ESTD : 1946

2022

OBE & CBCS

CURRICULUM

UNDER - GRADUATE PROGRAMME
Second Year

Department of Information Science & Engineering (2022 -2026)

THE NATIONAL INSTITUTE OF ENGINEERING
(An Autonomous Institute under Visvesvaraya Technology University, Belagavi)
Recognised by AICTE, New Delhi

North Campus, Hootagalli Industrial Area, Mysuru-570018.
Phone; 0821 - 4004900, 2481220; Fax: 0821 - 2485802
E-mail: info@nie.ac.in; Website: https://nie.ac.in/

VISION

“The Department will provide quality and value based education to produce innovative world-class computing engineers and will enhance quality research for the betterment of society”

MISSION

- To impart high quality training, education and competence in information science domain through best-in class faculty and facilities

- To produce globally acceptable information science graduates who can contribute professionally to the industry and research activities by offering courses on emerging technologies.

- To provide platforms to work effectively and innovatively in multi-disciplinary domain.

Programme Educational Objectives

PEO 1: Professionally successful in the field of Information and emerging technologies.
PEO 2: Successful in pursuing higher studies at globally recognized institutions.

Programme Specific Objectives

PSO 1: Apply the knowledge of information Systems in the field of Engineering to provide Solution through programming skills

PSO 2: Collaborate and communicate effectively with professionals in the field of computing, involve in continuous learning and address societal issues

Programme Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

The National Institute of Engineering

Scheme of Teaching & Examination - 2022

Effective from the Academic year 2023-24

Department: Information Science and Engineering

B.E. 2022 Admitted Batch

Semester : III

Sl.No	Type of Course	Course Code	Course Title	Teaching Department (TD)	Question Paper setting Board (PSB)	Teaching Hrs/Week				Examination				Credits
						L	T	P	S	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
1	PCC/BSC	BCS301	Mathematics for Computer Science	TD: Maths	PSB: Maths	3	2	0		3	50	50	100	**4**
2	IPCC	BCS302	Digital Design & Computer Organization	TD: IS	PSB: IS	3	0	2		3	50	50	100	**4**
3	IPCC	BCS303	Operating Systems	TD: IS	PSB: IS	3	0	2		3	50	50	100	**4**
4	PCC	BCS304	Data Structures and Applications	TD: IS	PSB: IS	3	0	0		3	50	50	100	**3**
5	PCCL	BCSL305	Data Structures Laboratory	TD: IS	PSB: IS	0	0	2		3	50	50	100	**1**
6	ESC	BCS306x	ESC/ETC/PLC	TD: IS	PSB: IS	2	0	2		3	50	50	100	**3**
7	UHV	BSCK307	Social Connect & Responsibilities	Any Department		0	0	2		1	100	_	100	**1**
8	AEC/SEC	BCS358x	Ability Enhancement Course (AEC)	TD: IS	PSB: IS	If the course is a Theory					50	50	100	**1**
						1	0	0		1				
						If the course is a Laboratory								
						0	0	2		2				
9	MC	BNSK359	National Service Scheme (NSS)	NSS Coordinator		0	0	2		_	100	_	100	**0**
		BPEK359	Physical Education (PE) (Sports & Athletics)	PED										
		BYOK359	Yoga	Yoga Teacher										
										Total	**550**	**350**	**900**	**21**

PCC: Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, MC: Mandatory Course (Non-credit), AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, L: Lecture, T: Tutorial, P: Practical: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation.K :This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

Engineering Science Course (ESC/ETC/PLC)

(Note- Student should opt for the course which should not be similar to the course opted in 1st Year)

BCS306A	Object Oriented Programming with Java	BCS306C	Unix and Shell Programming
BCS306B	Object Oriented Programming with C++	BCS306D	JavaScript

Ability Enhancement Course – III

BCS358A	Data Analytics with Excel	BCS358C	Project Management with Git
BCS358B	Data Analytics with R	BCS358D	Data Visualization with Python

Code: BCS301	**Course: Mathematics for Computer Science**
Credits: 4	**CIE: 50 Marks**
L:T:P – 3:2:0	**SEE: 50 Marks**
SEE Hours: 3	**Total Marks: 100 Marks**

Prerequisites if any	Mathematics1, Mathematics2
Course objectives	1. To introduce the concept of random variables, probability distributions, specific discrete and continuous distributions with practical application in Computer Science Engineering and social life situations. 2. To Provide the principles of statistical inferences and the basics of hypothesis testing with emphasis on some commonly encountered hypotheses. 3.To Determine whether an input has a statistically significant effect on the system's response through ANOVA testing.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Explain the basic concepts of probability, random variables, probability distribution	Understand
CO2	Apply suitable probability distribution models for the given scenario.	Apply
CO3	Apply the notion of a discrete-time Markov chain and n-step transition probabilities tosolve the given problem	Apply
CO4	Use statistical methodology and tools in the engineering problem-solving process.	Apply
CO5	Compute the confidence intervals for the mean of the population.	Analyze
CO6	Apply the ANOVA test related to engineering problems.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	-	-	-	2	-	-	-	-	-	-		3	-
CO2	3	-	2	-	-	-	-	-	-	-	-	-		3	-
CO3	3	-	-	3	-	-	-	-	-	-	-	-		-	2
CO4	3	-	-	-	1	-	-	-	-	-	-	-		-	2
CO5	3	-	-	-	1	-	-	-	-	-	-	-		-	2
CO6	3	-	-	3	-	-	-	-	-	-	-	-		-	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

	Modules	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
	Module – 1: Probability Distributions			
1.1	**Probability Distributions**: Review of basic probability theory. Random variables (discrete and continuous), probability mass and density functions. Mathematical expectation, mean and variance. Binomial, Poisson and normal distributions- problems (derivations for mean and standard deviation for Binomial and Poisson distributions only). Normal distribution.	10	2	-
	Module – 2: Joint probability distribution & Markov Chain			
2.1	Joint probability distribution: Joint Probability distribution for two discrete random variables, expectation, covariance and correlation. Markov Chain: Introduction to Stochastic Process, Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states.	10	2	-
	Module – 3: Statistical Inference 1			
3.1	Introduction, sampling distribution, standard error, testing of hypothesis, levels of significance, test of significances, confidence limits, simple sampling of attributes, test of significance for large samples, comparison of large samples.	10	2	-
	Module – 4: Statistical Inference 2			
4.1	Sampling variables, confidence limit for mean. Test of Significance for means of two small samples, students 't' distribution, Chi-square distribution as a test of goodness of fit. F-Distribution.	10	2	-
	Module – 5: Design of Experiments & ANOVA			
5.1	Principles of experimentation in design, Analysis of completely randomized design, randomized block design. The ANOVA Technique, Basic Principle of ANOVA, One-way ANOVA, Two-way ANOVA, Latin-square Design, and Analysis of Co-Variance.	10	2	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**10**	
	Total No. of Practical Hours			**00**

Textbook:

1. **Ronald E. Walpole, Raymond H Myers, Sharon L Myers & Keying Ye** "Probability & Statistics for Engineers & Scientists", Pearson Education, 9th edition, 2017.
2. **Peter Bruce, Andrew Bruce & Peter Gedeck** "Practical Statistics for Data Scientists" O'Reilly Media, Inc., 2nd edition **2020**.

