Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade
P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India
Phone: +91 821 4276201 /202 /225, Fax: +91 824 2510677
Web: http://www.vvce.ac.in @vvceofficial

Scheme of Teaching and Examination for BE (Autonomous Scheme: 2022)

III SEMESTER – Information Science & Engineering

Sl. No.	Course Area	Course Code	Course Name	Teaching Department	Contact Hours / week			Examination				Credits
					L	T	P	Duration (Hrs.)	CIE Marks	SEE Marks	Total	
1	BS	BISMA301*	Transformers Partial Differential Equation & Combinatorics	Mathematics	2	2	0	3	50	50	100	3
2	IPCC	BISCO302#	Digital Design & Computer Organization	ISE	3	0	2	3	50	50	100	4
3	IPCC	BISDS303*	Data Structures & its Applications	ISE	3	0	2	3	50	50	100	4
4	PCC	BISOS304#	Operating System	ISE	3	0	0	3	50	50	100	3
5	HS	BGPEK305	General Proficiency Enhancement Course – I	TAP	0	2	0	2	50	50	100	1
6	HS	BUHVK306	Universal Human Values (CS, IS, AI ML, EEE)	ISE	2	0	0	2	50	50	100	2
7	PLC	BISPL317X	Professional Elective- I	ISE	0	0	4	2	50	50	100	2
8	AEC	BISTP308	Technical Proficiency Enhancement Course- I	TAP	0	0	2	2	50	50	100	1
TOTAL					**13**	**4**	**10**	**-**	**400**	**400**	**800**	**20**
9	Scheduled activities for III to VIII semesters	BNSSK309 / BPEDK309 / BYOGK309	NSS/ PE / Yoga *	NSS/ PE / Yoga	All students have to register for any one of the courses namely National Service Scheme, Physical Education (PE)(Sports and Athletics), and Yoga with the concerned coordinator of the course during the first week of III semester. The activities shall be carried out between III semester to VIII semester (for 5 semesters). Successful completion of the registered course is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities							

***Non–credit mandatory courses (NCMC):**

(A) National Service Scheme/Physical Education (Sport and Athletics)/ Yoga:
(1) Securing 40 % or more in CIE, 35 % or more marks in SEE and 40 % or more in the sum of CIE + SEE leads to successful completion of the registered course.
(2) In case, students fail to secure 35 % marks in SEE, they have to appear for SEE during the subsequent examinations conducted by the University.
(3) In case, any student fails to register for NSS, PE or Yoga/fails to secure the minimum 40 % of the prescribed CIE marks, he/she shall be deemed to have not completed the requirements of the course. In such a case, the student has to fulfill the course requirements during subsequent semester/s to earn the qualifying CIE marks.
(4) Successful completion of the course shall be indicated as satisfactory in the grade card. Non-completion of the course shall be indicated as Unsatisfactory.
(5) These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the courses shall be mandatory for the award of degree.

Professional Elective – I		**TPEC-I**	
BISPL3171*	**Advanced Web Programming Laboratory**	BISTP308*	**Problem solving using C and C++**
BISPL3172#	**Advanced Python Programming Laboratory**		
BISPL3173#	**Advanced Java Programming Laboratory**		

****All courses are common to (CSE/ISE/AI&ML) programs.*** ***#Common to CS/IS***

Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade
P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India
Phone: +91 821 4276201 /202 /225, Fax: +91 824 2510677
Web: http://www.vvce.ac.in @vvceofficial

Scheme of Teaching and Examination for BE (Autonomous Scheme: 2022)

IV SEMESTER - Information Science & Engineering

Sl. No.	Course Area	Course Code	Course Name	Teaching Department	Contact Hours / week			Examination				Credits
					L	T	P	Duration (Hrs.)	CIE Marks	SEE Marks	Total	
1	BS/PCC	BISMA401*	Advance Linear Algebra Probability Statistics & Optimization	ISE	2	2	0	3	50	50	100	3
2	IPCC	BISDA402*	Design & Analysis of Algorithms	ISE	3	0	2	3	50	50	100	4
3	IPCC	BISDB403*	Database Management System	ISE	3	0	2	3	50	50	100	4
4	PCC	BISSE404#	Software Engineering	ISE	3	0	0	3	50	50	100	3
5	HS	BGPEK405	General Proficiency Enhancement Course– II	TAP	0	2	0	2	50	50	100	1
6	AEC	BISBI406	Biology for IT Engineers (CS, IS, AI ML, EEE)	ISE	2	0	0	2	50	50	100	2
7	ESC/ETC /PLC	BISPL417X	*Professional Elective- II	ISE	0	0	4	2	50	50	100	2
8	AEC	BISTP408	Technical Proficiency Enhancement Course-II	ISE	0	0	2	2	50	50	100	1
TOTAL					**13**	**4**	**10**	**-**	**400**	**400**	**800**	**20**
9	Scheduled activities for III to VIII semesters	BNSSK409 / BPEDK409 / BYOGK409	NSS/ PE / Yoga *	NSS/ PE / Yoga	All students have to register for any one of the courses namely National Service Scheme, Physical Education (PE)(Sports and Athletics), and Yoga with the concerned coordinator of the course during the first week of III semester. The activities shall be carried out between III semester to VIII semester (for 5 semesters). Successful completion of the registered course is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities							

Note: *For CS/IS/CV a course related to mathematics shall be offered as an professional elective course. Students who are aspiring for higher studies shall take mathematics as an elective course during 4th semester if it is not a mandatory course.

Professional Elective – I			**TPEC-II**	
BISPL4171*	**Advanced Web Programming Laboratory**		BISTP408*	**Data Structures and Algorithms –I**
BISPL4172#	**Advanced Python Programming Laboratory**			
BISPL4173#	**Advanced Java Programming Laboratory**			

All courses are common to (CSE/ISE/AI&ML) programs. ***#Common to CS/IS***

Course Content

SEMESTER – III		
Course Name	**: Transformers, Partial Differential Equations and Combinatorics**	**Course Code: BISMA301**
Number of Lecture Hours / Week	**:02**	**CIE Marks: 50**
Number of Tutorial / Practical Hours / Week	**:02**	**SEE Marks: 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 40+10=50**	**SEE Duration: 03 Hours**
L:T:P	**: 2:2:0**	**CREDITS: 03**

COURSE PREREQUISITE

The students should have strong basis of integral calculus, trigonometry, partial derivative and permutation and combination.

