else, elseif, Looping constructs: for, while, foreach, Using arrays and associative arrays in PHP.

Unit 3: Functions, Forms, and File Handling in PHP:- Creating and calling user-defined functions, Handling form data using PHP: POST and GET methods, File input/output operations in PHP: reading from and writing to files, Validating and sanitizing user input in web forms, Uploading files and processing file uploads in PHP.

Unit 4: Database Integration with MySQL and PHP:- Introduction to MySQL database management system, Connecting to MySQL database from PHP, Performing database operations: querying, inserting, updating, deleting data, Prepared statements and SQL injection prevention, Handling MySQL errors and exceptions in PHP.

Unit 5: Advanced Topics in PHP for Data Analytics:- Integrating PHP with statistical libraries and packages (e.g., R, Python), Retrieving and processing data from external sources: APIs, web scraping, Implementing statistical algorithms and data analysis techniques in PHP, Generating dynamic visualizations

and reports using PHP and JavaScript libraries, Building interactive web applications for data analytics with PHP.

Reference Book:

"PHP and MySQL for Dynamic Web Sites" by Larry Ullman, "PHP: The Complete Reference" by Steven Holzner, "PHP Solutions: Dynamic Web Design Made Easy" by David Powers, "Learning PHP, MySQL & JavaScript" by Robin Nixon, "PHP Advanced and Object-Oriented Programming" by Larry Ullman.