Reference Book:

1. **Erwin Kreyszig**, “Advanced Engineering Mathematics”, John Wiley & Sons, 9th Edition, 2006.
2. **B. S. Grewal** “Higher Engineering Mathematics”, Khanna publishers, 44th Ed., 2021.
3. **G Haribaskaran** “Probability, Queuing Theory & Reliability Engineering”, Laxmi Publication, Latest Edition, 2006
4. **Irwin Miller & Marylees Miller,** John E. Freund’s “Mathematical Statistics with Applications” Pearson. Dorling Kindersley Pvt. Ltd. India, 8th edition, 2014.
5. **S C Gupta and V K Kapoor**, “Fundamentals of Mathematical Statistics”, S Chand and Company, Latest edition.
6. **Robert V. Hogg, Joseph W. McKean & Allen T. Craig**. “Introduction to Mathematical Statistics”, Pearson Education 7th edition, 2013.
7. **Jim Pitman**. Probability, Springer-Verlag, 1993.
8. **Sheldon M. Ross,** “Introduction to Probability Models” 11th edition. Elsevier, 2014.
9. **A. M. Yaglom and I. M. Yaglom**, “Probability and Information”. D. Reidel Publishing Company. Distributed by Hindustan Publishing Corporation (India) Delhi, 1983.
10. **P. G. Hoel, S. C. Port and C. J. Stone**, “Introduction to Probability Theory”, Universal Book Stall, (Reprint), 2003.
11. **S. Ross**, “A First Course in Probability”, Pearson Education India, 6th Ed., 2002.
12. **W. Feller**, “An Introduction to Probability Theory and its Applications”, Vol. 1, Wiley, 3rd 1968
13. **N.P. Bali and Manish Goyal**, A Textbook of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
14. **Veerarajan T**, Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi, 2010

Online Resources:

http://nptel.ac.in/courses.php?disciplineID=111 http://www.class-central.com/subject/math(MOOCs)http://academicearth.org/ http://www.bookstreet.in.
VTU EDUSAT PROGRAMME – 20
VTU e-Shikshana Program

Code: BCS302	**Course: Digital Design & Computer Organization**
Credits: 4	**CIE: 50 Marks**
L:T:P - 3:0:2	**SEE: 50 Marks**
SEE Hours: 3	**Total Marks:100**

Prerequisites if any	**Fundamentals of Logic**
Learning objectives	1. To provide the knowledge to explain the fundamentals of Logic circuits and combinational circuits. 2. To introduce the design of sequential circuit systems, Registers and Counters 3. The basics involved in number representation and arithmetic operations in the computer system. 4. Basic processor concept, instruction execution and Bus architecture, Memory architecture and mapping techniques.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Use logic minimization techniques and design combinational circuits using logic gates.	Apply
CO2	Design and analyze sequential circuits systems, Registers and Counters.	Apply
CO3	Use algorithms to perform fast multiplication, division and to represent floating point numbers in binary.	Apply
CO4	Explain the basic processing unit and design of control system, memory architecture and mapping techniques.	Understanding

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	1	1	-	-	-	1	1	-	1		-	2
CO2	3	3	2	1	1	-	-	-	1	1	-	1		-	2
CO3	3	2	2	-	-	-	-	-	2	2	-	1		-	3
CO4	3	2	2	-	-	-	-	-	2	2	-	1		-	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

	Module 1-Combinational Circuits	**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	**Simplification of Boolean Functions**: The Map Method, Two-and Three Variable Maps, Four- Variable Maps, Don't-Care Conditions	2	0	-
1.2	**Combinational Logic**: Adders, Subtractors	2	0	2
1.3	Binary Parallel Adder, Decimal Adder	3	0	1
1.4	Decoders, Multiplexer	3	0	2
	Module 2-Sequential Circuits			
2.1	**Sequential Logic**: Introduction, Flip-Flops, Triggering of Flip Flops	2	0	2
2.2	Analysis of Clocked Sequential Circuits	2	0	-
2.3	State Reduction and Assignment	1	0	-
2.4	Design Procedure	1	0	-
2.5	**Registers, Counters**: Introduction, Registers	1	0	-
2.6	Shift Registers – Serial Transfer, Bidirectional shift register with Parallel load	2	0	1

2.7	Synchronous counters Asynchronous Counters – Ripple Counters – Binary ripple counter	1	0	2
Module 3-Arithmetic Unit				
3.1	**Arithmetic unit:** Multiplication of Positive numbers A signed operand multiplication	2	0	-
3.2	Bit pair recoding of multipliers, carry save addition of summands	3	0	-
3.3	Integer division – Restoring and Non-restoring division	2	0	-
Module 4-The Processor				
4.1	**The Processor:** Fundamental concepts	1	0	-
4.2	Execution of complete instruction, Multiple-Bus organization	2	0	-
4.3	Micro programmed control unit – Organization of CU to allow conditional branching, Microroutine for instruction branch < 0	4	0	-
Module 5-Memory Unit				
5.1	**Memory Unit:** Basic concepts, Internal organization of memory chips, Structure of larger memory.	3	0	-
5.2	Cache memories, Mapping functions	3	0	-
Total No. of Lecture Hours		**40**	**-**	**-**
Total No. of Tutorial Hours			**00**	**-**
Total No. of Practical Hours				**10**

Integrated Lab Component: Digital Design & Computer Organization		
Sl. No	**COs**	**PART-A** **HARDWARE**
1.	CO1	a) Implement Half adder and Full adder using logic gates. b) Implement Half subtractor, Full subtractor using logic gates
2.	CO1	Implement BCD to Excess-3 using basic gates.
3.	CO1	Realize a full adder or any Boolean function using 3:8 decoder IC.
4.	CO1	Given any four variable logic expression, realize the simplified logic expression using 8:1 multiplexer
5.	CO2	Realize a JK master Slave Flip- Flop and verify its truth table.
6.	CO2	Realize mod-4 and mod-6 counter using Synchronous counter design
PART-B **SIMULATION USING XILINX**		
1.	CO1	Design Verilog HDL to implement simple circuits using structural, Data flow and Behavioural model.
2.	CO1	Write a VHDL code for a Half –Adder , Simulate and verify its working.
3.	CO1	Write a VHDL code for a Full –Adder , Simulate and verify its working.
4.	CO1	Write a VHDL code for a Full –Subtractor, Simulate and verify its working.
5.	CO1	Write a VHDL code for 2:1, 4:1 and 8:1 multiplexer , Simulate and verify its working.
6.	CO2	Write a VHDL code for a SR, D and JK flip flop , Simulate and verify its working.

Textbooks:

1. **Digital Logic and Computer Design :** M. Morris Mano, Pearson, 2016
2. **Computer Organization:** C Hamacher, Z Vranesic, S Zaky:**,** Tata McGraw Hill, 5th Edition, 2011.