COURSE OVERVIEW:

Transforms, Partial Differential Equations and Combinatorics is a course which provides mathematical techniques in the advanced areas of mathematics like Fourier, Z-transformation, numerical methods, partial differential equations, advanced linear algebra and combinatorics that are of at most relevance to the Engineering disciplines. The purpose of this course is to provide the skills and knowledge required to perform mathematical procedures and processes for solution of Engineering problems.

COURSE LEARNING OBJECTIVES (CLO)

The objective is to enable the students to apply the knowledge of Mathematics in various fields of Engineering by the following means:

- Explain the concept of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equations and combinatorics applying it appropriately in solving Engineering problems.
- Explain how to analyze the Engineering problems by making use of the concepts of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equations, and combinatorics.
- Explain the usage of modern tools to understand the concepts of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equations, and combinatorics.

MODULES	TEACHING HOURS
MODULE 1: **Fourier Series and Harmonic Analysis** **Fourier Series**: Periodic functions, Dirichlet's condition. Fourier series of periodic functions. Half range Fourier series, deducing some important series. **Practical Harmonic Analysis:** Harmonic Analysis. **SLT: Half range harmonic series.**	**08**
MODULE 2: **Fourier Transforms and Z –Transforms (I-C)** **Fourier Transforms:** Fourier transforms, Fourier sine and cosine transforms, Inverse Fourier transforms (direct method).	**08**

Z-Transforms: Basic definition, problems on Z-Transforms of standard functions (without proof), Damping and shifting rules (Problems only). Inverse Z-Transform and applications to solve difference equations. **SLT: Derivation Z-Transforms of standard functions.**	
MODULE 3: **Partial Differential Equations & Numerical solution of PDE(I-C)** Formation of PDE's by elimination of arbitrary functions. Solution of non-homogeneous PDE by direct integration, Solution of Linear PDE by Lagrange's multiplier method. **Numerical Methods to Solve PDE's:** Hyperbolic and Laplace equations. **SLT: Numerical methods to solve Parabolic equation.**	**08**
MODULE 4: **Sets, Relations, and Functions** **Set theory**: Operations and laws of set theory, inclusive and exclusive principles. **Relation and Function**: Relations and properties, posets, functions and properties, special functions, pigeonhole principles. **SLT: Venn diagram analysis.**	**08**
MODULE 5: **Combinatorics** Permutation & combination (with and without repetition), Multinomial expansion, rook's polynomial, generating functions, recurrence relations using generating function. **SLT: Forbidden analysis using rook's polynomial.**	**08**

Textbooks

1. B. S. Grewal, Higher Engineering Mathematics, Latest edition, Khanna Publishers.
2. B. V. Ramana, Higher Engineering Mathematics, Latest edition, Tata Mc. Graw Hill Publications
3. E. Balaguruswamy, Numerical Methods, Tata-McGraw-Hill Publication Limited
4. Dr D S Chandrashekaraiah, Combinatorics and Graph Theory, Prism Books Pvt.

Reference Books

1. Erwin Kreyszig, *Advanced Engineering Mathematics*, Latest edition, Wiley Publications.
2. Peter V. O'Neil, Advanced *Engineering Mathematics*, 7th edition, CENGAGE Learning India Pvt. Ltd. Publishers.
3. V. N. Ghougule, M. T. Puranic, *Numerical Methods and Optimization*, Niralli Publication.
4. M. K. Jain, S. R. K. Iyengar, *Numerical Methods,* sixth ed., New Age, International, Pub.
5. Ralph P. Grimaldi, Discrete and combinatorial Mathematics, Pearsons Educations.

Course outcomes (cos)

At the end of the course students will be able to:

CO1	**Understand** the basic concepts of Fourier, Z-Transformation, Numerical Methods,
CO2	**Apply** the concept of Fourier, Z-Transformation, Numerical Methods, Partial
CO3	**Analyze** mathematical problems arising in Engineering, using the concepts of
CO4	Use **modern tool** to **solve/visualize** mathematical problems arising in Engineering.

CO – PO – PSO Matrix

CO	PO												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1															
CO2	2															
CO3		2														
CO4					2											
Avg.	1.5	2			2											

Course Content

SEMESTER – III		
Course Name	**: Digital Design and Computer Organization**	**Course Code: BISCO302**
Number of Lecture Hours / Week	**:03**	**CIE Marks: 50**
Number of Tutorial / Practical Hours / Week	**:02**	**SEE Marks: 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 40+24=64**	**SEE Duration: 03 Hours**
L:T:P	**: 3:0:2**	**CREDITS: 04**

COURSE PREREQUISITE

Basic knowledge on Number systems, any programming language and components of computer systems.

COURSE OVERVIEW

This course is paced to provide a Basic concepts of digital systems, design of sequential and combinational circuits also provides understanding of structure of computer system, various I/O ,memory systems and arithmetic operation.

COURSE LEARNING OBJECTIVES (CLO)

- To understand basic concepts of digital systems
- To analyze and design sequential and combinational circuits
- To understand Basic structure of computer system and their operation.
- To explain the concept of various I/O and memory systems.