Reference Books:

1. **Computer Architecture A Quantitative Approach:** John L Hennessy, David A Patterson, Elsevier, 5th Edition 2012.
2. **The Elements of Computing System – Building the Modern Computer from First Principles:** Noam Nisan, Shimon Schocken, The MIT Press (2005).
3. **Digital Principles and Applications:** Donald P Leach, Albert Paul Malvino & Goutham Saha, TMH, 6th Edition, 2006**.**

Online Resources:

1. **NPTEL:** Switching Circuits and logic designhttps://onlinecourses.nptel.ac.in/noc21_cs64
2. **NPTEL:** Computer Organization and Architecture A Pedagogical Aspect https://onlinecourses.nptel.ac.in/noc19_cs04/preview
3. Edx: Computation Structures 3:Computer Organization https://www.edx.org/course/computation-structures 3-computer-mitx-6-004-3x0
4. **Coursera:** Digital Systems - https://www.coursera.org/learn/digital-systems

Code: BCS303	**Course: Operating Systems**
Credits: 4	**CIE: 50 Marks**
L:T:P - 3:0:2	**SEE: 50 Marks**
SEE Hours: 3	**Total Marks: 100**

Prerequisites if any	None
Learning objectives	1. To demonstrate the need for Operating Systems. 2. To discuss about Process Scheduling, Memory management and Deadlock concepts.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand the process management policies, Process scheduling and the concepts of Process synchronization.	Understanding
CO2	Discuss various strategies of Memory Management; Paging and Segmentation.	Apply
CO3	Demonstrate the different Page Replacement Techniques and identify the design issues in paging.	Apply
CO4	Discuss suitable techniques to handle deadlock situations.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	-	2	-	-	-	-	-	-	2		2	2
CO2	2	2	2	-	2	-	-	-	-	-	-	2		2	2
CO3	3	2	2	-	2	-	-	-	-	-	-	2		2	2
CO4	2	3	2	-	2	-	-	-	-	-	-	2		2	2

Mapping Strength: **Strong– 3** **Medium – 2** **Low – 1**

Course Structure

Module – 1: Processes, Threads and Scheduling		**No. of Lecture Hours**	**No. of Tutorial Hours**	**No. of Practical Hours**
1.1	**Introduction:** what is an operating system? The Operating system as an extended machine, The Operating system as a resource manager	1	-	-
1.2	System calls for process Management	1	-	1
1.3	**Processes**: The process model, Process creation, Process termination	1	-	-
1.4	Process hierarchies, Process states.	1	-	-
1.5	**Threads**: Thread Usage, Thread Model	1	-	-
1.6	Implementing threads in user space, implementing threads in the kernel. Hybrid implementations	1	-	-
1.7	**Scheduling**: Introduction to scheduling, Scheduling in Batch Systems (FCFS, SJF, SRTN)	2	-	1
1.8	Scheduling in interactive systems: Priority based Scheduling, Round robin Scheduling	1	-	2
1.9	Guaranteed, Lottery and Fair Share Scheduling	1		
Module – 2: Interprocess Communication				
2.1	**Interprocess communication:** Race conditions, Critical regions	1	-	-
2.2	Mutual exclusion with busy waiting	1	-	-
2.3	Sleep and wakeup	1	-	-
2.4	Semaphores	1	-	1
2.5	Mutexes, Message Passing	1	-	-
2.6	**Classical IPC problems:** The dining philosophers' problem	1	-	-
2.7	The Readers and Writers problem.	1	-	-
Module – 3: Memory Management				
3.1	**Memory Management:** Background	1	-	-
3.2	Swapping	1	-	-
3.3	Contiguous Memory Allocation	1	-	1
3.4	Paging	1	-	-
3.5	Structure of the page Table	1	-	-
3.6	Segmentation.	1	-	-
3.7	**Virtual Memory**: Background; Demand Paging, copy-on write	1	-	-
3.8	Thrashing	1	-	-
Module – 4: Page replacement Algorithms and Design Issues				
4.1	**Page Replacement Algorithms**: The optimal page replacement algorithm	1	-	-
4.2	The not recently used page replacement algorithm	1	-	-
4.3	The first-in first-out.	1	-	1
4.4	The second chance page replacement algorithm	1	-	-
4.5	The clock page replacement algorithm, The least recently used page replacement Algorithm.	1	-	1
4.6	**Design issues for paging systems**: Local versus Global allocation policies, Load control, Page size	1	-	-
4.7	**File system implementation**: File System Layout, Implementing Files	1	-	-
4.8	Implementing Directories.	1	-	-
Module – 5: Deadlocks				
5.1	**Deadlocks**: **Introduction to deadlocks**: Conditions for deadlock, Deadlock modeling.	1	-	-
5.2	**Deadlock detection and recovery**: Deadlock detection with one resource of each type,	1	-	1
5.3	Deadlock detection with multiple resource of each type, Recovery from deadlock.	1	-	-

5.4	**Deadlock avoidance**: Resource trajectories, Safe and Unsafe States, The Banker's algorithm for a single resource,	1	-	1
5.5	The banker's algorithm for multiple resources.	1	-	1
5.6	**Deadlock prevention**: Attacking the Mutual-Exclusion Condition, Attacking the Hold-and-Wait condition, Attacking the No-Preemption condition, Attacking the Circular Wait condition	1	-	-
5.7	**Issues**: Two phase locking, Starvation	1	-	-
	Total No. of Lecture Hours	**40**	-	-
	Total No. of Tutorial Hours		**00**	-
	Total No. of Practical Hours			**10**

Integrated Lab Programs:

Sl. No.	COs	Programs
1.	CO1	Implement the process system calls fork (), exec () and wait ()
2.	CO1	a) Implement FCFS (First Come First Serve) scheduling algorithm b) Implement SJF (Shortest Job First) scheduling algorithm.
3.	CO1	Implement Priority Scheduling algorithm.
4.	CO1	Implement Round Robin Scheduling algorithm.
5.	CO1	Write a program to implement Producer-Consumer Problem using semaphores.
6.	CO2	Implement Memory Allocation methods for Fixed partition using First Fit, Worst Fit, Best Fit
7.	CO3	Write a program to implement FIFO page replacement algorithm.
8.	CO3	Write a program to implement LRU page replacement algorithm.
9.	CO4	Write a program to detect Deadlock.
10.	CO4	Write a program to implement Banker's algorithm for Deadlock avoidance.

Textbooks:

1. **Modern Operating systems, 4th Edition**, Andrew S.Tanenbaum, Herbert Bos, Pearson Education Limited; Global Edition. **(MODULE1, MODULE 2, MODULE 4, MODULE 5)**
2. **Operating System Concepts –** Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Wiley-India, **8th edition (MODULE 3)**

Note: Refer **Text book no.2** for problems on Process scheduling and Page Replacement Algorithms

Reference Books:

1. Operating Systems, William Stallings, PHI, Fourth Edition
2. Operating Systems, Milan Kovic, Tata McGraw Hill, 2001.
3. Operating System Design: v. 1: The Xinu Approach (Prentice-hall Software) Douglas Comer.
4. TUNIX Operating Systems: A Practical Approach, Robert Switzer, Prentice-Hall, 1993.