MODULES	TEACHING HOURS
MODULE 1 **Introduction to Digital Design:** Binary Logic, Basic Theorems And Properties of Boolean Algebra, Boolean Functions, Digital Logic Gates, Introduction, The Map Method, Four-Variable Map, NAND and NOR Implementation **SLE:** Don't Care conditions **Text book 1: 1.9, 2.4, 2.5, 2.8, 3.1, 3.2, 3.3, 3.5, 3.6, 3.9**	**08**
MODULE 2 **Combinational Logic**: Introduction, Combinational Circuits, Design Procedure, Binary Adder- Subtractor, Decoders, Encoders, Multiplexers. **Sequential Logic**: Introduction, Sequential Circuits, Storage Elements: Latches, Flip-Flops. **SLE**: JK/Master-slave FLIP-FLOPs **Textbook 1 :4.1, 4.2, 4.4, 4.5, 4.9, 4.10, 4.11, 4.12, 5.1, 5.2, 5.3, 5.4**	**08**
MODULE 3 **Basic Structure of Computers**: Basic Operational Concepts, Bus Structures, Performance-Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. **Machine Instructions and Programs**: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes. **SLE**: Additional instructions **Textbook2 :** 1.3-1.4,1.6(1.6.1-1.6.4,1.6.7),2.2-2.3,2.4(2.4.1-2.4.5),2.5,2.10	**08**

MODULE 4 **Input/output Organization**: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Direct Memory Access: Bus Arbitration, Speed, size and Cost of memory systems. Cache Memories – Mapping Functions. **SLE**: Read only Memories **Textbook2 :** 4.1, 4.2.1, 4.2.2, 4.2.3, 4.4, 5.4, 5.5.1	**08**
MODULE 5 **Arithmetic :** Addition & subtraction of signed numbers, Design of fast adders; Multiplication of positive numbers, Signed-operand multiplication, Integer division. **Basic Processing unit:** Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization. **SLE**: Fast Multiplication. **Textbook2:** 6.1-6.6,7.1-7.3	**08**
PRACTICAL MODULE **Simulation packages preferred: Multisim, Modelsim, Logisim or any other relevant** **Exercise** 1. Realization of Basic gates using Logisim. 2. Realization of Universal gates using Logisim. 3. Design and implement different types of multiplexer using Logisim. **Structured Enquiry** 4. Design and implementation of various types of Flip-Flops using Logisim. 5. Realize various types of shift registers using Logisim 6. Design of synchronous counter using Logisim **Demonstration** 7. Realization of ripple carry adder using Logisim. 8. Design of 8 bit ALU using Logisim. **Open Ended Programs** 1. Design 4 bit adder and subtractor using any simulator. 2. Realization of Binary to Grey converter using any simulator. **3.** Design of demultiplexer using any simulator. 4. Design and implement JK flip flop using any simulator. **5.** Design of Asynchronous counters using any simulator. **Experiment Weightage**	**24**

Type of Experiment	Program-No	Weightage
Demonstration	7,8	20%
Exercise	1,2,3	35%
Structured Enquiry	4,5,6	35%
Open ended	-	10%

Textbooks

1. M. Morris Mano & Michael D. Ciletti, Digital Design With an Introduction to Verilog Design, 5e, Pearson Education-2018
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, Fifth Edition, McGraw Hill Education (India), 2012.

Reference Books

1. Morris Mano, Michael D. Ciletti, "Digital Design : With an Introduction to the Verilog HDL, VHDL, and System Verilog", Sixth Edition, Pearson Education, 2018.
William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015

Course Outcomes (COs)

At the end of the course students will be able to:

CO1	**Understand** the simplifications of logic expression and basic structure of a computer.
CO2	**Illustrate** the applications of K map, combinational, sequential circuit and machine instructions.
CO3	**Analyze** the appropriate technique for arithmetic operation,memory and I/O systems.
CO4	**Design and Simulate** for given logic circuits and computer architecture using any simulator.

CO – PO – PSO Matrix

CO	PO												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2												2			
CO2	3												3			
CO3		2											2			
CO4			2		2								2			
Avg.	2.5	2	2		2								2.25			

Course Content

SEMESTER – III		
Course Name	**: Data Structures and its Applications**	**Course Code: BISDS303**
Number of Lecture Hours / Week	**:03**	**CIE Marks : 50**
Number of Practical Hours / Week	**:02**	**SEE Marks : 50**
Total Number of Lecture + Practical Hours	**: 40+24=64**	**SEE Duration : 03 Hours**
L:T:P	**: 3:0:2**	**CREDITS : 04**

Course Prerequisites

Basic knowledge of C programming language, mathematics, organizing and problem-solving ability.

Course Overview

This Course focuses on basic algorithms, data structures and its implementation. Some of the data structures we will encounter include linked lists, stacks, queues, trees, heaps, hash tables and graphs. Course will also focus on how to analyse algorithms for searching, traversing trees, hashing, manipulating priority queues and finding shortest paths in graphs.

Course Learning Outcomes (CLO)

This course will enable students to,

- Understand the foundations of data structure and how different data structures are used for effective data access and data manipulation.
- Investigate various data structures such as stacks, queues, link lists, trees and graphs.
- Understand the context of problem definition and implement a suitable data structure to solve it.

MODULES	TEACHING HOURS
MODULE 1 **Introduction**: Data structures, pointers and dynamic memory allocation, data abstraction. **Arrays:** Dynamically allocated arrays, Sparse matrix and polynomial representation. **String Processing & Pattern matching Algorithms:** Naive Pattern Searching, KMP Algorithm. **Sorting Techniques:** Insertion Sort, Radix Sort **SLT:** String Operations **Textbook 1:** Ch.1-1.2,1.4, Ch.2-2.2,2.4,2.5,2.7.3 **Textbook 2:** Ch.1-1.4, Ch.3-3.7, Ch. 9-9.3,9.7	**08**
MODULE 2 **Stacks:** Introduction, Array representation, Applications of stacks: Infix to Postfix, Evaluation of Postfix, Recursion, Tower of Hanoi. **Queues:** Introduction, Circular queues, Deques, Priority queue. **SLT:** Multiple stacks and queues **Textbook 1:** Ch. 3- 3.4,3.6 **Textbook 2:** Ch. 6-6.1 to 6.3,6.5, 6.7, 6.8, 6.10, 6.12,6.13	**08**