Online Resources:

1. http://www.nptel.ac.in/courses/106108101/

Course Code: BCS304
Credits: 3
L:T:P: 3:0:0
SEE Hours: 3

Course: Data Structures and Applications
CIE: 50 Marks
SEE: 50 Marks
Total Marks: 100

Prerequisites if any	**Problem Solving through Programming**
Learning objectives	1. Understand the fundamentals of data structures for problem solving, including linear structures like linked lists, stacks, and queues. 2. Apply non-linear data structures such as trees, sorting algorithms, and hash functions to solve real-world problems.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply the concepts of pointers and structures in problem solving.	Apply
CO2	Use different types of linked lists to solve problems.	Apply
CO3	Demonstrate stack and queue data structures to solve problems.	Apply
CO4	Illustrate the operations performed on tree data structures, hash functions for problem solving.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2	-	-	-	-	-	-	-	-	2		1	1
CO2	3	2	3	-	-	-	-	-	-	-	-	2		3	3
CO3	3	2	3	-	-	-	-	-	-	-	-	2		3	3
CO4	3	2	3	-	-	-	-	-	-	-	-	2		3	3

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No. of Lecture Hours	No. of Tutorial Hours
Module – 1 Pointers and Data Structures Fundamentals			
1.1	**Pointers:** Introduction	1	0
1.2	Structures, Nested Structures, Arrays of Structures	2	0
1.3	Structures and Functions, Self-Referential Structures	2	0
1.4	**Introduction to Data Structures,** Classification of data structures, Operations on data structures, Abstract data type	1	0
Module – 2 Linked Lists			
2.1	**Linked Lists -** Singly linked lists	3	0
2.2	Circular linked lists	2	0
2.3	Doubly linked lists	2	0
2.4	Circular doubly linked lists	2	0
Module – 3 Stacks and Queues			
3.1	**Stacks**: Introduction, Array Representation of Stacks	1	0
3.2	Operations on a Stack, Linked Representation of Stacks	1	0
3.3	**Applications of Stacks:** Conversion of an infix expression into a postfix expression, Evaluation of a postfix expression, Recursion.	3	0
3.4	**Queues:** Introduction, Array Representation of Queues, Linked Representation of Queues	2	0
3.5	**Types Of Queues:** Circular Queue, Priority Queues	2	0
Module – 4 Trees			
4.1	**Trees:** Introduction, Types of Trees	1	0
4.2	**Traversing** A Binary Tree,	1	0
4.3	Binary Search Trees, Operations on Binary Search Trees	3	0
4.4	**Threaded Binary Trees:** One-way Threading,	1	0
4.5	AVL Tree,Binary Heaps operations	3	0
Module – 5 Sorting and Hashing			
5.1	**Sorting:** Introduction to Sorting , Radix Sort , Heap Sort	2	0
5.2	**Hashing and Collision:** Introduction, Hash Tables, Different Hash Functions, Collisions, Pros and Cons of Hashing, Applications of Hashing	5	0
	Total No. of Lecture Hours	40	
	Total No. of Tutorial Hours		-

Text Books:

1. "**Data Structures using C**", Reema Thareja, 2nd Edition, 2018, Oxford University Press.
2. "**Data Structures using C**", Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, 2014, low price edition, Pearson education.

Reference Books:

1. Richar F Gilberg and Behronz A Forouzan, "**Data Structures, A Pseudocode Approach with C"**, 2nd Edition, 2012, Thomson.
2. Horowitz, Sahni, Anderson-Freed, "**Fundamentals of Data Structures in C**", 2nd Edition, 2011, Universities Press.

Online Resources:

1. NPTEL: Programming and Data structures- https://nptel.ac.in/courses/106/105/106105085/
2. Coursera: Data Structures - https://www.coursera.org/learn/data-structures
3. Programming & Data structures: http://nptel.ac.in/courses/106106130/
4. Programming, Data structures and Algorithms: http://nptel.ac.in/courses/106106133/

Course Code: BCSL305	**Course: Data Structures Laboratory**
Credits: 1	**CIE: 50 Marks**
L:T:P: 0:0:2	**SEE: 50 Marks**
SEE Hours: 2	**Total Marks: 100**

Prerequisites if any	**Fundamentals of C programming**
Learning objectives	1. To gain proficiency in applying advanced C concepts, such as pointers and dynamic memory allocation, for analyzing and solving complex problems. 2. To gain a comprehensive understanding of different data structures, including linked lists, stack, queue, binary trees, sorting algorithms, and hashing, enabling students to analyze, compare, and apply them effectively in various problem-solving contexts.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply concepts, such as pointers and dynamic memory allocation, to analyze and solve complex problems.	Apply
CO2	Create and evaluate different types of linked lists to develop robust and efficient applications.	Apply
CO3	Demonstrate the ability to analyze, compare, and apply stack, queue, binary tree, sorting, and hashing data structures for problem-solving in various contexts.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2	-	-	-	-	1	-	-	-	2		1	1
CO2	3	2	3	-	-	-	-	1	-	-	-	2		3	3
CO3	3	2	3	-	-	-	-	1	-	-	-	2		3	3

Mapping Strength : 3 – Strong 2 – Medium 1 – Low

List of Experiments

SL. No	CO	Experiment
1.	CO1	(a) Write a program to accept 3 integers and find the maximum among 3 numbers using functions and pointers. (b) Write a C program using pointer for searching the desired element from the array using pointers. (c) Write a program to find the maximum element in each row of the matrix using pointers.
2.	CO1	(a) Write a C program to read and display the Time in specified format. Create a structure called TIME with hour (int), minute(int), second(int) and next(self- referencing pointer) as its members. Dynamically create two variables of structure TIME and link the first variable to the second one and display it. Write a Display function that takes address of first TIME variable and displays both times in the format h:m:s. (b) Write a C program to read and display the student details. Define a structure 'Student' with fields name(string) , usn(int), marks of 3 subjects (int) and average(float) in it. Store the details of n students in an array of structure 'Student'. Display the details of all students in the descending order their total marks.
3.	CO2	(a) Write a C program using dynamic variables and pointers, to construct a singly linked list. The operations to be supported are: i) Insert at the front of a list. ii) Insert at any position in the list. iii) Deleting a node based on specified value. iv) Searching a node based on specified value. v) Displaying all the nodes in the list. (b) Write a C program to reverse a linked list elements.
4.	CO2	Write a C program to support the following operations on a doubly linked list where each node consists of integers. i) Create a doubly linked list by adding each node at the front. ii) Insert a new node to the left of the node whose key value is read as an input. iii) Delete the node of a given data, if it is found, otherwise display appropriate message. iv) Display the contents of the list.
5.	CO3	Write a program to design, Develop and Implement a menu- driven program in C for the following operations on STACK of integers (Array implementation of the stack with maximum size MAX = 4). i) Push an element on to stack. ii) Pop an element from the stack. iii) Check Overflow and Underflow situations on the stack. iv) Display the contents of stack. v) Exit. Support the program with appropriate functions for each of the above operations.
6.	CO3	a) Write a C program to convert an expression given in “infix” form to “postfix” form using stack concept. b) Check whether a given string is a palindrome or not by using a stack. c) Write a program to find the nth term in the Fibonacci series using recursion.
7.	CO3	a) Write a C program to simulate the working of a queues using an array provide and implement the following operations: i) Insert ii) Delete iii) Display Assume that the size of the queue is 5. b) Write a C program to implement a circular queue using linked lists.
8.	CO3	a) Write a C program to construct a binary search tree of integers and also display the elements in the tree using Inorder, Preorder and Postorder traversals. b) Write a C program to find the number of leaf nodes in a BST c) Write a C program to print all root to leaf paths of a BST.

9.	CO3	a) Write a C program to sort set of integers using radix sorting technique. b) Write a C program to search using closed hashing.

Textbooks:

1. **“Data Structures using C”,** Reema Thareja, 2nd Edition, 2018, Oxford University Press.
2. “**Data Structures using C**”, Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, 2014, low price edition, Pearson education.

Reference Books:

1. Aaron M Tenenbaum, YedidyahLangsam and Moshe J Augenstein, **“Data Structures using C”,** 2014, low price edition,Pearson education,.
2. Richar F Gilberg and Behronz A Forouzan, **“Data Structures, A Pseudocode Approach with C”,** 2nd Edition,2012, Thomson.
3. Horowitz, Sahni, Anderson-Freed, **“Fundamentals of Data Structures in C”,** 2nd Edition 2011, Universities Press.

Online Resources:

1. NPTEL: Programming and Data structures- https://nptel.ac.in/courses/106/105/106105085/
2. Coursera: **Data Structures -** https://www.coursera.org/learn/data-structures
3. **Programming & Data structures:** http://nptel.ac.in/courses/106106130/
4. **Programming, Data structures and Algorithms:** http://nptel.ac.in/courses/106106133/

ENGINEERING SCIENCE COURSE (ESC/ ETC/ PLC)

Code: BCS306A
Credits: 3
L:T:P - 2:0:2
SEE Hours: 3

Course: Object Oriented Programming with Java
CIE: 50 Marks
SEE: 50 Marks
Total Marks: 100

Prerequisites if any	**C programming**
Learning objectives	1. Distinguish Object-Oriented programming paradigm from Procedure-Oriented Programming 2. Use the Java programming language for various programming technologies.

Course Outcomes:
On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Develop Java programs using Object-Oriented paradigm	Understanding
CO2	Demonstrate Java code utilities in packages, interfaces and String class	Apply
CO3	Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	3	1	1	-	-	-	-	-	-	-		3	-
CO2	3	3	3	2	1	-	-	-	-	-	-	-		2	2
CO3	3	3	2	1	1	-	-	-	-	-	-	-		3	1

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Module – 1: Introduction to Object Oriented development		No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
1.1	What is Object Orientation? What is OO development? OO themes	1	-	-
1.2	Evidence for usefulness of OO development; OO modeling history	1	-	-
1.3	Modeling as Design Technique: Modeling; abstraction	1	-	-
1.4	The three models. Class Modeling: Object and class concepts	1	-	2
1.5	Link and associations concepts; Generalization and inheritance	1	-	-
Module – 2: Java & JDK				
2.1	Java's magic: Byte code; Java Development Kit(JDK)	1	-	-
2.2	The Java Buzzwords, Simple Java programs	1	-	-
2.3	Datatypes, arrays, Control Statements	1	-	1
2.4	Classes :Classes fundamentals; Declaring objects	1	-	1
2.5	Constructors	1	-	1
2.6	This keyword, garbage collection	1	-	-
Module – 3: Inheritance				
3.1	Inheritance: inheritance basics, using super	1	-	1
3.2	Creating multi-level hierarchy, method overriding	2		1
Module – 4: Packages & Interfaces				
4.1	Packages	1	-	1
4.2	Access Protection in packages	1	-	1
4.3	Importing Packages	1	-	1
4.4	Interfaces: references and variables, Default Interface methods	2	-	1
4.5	Exception handling in Java - fundamentals	1	-	-
Module – 5: Threads				
5.1	Multi-Threaded Programming: What are threads?	1	-	1
5.2	How to make the classes 'threadable'	1	-	1
5.3	Extending threads; Implementing 'runnable'	1	-	1
5.4	Thread Synchronization	2	-	1
	Total No. of Lecture Hours	**25**	-	-
		Total No. of Tutorial Hours	**00**	-
			Total No. of Practical Hours	**15**

Integrated Lab Programs:

Sl. No.	COs	Programs
1.	CO1	Develop basic Java programs using command-line arguments.
2.	CO1	Demonstrate Object-Oriented concepts in Java.
3.	CO1	Illustrate various forms of inheritance, method overloading and overriding.
4.	CO2	Create, import packages and use access protection.
5.	CO2	Demonstrate use of interfaces.
6.	CO3	Illustrate exception handling mechanism.
7.	CO3	Create customized exceptions.
8.	CO3	Demonstrate multithreaded programming.
9.	CO2	Illustrate usage of String class.

Textbook:

1. Herbert Schildt, ***Java The Complete Reference-Eleventh Edition***, McGraw Hill; 11th Edition.
2. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2ndEdition, Pearson Education/ PHI, 2007

Reference Book:

1. Dr. R. Nageswara Rao, Core Java, An Integrated Approach, Dream tech Press, 2016.
2. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008
3. Rajkumar Buyya, S Thamarai selvi, xing chenchu, Object oriented Programming with java, Tata McGraw Hill Education Private Limited, 2009
4. Richard A Johnson, An Introduction to Java Programming and Object-Oriented Application Development, Delmar Cengage Learning, 2007.
5. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2nd Edition, Pearson Education/ PHI, 2007.

Online Resources:

1. Udacity Free Course
   https://www.udacity.com/course/java-programming-basics--ud282
   Introduction To Programming In Java (by Evan Jones, Adam Marcus and Eugene Wu):
   https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-januaryiap-2010/
   Java Tutorial: https://www.w3schools.com/java/
   Java Tutorial: https://www.javatpoint.com/java-tutorial
   Java Tutorial: https://www.geeksforgeeks.org/java/

Course Code: BCS306B	**Course: Object Oriented Programming with C++**
Credits: 3	**CIE: 50 Marks**
L:T:P - 2:0:2	**SEE: 50 Marks**
SEE Hours: 3	**Total Marks: 100**

Prerequisites if any	**C Programming**
Learning objectives	1. To provide the knowledge of fundamental principles of Object-Oriented Programming. 2. To introduce the concepts of C++ language such as class, object, inheritance, overloading, virtual functions, STL etc.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain principles of Object-Oriented Programming using class and objects.	Understanding
CO2	Illustrate the concepts of function and operator overloading.	Apply
CO3	Demonstrate reusability using inheritance and virtual functions.	Apply
CO4	Apply function template and class template, STL and Exception Handling using real world problems for C++ programs.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	-	2	1	3	-	-	-	1	1	1	2		-	3
CO2	3	-	2	1	3	-	-	-	1	1	1	2		-	3
CO3	3	-	2	1	3	-	-	-	1	1	1	2		-	3
CO4	3	-	2	1	3	-	-	-	1	1	1	2		-	3