MODULE 3 **Linked Lists:** Introduction, Representation of linked list in memory, traversing and searching linked list, insertion, deletion from the linked list, header linked list, two-way linked list**,** Linked list representation of stacks and queues, Circular linked list **SLT:** Addition and concatenation of two lists **Textbook 1:** Ch.4 - 4.4.4 **Textbook 2:** Ch. 5 - 5.1 to 5.4, 5.7 to 5.10, Ch.6 – 6.4, 6.11	**08**
MODULE 4 **Trees:** Introduction, Binary Trees, Binary tree Traversal, Additional binary tree operations, Threaded Binary trees, Binary search Trees **SLT**: Expression trees, Heaps **Textbook 1:** Ch. 5- 5.1 to 5.5, 5.7	**08**
MODULE 5 **Trees:** Selection Trees, Forest, Counting binary tree **Efficient Binary Search tree**: AVL Tree, Red Black tree **Graphs:** ADT, Elementary graph operations: BFS, DFS **Hashing**: Introduction, Static Hashing, Dynamic Hashing. **SLT:** Optimal Binary Search trees **Textbook 1:** Ch. 5 - 5.8 to 5.11, Ch.6 – 6.1 to 6, Ch.8- 8.1 to 8.3, Ch.10 - 10.2 to 10.4	**08**
PRACTICAL MODULE **Exercise Experiments:** **1.** Design, Develop and Implement a Program in C for the following operations on Strings: 1. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) 2. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR. 3. Pattern Matching Algorithm: Brute Force / KMP 4. Support the program with functions for each of the above operations. Don't use Built-in functions. 5. Check the following test cases. **Test Case 1:** STR = "VVCE MYSURU", PAT=" MYSURU", REP=" KARNATAKA", OUTPUT=" VVCE KARNATAKA" **Test Case 2:** STR = "COMPUTER SCIENCE", PAT=" COMPUTER", REP=" BASIC", OUTPUT=" BASIC SCIENCE" **2.** Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX). 1. Push an Element on to Stack. 2. Pop an Element from Stack. 3. Demonstrate how stack can be used to check palindrome. 4. Demonstrate Overflow and Underflow situations on Stack. 5. Display the status of Stack. 6. Exit	**24**

3. Design, Develop and Implement a Program in C for the following operations on expression.

a. Read infix expression String (INFIX)
b. Convert the infix expression (INFIX) to a postfix expression using stacks.
c. Evaluate the postfix expression using stacks.
d. Check the following test cases.

Test Case 1: Infix = "(1+ (2-3) *4)", Postfix="123-4*+", Result = -3

Test Case 2: Infix = "4/2-2+3*3-4*2", Postfix="42/233*42*-+-", Result = -1

Note: Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.

Demonstration Experiments:

4. Design, Develop and implement priority queue for job processing.

Note:

- The Program should allow users to add or remove items from the queue.
- It should also display current status i.e. the total number of items in the queue.

5. Design, Develop and implement c program using singly linked list for the following scenario

a. There are two linked list A and B containing the following data:
A: 3,7,10,15,16,09,22,17,32 and B: 16,02,09,13,37,08,10,01,28
b. Create a linked list C that contains only those elements that are common in linked list A and B
c. Create a linked list D which contains all elements of A as well as B ensures that there is no repetition of elements.

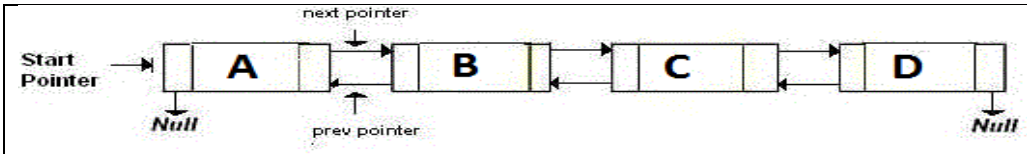
6. Design, Develop and Implement a menu driven Programming in C for the following operations on Binary Search Tree (BST) of Integers.

a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
b. Traverse the BST in In-order, preorder, post-Order, zigzag order
c. Search the BST for a given element (KEY) and report the appropriate message.
d. Display the height of binary trees.
e. Exit

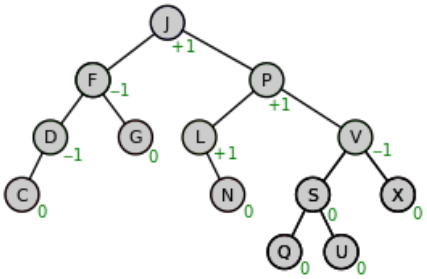
Structured Enquiry:

7. Design, Develop and implement C program for the following operations on doubly linked list.

a. Create doubly linked list of N nodes with integer data by adding each node at the front.
b. Delete the node of a given data if it is found, otherwise display appropriate message.
c. Insert a node to the left of the node whose key value is read as input.
d. Display the contents of the list.

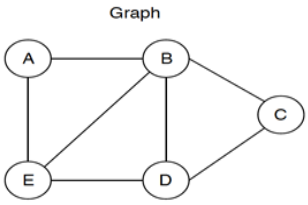


8. Design, Develop a program in C to implement **AVL tree** operations.



9. Design, Develop and Implement a Program in C for the following operations on Graph (G) of Cities

a. Create a Graph of N cities using Adjacency Matrix.
b. Print all the nodes reachable from a given starting node in a digraph using the DFS / BFS method.



10. Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F.

a. Assume that file F is maintained in memory by a Hash Table (HT) of M memory locations with L as the set of memory addresses (2-digit) of locations in HT.
b. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash Function H: K%L as I (remainder method), and implement hashing techniques to map a given key K to the address space L.
Resolve the collision (if any) using linear probing

Open Ended Enquiry:

- **Design and Develop C program to implement following (Not limited to)**
  a. Red Black Tree.
  b. Addition of Two polynomial using Linked List.
  c. Maze problem.
  d. Undo & Redo operation using stack.
  e. Dynamic Hashing.
- **Design and Develop C program to implement following search algorithms:**
  a. Jump Search.
  b. Interpolation Search.
  c. Exponential Search.

d. Sub list Search (Search a linked list in another list) Fibonacci Search.
e. The Ubiquitous Binary Search.

Weightages:

Type of Experiment	Program Numbers	Weightage
Demonstration	1 & 2	18%
Exercise	3,4,5 & 10	36%
Structured Enquiry	6,7,8,9	36%
Open Ended		10%

Textbooks

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, reprint 2018.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

Reference Books

1. Programming and Data Structure by Jackulin C Salini etal., Ane books publishers, 2019
2. Learning JavaScript data structures and algorithms hone your skills by learning classic data structures by Loiane Groner, Pack T publishing, 2019

Course Outcomes (COs)

At the end of the course students will be able to:

CO1	**Explain** the fundamentals of basic data structures.
CO2	**Apply** the various data structures to solve the given problem.
CO3	**Analyze** the applications of various data structures.
CO4	**Design** appropriate solution by implementing suitable data structure for a given problem as individual/team**.**

CO – PO – PSO Matrix

CO	PO												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2												2			
CO2	3												3			
CO3		2					2						2			
CO4			3						2				3			
Avg	**2.5**	**2**	**3**				2		**2**				**2.5**			

Course Content

SEMESTER – III		
Course Name	**: Operating System**	**Course Code: BISOS304**
Number of Lecture Hours / Week	**:03**	**CIE Marks : 50**
Number of Tutorial / Practical Hours / Week	**:00**	**SEE Marks : 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 40**	**SEE Duration : 03 Hours**
L:T:P	**: 3:0:0**	**CREDITS : 03**

Course Prerequisites

Basic knowledge of components of the computer and its working and fundamental knowledge of C programming

Course Overview

An operating system is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all its software and hardware. This course introduces the basic concepts of an operating system. Students can apply and analyse different memory allocation schemes, CPU scheduling and process synchronization algorithms.

Course Learning Outcomes (CLO)

- To make aware of basics of Operating System and their services
- To learn different process scheduling algorithms and synchronization techniques
- To understand memory management concepts
- To explore features of various operating systems

MODULES	TEACHING HOURS
MODULE 1 **Introduction to operating systems**: What Operating Systems do, Operating System structure, Operating System operations, Operating System services, System Calls, System Programs, Design and Implementation, Operating System structure– Simple, Layered, Microkernels and Modules. **SLE:** Types of system calls **TextBook:** Ch. 1.1, 1.4, 1.5, 2.1 – 2.6, 2.7.1 - 2.7.4	**08**
MODULE 2 **Processes and CPU Scheduling:** Process concept, Process scheduling, Operations on processes, Inter process communication, Overview of Threads, Multithreaded models, Basic concepts of scheduling, Scheduling criteria and algorithms. **SLE:** Threading issues **TextBook:** Ch. 3.1 - 3.4, 4.1, 4.3, 4.6, 5.1 - 5.3	**08**
MODULE 3 **Process Synchronization:** Synchronization Background, Critical section problem, Peterson's solution, Semaphores, Classic problems of synchronization.	**08**

Memory Management: Background, Swapping, Contiguous Memory allocation, Segmentation. **SLE:** Monitors **Textbook**: Ch. 6.1 - 6.3, 6.6 - 6. 8, 8.1 - 8.4	
MODULE 4 **Memory Management** : Paging, Structure of page table **Virtual memory:** Background, Demand paging, Copy on write, Basic Page replacement, Page replacement algorithms: FIFO, Optimal and LRU. **SLE:** Thrashing **Textbook:** Ch. 8.5- 8.6, 9.1 - 9.3, 9.4.1 - 9.4.4, 9.6	**08**
MODULE 5 **Secondary Storage Structures:** Overview of Mass-storage Structures, Disk Scheduling. **Deadlocks:** System Model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance: Banker's algorithm, Deadlock detection, Recovery from deadlock. **SLE:** Disk structure, Disk Attachment **Textbook:** Ch. 7.1 - 7.7, 12.1 – 12.4	**08**

Textbooks

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, 9th edition, Wiley, 2018.

Reference Books

1. Modern Operating Systems, Andrew S Tanenbaum and Herbert Bos, Fourth Edition, Pearson Education,2014
2. Thomas Anderson and Michael Dahlin, Operating Systems: Principles and Practice, Recursive Books, Second Edition, 2014
3. Stallings, Operating Systems internals and design Principles, 7th Edition, 2017

Course Outcomes (COs)

At the end of the course students will be able to:

CO1	**Explain** the basics and functions of operating system.
CO2	**Apply** the concepts of memory management policies, virtual memory, and scheduling techniques for processes handling.
CO3	**Analyze** processes and memory management in operating system.
CO4	**Design** and develop solutions for process management in a team. **(Additional CO- PO4 & PO9)**

CO – PO – PSO Matrix

CO	PO												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2												2			
CO2	3												3			
CO3		2											2			
CO4				2					2				2			
Avg.	**2.5**	**2**		**2**					**2**				**2.25**			

Course Content

SEMESTER – III		
Course Name	**: General Proficiency Enhancement Course-I**	**Course Code: BGPEK305**
Number of Lecture Hours / Week	**:00**	**CIE Marks : 50**
Number of Tutorial / Practical Hours / Week	**:02**	**SEE Marks : 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 30**	**SEE Duration : 02 Hours**
L:T:P	**: 0:2:0**	**CREDITS : 01**

Course Prerequisites
Knowledge of basic mathematics.

Course Overview
Aptitude training is designed to enhance an individual's problem-solving, analytical, and critical thinking skills. It is often a crucial component of career development, competitive exams, and job recruitment processes.

Course Learning Outcomes (CLO)

- Develop Problem Solving Skills.
- Enhance Quantitative Aptitude, Logical Reasoning and Verbal Ability Skills.
- Prepare students for job recruitment process and competitive exams.

MODULES	TEACHING HOURS
MODULE 1 **Quantitative Aptitude**: Number System – Divisibility & Remainder, Multiples & Factors, Integers, HCF & LCM, Decimal Fractions, Surds & Indices, Simplification.	**06**
MODULE 2 **Quantitative Aptitude***:* Percentages, Profit and Loss. **Logical Reasoning:** Blood Relations, Direction Sense Test.	**06**
MODULE 3 **Verbal Ability:** Change of Speech & Voice, Sentence Correction, Sentence Completion, Closet Test.	**06**
MODULE 4 **Quantitative Aptitude***:* Simple and Compound Interest, Averages. **Logical Reasoning:** Number & Letter Series, Coding and Decoding, Analogy.	**06**
MODULE 5 **Quantitative Aptitude***:* Alligations and Mixtures, Ratios, Proportions and Variations, Partnership. **Logical Reasoning**: Seating Arrangements.	**06**

Textbooks:

1. Quantitative Aptitude for Competitive Examinations by R.S Aggarwal
2. A Modern Approach to Verbal & Non-Verbal Reasoning by R.S. Aggarwal

COURSE OUTCOMES (COs) At the end of the course, students will be able to	
CO1	solve complex problems using logical and structured approaches.
CO2	Analyze complex problems using logical and structured approaches
CO3	develop critical thinking skills, enabling them to evaluate information and make effective decisions.