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No of Lecture Hours	No of Tutorial Hours	No of Practical Hours
Module-1: Classes and Objects				
1.1	**Classes and Objects:** An overview of C++**,**classes and objects, Relationship of Structure, Union and Class in C++	1	0	-
1.2	Friend Functions, Friend Classes, Inline Functions- Defining Inline Functions Within a Class	1	0	-
1.3	Constructors and Destructors, Parameterized Constructors-Constructors with One Parameter: A Special Case, copy constructor.	1	0	-
1.4	Static Class Members- Static Data Members and Static Member Functions, When Constructors and Destructors Are Executed, Passing Objects to Functions, Returning Objects.	1	0	1
1.5	**Arrays, pointers, References, and the Dynamic Allocation Operators:** 'this' Pointer, References – Reference Parameters, Passing References to Objects and Returning References.,	1	0	1
1.6	C++'s Dynamic Allocation Operators-Initializing Allocated Memory, Allocating Arrays and Allocating Objects	1	0	1
Module-2: Overloading				
2.1	**Function Overloading, Copy Constructor, and Default Arguments:** Function Overloading	1	0	-
2.2	Copy Constructor, Default Function Arguments, Default Argument vs. Overloading.	1	0	1
2.3	**Operator Overloading:** Creating a Member Operator Function-Creating Prefix and Postfix Forms of the Increment and Decrement Operators	1	0	1
2.4	Operator Overloading Using a Friend Function – Using a Friend to Overload ++ or --, Friend operator Functions Add Flexibility, Overloading << and >>.	2	0	1
Module – 3: Inheritance				
3.1	**Inheritance:** Base-Class Access Control, Inheritance and protected Members- Protected Base-Class Inheritance, Inheritance Multiple Base Classes	2	0	1
3.2	Constructors, Destructors, and Inheritance- When Constructors and Destructors are Executed, Passing Parameters to Base-Class Constructors. Granting Access, Virtual Base Classes.	1	0	1
3.3	**Virtual Functions and Polymorphism:** Virtual Functions- Calling a Virtual Function Through a Base- Class Reference, The Virtual Attribute vs. Inherited,	1	0	1
3.4	Virtual Functions Are Hierarchical, Pure Virtual Functions- Abstract Classes, Using Virtual Functions, Early vs. Late Binding	1	0	1
Module – 4: Generic functions				
4.1	**Templates:** Generic Functions- A Function with Two Generic Types, Explicitly Overloading a Generic Function,	1	0	-
4.2	Overloading a Function Template, Using Standard Parameters with Template Functions, Generic Function Restrictions.	1	0	1
4.3	**Applying Generic Function:** A generic Sort.	2	0	-
4.4	**Applying Generic Classes:** An Example with Two Generic Data Types, A Generic Array Class, Using Non-Type Arguments with Generic classes, Using Default Arguments with Template Classes.	1	0	1

Module – 5: Exception handling				
5.1	**Exception Handling:** Fundamentals, Handling derived class Exception	1	0	1
5.2	Exception Handling options.	1	0	1
5.3	**Introducing the Standard Template Library:** An Overview of the STL, Container Classes,	1	0	-
5.4	General Theory of Operation, Vector container class, Algorithms.	1	0	1
	Total No. of Lecture Hours	**25**	**-**	**-**
		Total No. of Tutorial Hours	**-**	**-**
			Total No. of Practical Hours	**15**

Integrated Lab Programs

Sl. No.	COs	Programs
1.	CO1	a. Write a C++ program to read three numbers from the keyboard and display the largest value on the screen. b. Write a C++ program to check whether the given number is an Armstrong number or not.(Hint: Armstrong number 153=1^3 + 5^3 + 3^3). c. Write a C++ program to find and display Factorial of a number. (Using an iterative method or recursion). d. Write a C++ program to print the ASCII value of a character. Write a C++ program that will accept an array of numbers and display the number of times the given number occurred in the array.
2.	CO1	a. Convert time from HH:MM:SS format to seconds using class in C++. b. A phone number, such as (044) 234-8900, can be thought of as having three parts: the area code (044), the exchange (234) and the number (8900). Write a program that uses a class to store these three parts of a phone number separately. Call the class phone. Create two class objects of type phone. Initialize one, and have the user input a number for the other one. Display both the numbers.
3.	CO1	Create two classes DM and DB which store the value of distances. DM stores distance in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.
4.	CO1	Write a C++ program to create a class Rectangle with data members: length, breadth, area and member functions as: Rectangle & set Dimensions (const int&, const int&) – sets the length and breadth of arguments using constant references. Use this pointer to return the resized Rectangle by reference. i. int computeArea() – to compute and returns area of a rectangle. ii. int computePerimeter() – to compute and return perimeter of a rectangle. ii. Two constructors, default constructor to initialize data members to zero and an overloaded constructor as arguments with breadth having a default value. Define all the member functions outside the class. Create objects of Rectangle type and test all the functions. Delete the objects before the program terminates.

5.	CO2	Write a C++ program to create a class called Complex and implement the following overloading member functions that return a Complex number after performing addition of input arguments. • Complex ADD (int a, Complex s) – where a is an integer (real part) and s is a complex number. • Complex ADD (Complex &s1, Complex &s2) Create objects of Rectangle type on heap utilizing both constructors. Use the member functions to modify the rectangle size, compute area, perimeter. Display the area & perimeter of each rectangle. Delete the objects before program termination.
6.	CO2	Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators ‘+’ and ‘-’. S1 = S1 + element; where S1 is the object of class STACK and element is an integer to be pushed on the top of stack. int element = S1--; where S1 is the object of class STACK. ‘—‘operator pops the top element. Handle the STACK empty and full conditions and also display the contents after every operation by overloading << operator.
7.	CO3	Write a C++ program to read and print Employee information (name, empID, gender) with Department (deptName, workAssigned) and with Loan information (loanDetails,loanAmt) using hierarchical inheritance.
8.	CO3	Write a C++ program to design a Student class representing USN and a Test class representing the scores of the student in various subjects and a Sports class representing the score in sports. The Sports and Test classes is inherited by Result class having the functionality to add the scores and display the final result of a student.
9.	CO3	Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students from UG and PG. Find the average age for all UG and PG students separately.

10.	CO3	Implement class Shape with the following specification:
		class Shape{ protected:
		float area, perimeter; public:
		Shape();
		virtual void initialize()=0; virtual float computeArea()=0; virtual float
		computePerimeter()=0; virtual
		~Shape();
		};
		Implement 2 classes Triangle and Rectangle publicly derived from class Shape, with suitable data members. Implement all the functions derived from class shape in each of the derived classes. Write a C++program to create objects of each of the derived class and assign to the base class (Shape) type pointer/reference. Demonstrate runtime polymorphism by calling the functions of the derived class objects by using the base class pointer/reference.

11.	CO4	Write two function templates in C++ to a. Sort the numbers. b. To search a given number Demonstrate the above functions on an array of integers and double. c. A point on the 2D can be represented by two numbers: an x co-ordinate and a y co- ordinate. The sum of two points can be defined as a new point whose x co- ordinate is the sum of x co-ordinates of both points and same for y co- ordinates. Using function template, find the third point in C++.
12.	CO4	a) Write a simple calculator using class template in C++ b) Write a program implementing stack and its operations using template class.
13.	CO4	Write a C++ program with the following: A function to read two double type numbers from the keyboard. a. A function to calculate the division of these two numbers. b. A try block to throw an exception when a wrong type of data is keyed in. c. A try block to detect and throw an exception if the condition ―divide-by-zero occurs. Appropriate catch block to handle the exceptions thrown.
14.	CO4	Perform these basic vector operation using Standard Template Library: Find the number of elements in the vector. i. Check whether the vector is empty or not. ii. Insert some elements into the vector. iii. Remove the element at a particular position. iv. Find the index of a particular element in a vector. b. Make a vector of random numbers and sort it in descending order using STL and also find its sum.

Textbooks:

1. **C++ The Complete Reference,** Herbert Schildt, TMH, McGraw-Hill, 4th Edition, 2017.