CO – PO – PSO Matrix

	PO												PSO			
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2															
CO2		2														
CO3			2													
Avg.	**2**	**2**	**2**													

Course Content

SEMESTER – III		
Course Name	**: Universal Human Values**	**Course Code: BUHVK306**
Number of Lecture Hours / Week	**:02**	**CIE Marks : 50**
Number of Tutorial / Practical Hours / Week	**:00**	**SEE Marks : 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 25**	**SEE Duration : 02 Hours**
L:T:P	**: 2:0:0**	**CREDITS : 02**

Course Prerequisites

Good knowledge of C, Computer Organization and Architecture, fundamentals of computer.

Course Learning Outcomes (CLO)

1. Development of a holistic perspective based on self-exploration about themselves (human being), family, society and nature/existence.
2. Understanding (or developing clarity) of the harmony in the human being, family, society and nature/existence
3. Strengthening of self-reflection.
4. Development of commitment and courage to act.

MODULES	TEACHING HOURS
MODULE 1 **Course Introduction - Need, Basic Guidelines, Content and Process for Value Education** Purpose and motivation for the course, recapitulation from Universal Human Values-I, Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and Experiential Validation- as the process for self-exploration, Continuous Happiness and Prosperity- A look at basic Human Aspirations, Right understanding, Relationship and Physical Facility- the basic requirements for fulfillment of aspirations of every human being with their correct priority. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario, Method to fulfill the above human aspirations: understanding and living in harmony at various levels.	**08**
MODULE 2 **Understanding Harmony in the Human Being - Harmony in Myself** Understanding human being as a co-existence of the sentient 'I' and the material 'Body', Understanding the needs of Self ('I') and 'Body' - happiness and physical facility, Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer), Understanding the characteristics and activities of 'I' and harmony in 'I', Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail, Programs to ensure Sanyam and Health.	**08**

MODULE 3 **Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship** Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship. Understanding the meaning of Trust; Difference between intention and competence. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship. Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals. Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family.	**08**
MODULE 4 **Understanding Harmony in the Nature and Existence - Whole existence as Coexistence** Understanding the harmony in the Nature, Interconnectedness and mutual fulfilment among the four orders of nature recyclability and self-regulation in nature. Understanding Existence as Co-existence of mutually interacting units in all pervasive space. Holistic perception of harmony at all levels of existence. Discussion on human being as cause of imbalance in nature, pollution, depletion of resources and role of technology etc.	**08**
MODULE 5 **Implications of the Holistic Understanding of Harmony on Professional Ethics** Natural acceptance of human values, Definitiveness of Ethical Human Conduct. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order. Competence in professional ethics: a. Ability to utilize the professional competence for augmenting universal human order b. Ability to identify the scope and characteristics of people friendly and eco-friendly production systems, c. Ability to identify and develop appropriate technologies and management patterns for above production systems. Case studies of typical holistic technologies, management models and production systems. Strategy for transition from the present state to Universal Human Order: a. At the level of individual: as socially and ecologically responsible engineers, technologists, and managers. b. At the level of society: as mutually enriching institutions and organizations.	**08**

Textbooks

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010.

Reference Books

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).

4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj - PanditSunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)

COURSE OUTCOMES (COs) At the end of the course, students will be able to	
CO1	Understand the need of human values to become more aware of themselves, and their surroundings (family, society, nature).
CO2	Understand the responsibilities in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.
CO3	Have better critical ability and also become sensitive to their commitment towards
CO4	Apply what they have learnt to their own self in different day-to-day settings in real life.

CO – PO – PSO Matrix

CO	**PO**												**PSO**			
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1						2										
CO2						2		2								
CO3						2		2								
CO4												2				
Avg.						**2**		**2**				**2**				

Course Content

SEMESTER – III		
Course Name	**: Advanced Web Programming Laboratory**	**Course Code: BISPL3171**
Number of Lecture Hours / Week	**:00**	**CIE Marks : 50**
Number of Tutorial / Practical Hours / Week	**:04**	**SEE Marks : 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 40**	**SEE Duration : 03 Hours**
L:T:P	**: 0:0:4**	**CREDITS : 02**

Course Prerequisites

Basic knowledge of programming, HTML, CSS and JavaScript are required to learn the course.

Course Overview

The focus of this course is on the World Wide Web as a platform for interactive, applications, content publishing and social services. This course is about the HTTP communication protocol, the markup languages HTML, the CSS for formatting and transforming web content, XML, AJAX, Node.js and PHP.

Course Learning Outcomes (CLO)

This course will enable students to,

- Learn the syntax and semantics of web technologies.
- Demonstrate the use of built-in functions.
- To understand the fundamentals of Nodejs, XML, AJAX and PHP.
- Interpret the use of procedural statements like conditional statements, loops and function calls in JavaScript

PART A

1. Design a HTML program to display two paragraphs with following specifications:

   P1: display the heading called “ADVANCED WEB PROGRAMMING” and set the text color to white and background color to blue.

   P2: A paragraph with the basic elements such as bold, italic, subscript, super script, mark and with a horizontal line and line break
2. Create a webpage to display a table consisting of a customer detail such as customerID, Name, Experience and Salary.
3. Create a webpage to display an online shopping website with the list of items available with the concept of ordered and unordered list.
4. Design a webpage to create a form for a student to enroll for a course which includes the following fields: Name, USN, Gender Subject interested with the drop-down options, email, Contact Number, Address with submit and reset button.

5. Design HTML webpage with following CSS specifications: Show any two-font family through text, decorated the text through underline, overline and line through, display the head text with shadow in red color.
6. Develop javascript code to check whether the given number is prime or not.
7. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference, and quotient.
8. Develop javascript code to find the largest of three numbers.

NOTE: The above programs will be assessed in CCE ONLY.

PART B

1. Develop a javascript to sort and accessing the array elements.

   Input: n=5

   Unsorted array elements: 10,1,-5,4,15

   Sorted Array: -5,1, 4, 10, 15
2. Create a class by the name rectangle with 2 attributes length and breadth. Include a parameterized constructor to assign values to data members and a function to calculate area of the rectangle. Demonstrate creation of object of class rectangle and display its area.

   Input: length=5, breadth=6

   Output: Area=30
3. Develop a javascript to demonstrate the working of callback and async functions.
4. Develop an arrow function in javascript that checks whether a year is leap year, alert the user with true if the year is leap year and false if year is non leap year. Validate centuries also.

   Input: 2000, Output: Leap year

   Input: 2100, Output: Non Leap year

   Input: 2004, Output: Leap year

   Input: 2006, Output: Non Leap year
5. Develop a javascript that accepts length and breadth of rectangle as parameter of an arrow functions. Call the function using spread and rest operator and alert the user with perimeter of the rectangle.

6. Develop a javascript to demonstrate the usage of optional and default parameters in a function.
7. Create a class by the name box with parameters length, breadth, and height. Create a class boxweight that extends box and include a new parameter weight. Create another class by the name boxcost that extends boxweight and has a parameter by the name shipmentcost. Include constructors in all the classes. Create an object of boxcost and display values of all parameters that represent multilevel inheritance.
8. Cretae a database with studentname and usn. Develop a javascript that accesses this database using get methods of REST API to display the contents of database in webpage.

Open-Ended Experiments

Students shall solve a problem (either given by the staff or student may come up with their own problem) using the design techniques.
1. Weather forecast website
2. Javascript Music Player.
3. Hospital Management System.
4. Online Shopping Webpage.

Experimental Weightage:

Type of Experiment	Program -No	Weightage
Exercise	5, 7, 8	33.3%
Demonstration	1, 2, 4	33.3%
Structured enquiry	3, 6	22.2%
Open-Ended experiments		11.1%

Course Outcomes:
At the end of the course, the students will be able to

CO1	Apply and demonstrate the usage of various programming constructs.
CO2	Design and Develop applications to solve problems across various technical and real-world domains.
CO3	Explore various programming tools and techniques in a team.

CO-PO-PSO Matrix

CO	PO												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1			2											2		
CO2			2											2		
CO3					2									2		
AVG.			2		2									2		

Course Content

SEMESTER – III		
Course Name	**: Advanced Python Programming Laboratory**	**Course Code: BISPL3172**
Number of Lecture Hours / Week	**:00**	**CIE Marks : 50**
Number of Tutorial / Practical Hours / Week	**:04**	**SEE Marks : 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 40**	**SEE Duration : 02 Hours**
L:T:P	**: 0:0:4**	**CREDITS : 02**

Course Prerequisites

Basic knowledge of programming like python and C.

Course Overview

This course provides hands-on coding exercises on programming concepts like classes and objects, custom functions, data exploration and manipulation.

Course Learning Outcomes (CLO)

- Understand and implement the basic constructs of object oriented programming.
- Develop programs for data exploration and manipulation using various python packages.
- Perform pattern matching using regular expressions.

PART-A

1. Write a python program to check whether the number is prime or not.

2. Write a Python program to calculate the length of a string without using built in function and check whether it is a palindrome or not.

3. Write a Python function that accepts a sentence from the user and counts the number of words in the sentence.

4. Write a Python program to sum all the numbers in a list.

5. Write a Python program to store phone numbers of persons in a dictionary and display the phone number of the requested person.

6. Develop a python program to copy the contents of only odd lines from one file to another file.

7. Write a python program to check if the number is even or odd using a user defined function.

8. Develop a python program to count occurrence of all characters present in a string.

NOTE: The above programs will be assessed in CCE ONLY.

PART-B

A- Demonstration

A1. Create a base class "polygon" and the derived class "triangle", demonstrate inheritance by inheriting findArea() function to calculate area of the triangle.

A2. Design a Python program using Regular expressions to

a) Extract Email IDs from a given text.

b) Validate the user password with minimum length=6 and maximum length=16 and must have at least one lower-case letter, upper-case letter, number and special symbol (#, @, $, _).

A3. Write a NumPy program to create a structured array from given student name, height, class and their data types. Finally sort the array on height.

A4. Read the given data "churn.csv" and save it as a dataframe called churn_data. Perform following operations on the dataframe i) Count total number of duplicate records in the dataframe ii) Count the no. of duplicate records in the churn dataframe based on the cutomerID column iii) Count number of missing values in each columns iv) Count the total no. of missing values for the variable TotalCharges v) Average monthly charge paid by a customer for the services he/she has signed up for vi) Display the records having "1@#" under the variable Dependents vii) Replace null values in churn dataframe by median value or by max count class category *(https://drive.google.com/file/d/1JSYGIIkIZr4jyheDEH0X1_TMSnQ2CzXc/view)*

B- Exercise

B1. Explore the bicycle counts on Seattle's Fremont Bridge Data with respect to i) Average daily bicycle counts ii) Average hourly bicycle counts by weekday and weekend.
(https://data.seattle.gov/api/views/65db-xm6k/rows.csv?accessType=DOWNLOAD)

B2. Using the data on births in the United States, provided by the Centers for Disease Control (CDC), Find i) Total number of US births by year and gender ii) Average daily births by day of week and decade
(https://raw.githubusercontent.com/jakevdp/data-CDCbirths/ master/births.csv)

C- Structured Inquiry

C1. Develop a python program to store and display Employee details such as EID, Name, Place, and Department. The EID must be auto-generated for each employee.

C2. Explore the automobile dataset and visualize the i) Distribution of the two and four door cars with respect to the type of fuel they use ii) Distribution of cars of different body styles with respect to the type of fuel they use ii) Total number of each type of body style cars categorized by fuel type iii) Horsepower of each of the fuel type with reference to the type of drive wheel present in cars.

(https://www.kaggle.com/datasets/toramky/automobile-dataset?resource=download)

Open Ended Experiments

The student can choose to solve any one open-ended problem to illustrate python application in the domains specified below using various python packages.