Reference Books:

1. **The C++ programming language,** Bjarnestroustrup , Pearson Education, 3rd Edition, 2013.
2. **C++ Primer**, Stanley B.Lippman and Josee Lajore, Addison Wesley, 3rd Edition, 2014.
3. **WEBLINK:** http://www.cplusplus.com/

Online Resources:

1. NPTEL: Programming in modern C++ https://nptel.ac.in/courses/noc22_cs43
2. Coursera: Programming in C++- https://www.coursera.org/specializations/hands-on-cpp

Code: BCS306C	**Course: Unix and Shell Programming**
Credits: 3	**CIE: 50 Marks**
L:T:P - 2:0:2	**SEE: 50 Marks**
SEE Hours: 3	**Total Marks:100**

Prerequisites if any	Programming Fundamentals and Operating System Knowledge
Learning objectives	1. Develop a solid understanding of Unix fundamentals, including the file system structure, navigation, and permissions, enabling efficient interaction with the Unix environment. 2. Acquire proficiency in shell scripting and system administration tasks, empowering the ability to automate processes, manage users and groups, and utilize advanced shell features for practical applications.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply effective navigation and interaction skills within the Unix operating system, demonstrating comprehension of its file system hierarchy, file types, and permissions.	Apply
CO2	Create and execute proficient shell scripts, employing variables, control structures, and conditional statements for automated task execution.	Analyze
CO3	Execute system administration tasks, manage user accounts, implement process management techniques, and employ networking commands in a Unix environment, showcasing the application of advanced skills.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	-	1	3	1	-	2	-	-	-	-		1	1
CO2	2	2	-	3	3	-	-	-	-	-	-	-		2	2
CO3	2	-	-	2	3	-	-	-	-	-	-	-		2	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Module – 1: Introduction and File Systems		No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
1.1	Background and Basic Commands, Introduction: Why UNIX, The UNIX environment, UNIX structure, Accessing UNIX, basic commands, other useful commands	2	-	1
1.2	File Systems: File names, File Types, Regular files, Directories, File System Implementation, Operations unique to directories, Operations unique to Regular files, Operations common to both.	3	-	2
Module – 2: Security and filters				
2.1	Security and File Permission: Users and groups, Security levels, Changing Permissions, User Masks, Changing Ownership and Group. Introduction to Shell	2	-	1
2.2	Filters: Filters and Pipes, Concatenating Files, Display Beginning and End of Files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count Characters, Words, or Lines	3	-	2
Module – 3: Regular expressions				
3.1	Communications: User Communication, Electronic Mail, File Transfer.	2	-	1
3.2	Regular Expressions and grep: Atoms,Operators, Operations, grep family, Examples, Searching for file content.	2	-	1
3.3	Awk: Execution, Fields and Records, Scripts, Operations, Patterns, Actions	2		1
Module – 4: Shell Scripting				
4.1	SHELL Scripting: Basic Script Concepts, Expressions, Decision Making Selections, Repetition, Special Parameters and Variables, Changing Positional Parameters, Argument Validation, Debugging Scripts.	4	-	3
Module – 5: Advanced programming				
5.1	Advanced Programming: Variable evaluation and Substitution, String Manipulation, Here Document, Functions, Arrays, Signals, Built-in Commands, Scripting Techniques, Shell Environment and Script.	5	-	3
	Total No. of Lecture Hours	**25**	**-**	**-**
		Total No. of Tutorial Hours	**00**	**-**
			Total No. of Practical Hours	**15**

SL.NO	COs	List of Experiments
1.	CO1	**Introduction to Unix Commands** (a) Write a shell script that displays your name and student ID when executed. (b) Use the 'ls' command to list files in a directory and apply different options to modify the output format.
2.	CO1	**Basic File Manipulation and Permissions** (a) Create a directory structure with nested subdirectories using 'mkdir'. (b) Copy files from one directory to another and modify file permissions using 'cp' and 'chmod'.
3.	CO2	**Simple Shell Scripting** (a) Write a shell script to calculate the sum of a series of numbers provided as command-line arguments. (b) Develop a script that prompts the user for their name and greets them using a personalized message.
4.	CO2	**Process Management and Job Control** (a) Use 'ps' and 'kill' commands to list and terminate processes. (b) Experiment with background and foreground job execution using '&' and 'fg' commands.

5.	CO2	**Shell Scripting with Control Structures** (a) Implement a script that checks if a given number is even or odd using conditional statements. (b) Create a script that generates the Fibonacci sequence within a specified range using loops.
6.	CO3	**Text Processing with 'sed' and 'awk'** (a) Use 'sed' to replace a specific word in a text file with another word. (b) Write an 'awk' script to extract and print specific columns from a CSV file.
7.	CO3	**User Management and Permissions** (a) Create a new user account and set user-specific environment variables. (b) Modify group memberships and observe the changes in user access.
8.	CO3	**Shell Scripting for System Tasks** (a) Write a script to display information about system memory usage. (b) Develop a script that monitors disk space utilization and sends an email alert if it exceeds a threshold.
9.	CO3	**Shell Customization and Aliases** (a) Create aliases for commonly used commands and demonstrate their usage. (b) Customize the shell prompt to display the current working directory and username.
10.	CO1, CO2, CO3	**Project-based experiments (Students can choose any one of the following)** **(a) Automation Script for File Management** Develop a shell script that automates the process of organizing files in a specified directory. The script should categorize files based on their extensions and move them to respective subdirectories. Provide options for users to specify source and destination directories and implement error handling for cases of missing directories or invalid inputs. **(b) System Monitoring and Reporting Tool** Create a comprehensive shell script that functions as a system monitoring and reporting tool. The script should gather information about system resources such as CPU usage, memory utilization, disk space, and network activity. Present this information in a readable format with clear headings and appropriate formatting. Allow users to specify reporting intervals and provide an option to save the report to a text file for later analysis.

Textbook:

1. UNIX and Shell Programming – A Textbook by Behrouz A Forouzan, Richard F Gilberg, Cengage Learning, 1st Edition, 2003.

Reference Book:

1. The Complete Reference UNIX by Kenneth Rosen, Douglas Host, James Farber and Richard Rosinski, Tata McGraw- Hill, Edition 2000.

Online Resources:

1. E-book: Shell Scripting – Expert Recipes for Linux, Bash and More by Steve Parker, Wrox Publications.
2. E-book: Linux Shell Scripting Cookbook by ShantanuTushar and SharathLakshman, 2nd edition, Packt Publications, 2013.