1.Excel file handling

2.PDF/word file manipulation

3. Web Scraping.

COURSE OUTCOMES (COs)

At the end of the course, students will be able to

CO1	Apply and demonstrate the usage of various programming constructs.
CO2	Design and develop applications to solve problems across various technical and real world domains.
C03	Explore various programming tools and techniques in a team.

CO – PO – PSO Matrix

CO	PO											PSO				
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1			2											2		
CO2			2											2		
CO3					2									2		
Avg			2		2									2		

Course Content

SEMESTER – III		
Course Name	**: Advanced Java Programming Laboratory**	**Course Code: BISPL3173**
Number of Lecture Hours / Week	**:00**	**CIE Marks : 50**
Number of Tutorial / Practical Hours / Week	**:04**	**SEE Marks : 50**
Total Number of Lecture + Tutorial/Practical Hours	**: 40**	**SEE Duration : 02 Hours**
L:T:P	**: 0:0:4**	**CREDITS : 02**

Course Prerequisites

Basic understanding of Java programming with OOPS.

Course Overview

The focus of this course is on design and implementation of advanced java concepts through hands on experience to develop real world applications.

Course Learning Outcomes (CLO)

This course will enable students to,

- Familiarize advanced features of Java concepts.
- Develop core java concepts such as multiple threads, JDBC, JSP, Applets and Servlets.

PART A

Following are Basic core Java programs.

1. Write a Java program that prompts the user for an integer N and generates all the prime numbers up to N.
2. Write a Java program to create a class box with instance variable width, height, depth and create an object using default constructors and parameterized constructors.
3. Write a Java program for adding two numbers using method overloading.
4. Write a Java program to convert an integer 257 to byte using narrowing type conversion and widening type conversion.
5. Write a Java program to sort the string elements in a 1-dimensional array.
6. Write a Java program to sum all the elements of an array using for-each version of for loop.
7. Create a Java class Customer with the following details as variables within it: CustID, Name, Age, Phone, Place. Write a Java program to create n Customers objects and print the CustID, Name, Age, Phone and Place of these objects with suitable headings using "this" keyword.
8. Design a super class called Employee with details as EmpID, Name, Phone, Salary, extend this class by writing two subclasses namely Tester (ProjectID, ProjectName), Developer (ProjectName). Write a Java Program to read and display at least 2 Employee objects of all two categories using Inheritance.

NOTE: The above programs will be assessed in CCE ONLY.

PART B

1. Develop a Java program to create an interface named Shape that contains a

method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the interface Shape. Each one of the classes contains only the method printArea() that prints the area of the given shape.

2. Create a class Book which contains four members: name, author, price, num of pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.
3. Write a program that demonstrates handling of exceptions in inheritance. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age () i.e when the input age is equal to father's age.
4. Write a program which creates two threads, one thread displaying "Vidyavardhaka College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.
5. Write a program that sorts an array of strings using compareTo() to determine bubble sort ordering.
6. Develop a java program to create an enum as session and demonstrate the usage of value(), valueOf() and ordinal() methods.
7. Write a Java program to implement the SQL commands using JDBC.
8. Write a JSP program which shows a Sample Order Form.

Open Ended Experiments

Students shall solve a problem (either given by the staff or student may come up with their own problem no restrictions) using the design techniques.

1. Develop JSP program which displays the System date and time.
2. Develop a Swings program to displaying image on the button.
3. Develop session handling using servlets.
4. Develop Chat Server using Java.

Experimental Weightage:

Type of Experiment	Program -No	Weightage
Exercise	1,2	10%
Structure Enquiry	3,7,8	37%
Demonstration	4,5,6	37%
Open ended Experiment		10%

COURSE OUTCOMES (COs)

At the end of the course, students will be able to

CO1	Apply and demonstrate the usage of various programming constructs.
CO2	Design and develop applications to solve problems across various technical and real world domains.
CO3	Explore various programming tools and techniques in a team.

CO-PO-PSO Matrix

CO	PO												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1			2											2		
CO2			2											2		
CO3					2									2		
AVG.			2		2									2		

Course Content

SEMESTER – III			
Course Name	**: Technical Proficiency Enhancement Course 1**	**Course Code: BISTP308**	
Number of Lecture Hours / Week	**:00**	**CIE Marks : 50**	
Number of Tutorial / Practical Hours / Week	**:02**	**SEE Marks : 50**	
Total Number of Lecture + Tutorial/Practical Hours	**: 24**	**SEE Duration : 02 Hours**	
L:T:P	**: 0:0:2**	**CREDITS : 01**	

MODULES	TEACHING HOURS
MODULE 1 **Introduction to Programming:** **Flowcharts:** Introduction to flowcharts, Decision making using flowcharts, Loops, Example problems. **Variables and Data types:** First program, Variables and data types, Taking input, How data is stored in memory, Arithmetic Operators. **Conditional statements:** Introduction to If else, Relational and logical operators, Nested conditionals.	**06**
MODULE 2 **Loops and Functions 1** While Loops: While loops, Flow of execution of statements in while loop, Example problems using while loop. **Patterns:** Introduction to patterns, Basic Patterns, Square Patterns, Triangular Patterns, Character Patterns, Reverse Triangle, Inverted patterns, Isosceles triangles.	**06**
MODULE 3 **Loops and Functions 2** **For Loops:** For loops, Break and Continue, increment - decrement operators. **Functions:** Introduction to functions, Working of function calling, Variables and its scope, Pass by value	**06**
MODULE 4 **Arrays** **Introduction to Arrays:** Introduction to arrays, How arrays are stored in memory, Passing arrays to functions. **Searching and Sorting:** Understanding Binary Search, Selection sort, Bubble sort, Insertion sort, Merging two sorted arrays.	**06**
MODULE 5 **Strings and 2D Arrays** **Strings:** Introduction to strings, storage of strings and their inbuilt functions, **2D Arrays:** 2D arrays, Storage of 2D arrays, Example problems using 2D Arrays	**06**

Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA | NAAC with 'A' Grade
Department of Information Science & Engineering
Phone: +91 821-4276210, Email: hodis@vvce.ac.in
Web: http://www.vvce.ac.in
@vvceofficial