Code: BCS306D	**Course: JavaScript**
Credits: 3	**CIE: 50 Marks**
L:T:P - 2:0:2	**SEE: 50 Marks**
SEE Hours: 3	**Total Marks:100**

Prerequisites if any	Programming Fundamentals
Learning objectives	Learners will be able to: ● Compose syntactically correct JavaScript programs, including a functional "Hello World" example. They will confidently apply essential lexical elements such as comments, literals, and identifiers. ● Proficiently manipulate JavaScript's core data structures. They will aptly create, modify, and query object properties, employ functions to solve tasks, comprehend closures for encapsulation, and effectively utilize the JavaScript Standard Library to solve challenges involving sets, maps, regular expressions, and error handling.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply the knowledge of JavaScript lexical elements, data types, values, and variables to write programs.	Apply
CO2	Design and develop interactive web applications using object-oriented concepts, various control structures for statements and loops, and effectively manage data using arrays and objects.	Analyze
CO3	Create programs which uses Document Object Model, manipulate web documents and styles dynamically to enhance user experiences.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	1	-	-	3	-	-	-	-	-	-	-		1	1
CO2	2	1	2	1	3	-	-	-	-	-	-	2		3	3
CO3	2	1	3	2	3	-	-	-	-	-	-	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: Introduction to JavaScript				
1.1	Introduction to JavaScript: Exploring JavaScript, Hello World, A Tour of JavaScript	1	-	1
1.2	Lexical Structure: The Text of a JavaScript Program, Comments, Literals, Identifiers and Reserved Words, Unicode, Optional Semicolons	2	-	1
1.3	Types, Values, and Variables: Numbers, Text, Boolean Values, null and undefined, Symbols, The Global Object, Immutable Primitive Values and MutableObject References, Type Conversions	2	-	1
Module – 2: Expressions, Operators and Statements				
2.1	Expressions and Operators: Primary Expressions, Object and Array Initializers, Function Definition Expressions, Property Access Expressions, Invocation Expressions, Arithmetic Expressions, Relational Expressions, Logical Expressions, Assignment Expressions, Evaluation Expressions,	3	-	2
2.2	Statements: Expression Statements, Compound and Empty Statements, Conditionals, Loops, Jumps, Declarations	2	-	1
Module – 3: Objects, Arrays and functions				
3.1	Objects: Introduction to Objects, Creating Objects, Querying and Setting Properties, Deleting Properties, Testing Properties, Enumerating Properties, Extending Objects, Serializing Objects, Object Methods	2	-	1
3.2	Arrays: Creating Arrays, Reading and Writing Array Elements, Sparse Arrays, Array Length, Adding and Deleting Array Elements, Iterating Arrays, Multidimensional Arrays, Array Methods, Array-Like Objects, Strings as Arrays	2	-	1
3.3	Functions: Defining Functions, Invoking Functions, Function Arguments and Parameters, Functions as Values, Functions as Namespaces, Closures, Function Properties, Methods and Constructor	2	-	1
Module – 4: Classes				
4.1	Classes: Classes and Prototypes, Classes and Constructors, Classes with the class Keyword, Adding Methods to Existing Classes, Subclasses	2	-	2
4.2	The JavaScript Standard Library: Sets and Maps, Typed Arrays and Binary Data, Pattern Matching with Regular Expressions, Dates and Times, Error Classes	3	-	1
Module – 5: Iterators and Generators				
5.1	Iterators and Generators: How Iterators Work, Implementing Iterable Objects, Generators	2	-	1
5.2	JavaScript in Web Browsers: Web Programming Basics, Events, Scripting Documents, Scripting CSS, Document Geometry and Scrolling, Web Components	2	-	2
	Total No. of Lecture Hours	**25**	**-**	**-**
		Total No. of Tutorial Hours	**00**	**-**
			Total No. of Practical Hours	**15**

SL.NO	COs	List of Experiments
1.	CO1	**Type Conversions** a. Create a program that demonstrates different types of type conversions in JavaScript. b. Convert a number to a string, a string to a number, and a boolean to a number.
2.	CO2	**Object Properties** Create an object representing a person with properties like name, age, and occupation. Access and modify these properties using both dot notation and bracket notation.
3.	CO2	**Array Manipulation** Write a program that demonstrates various array operations. Create an array of numbers, add new elements to it, remove elements, and iterate through the array using loops and array methods.

4.	CO2	**Function and Closure** Define a function that takes two parameters and returns their sum. Create a closure that uses a variable from its containing function's scope.
5.	CO2	**Classes and Inheritance** Create a class representing a basic shape. Extend the class to create subclasses like Circle and Square. Add methods to calculate their areas.
6.	CO2	**Error Handling** Write a program that uses a try-catch block to handle an error. Attempt to access a property of an undefined object and handle the resulting exception.
7.	CO1	**String Manipulation** Write a program that takes a sentence as input and reverses the order of words using string manipulation techniques.
8.	CO1	**Working with Dates** Create a program that displays the current date and time. Format it in a user-friendly way, such as "August 10, 2023, 3:30 PM".
9.	CO3	**Event Handling and DOM Manipulation** Create a simple to-do list. Allow users to add items, mark items as completed, and remove items using event listeners and DOM manipulation.
10.	CO2	**Generators** Implement a simple generator that yields a sequence of even numbers up to a specified limit.
11.	CO2	**Regular Expressions for Data Validation** Create a form with input fields for email, password, and phone number. Use regular expressions to validate the user's input in real-time as they type.
12.	CO3	**Creating a Basic Web Component** Explore creating a basic web component using the `CustomElementRegistry`. Build a custom element for a simple counter that users can interact with.

13.	CO1, CO2, CO3	**Project-based experiments (Students can choose any one of the following) (CO1, CO2 and CO3)** **a. Task Manager Web App -** Build a task manager web application using HTML, CSS, and JavaScript. The app should have the following features: ● Task List: Display a list of tasks with their titles and due dates. ● Add Task: Allow users to add new tasks with titles and due dates. ● Mark Complete: Users can mark tasks as completed, which will visually differentiate them from incomplete tasks. ● Delete Task: Implement a way to delete tasks from the list. ● Local Storage: Store tasks in the browser's local storage so that they persist even after the user refreshes the page. This project will require you to work with HTML for creating the structure, CSS for styling, and JavaScript for handling user interactions, managing tasks, and storing data. OR **b. Interactive Quiz Game -** Create an interactive quiz game using HTML, CSS, and JavaScript. The app should have the following features: ● Quiz Questions: Prepare a set of multiple-choice questions along with their correct answers. ● Display Questions: Display one question at a time with its options. ● Answer Selection: Allow users to select an answer from the provided options. ● Score Tracking: Keep track of the user's score based on the number of correct answers. ● Results: After all questions are answered, display the user's total score and a congratulatory message. To make the quiz more engaging, you could also implement a countdown timer for each question. Additionally, consider using animations to transition between questions.

Textbook:

1. JavaScript: The Definitive Guide, David Flanagan, 7th Edition (Released May 2020), O'Reilly Media,

Reference Book:

1. JavaScript from Beginner to Professional, Laurence Lars Svekis, Maaike van Putten, Rob Percival, Packt Publishing Limited, 2021.

Online Resources:

1. https://ocw.mit.edu/courses/6-170-software-studio-spring-2013/pages/lecture-notes/
2. https://web.stanford.edu/class/cs98si/courseinfo.html

Course Code: BSCK307
Credits: Non- Credit Mandatory Course
L:T:P - 1:0:0

Course: Social Connect & Responsibilities
CIE: 100 Marks
Total Marks: 100

Prerequisites if any	NIL
Learning objectives	● To make students understand and appreciate the important societal/ environmental issues like conservation/ sustainability/ waste management and the like and inculcate ethical responsibility towards the same. ● Provide a formal platform for students to communicate and connect to their surroundings and enable them to have a responsible connection with society.

Course Outcomes:
On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Understand social responsibility	Understand
CO2	Practice sustainability and creativity	Apply
CO3	Demonstrate planning and organizational skills	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	-	-	-	-	-	2	1	3	1	1	-	2		1	1
CO2	-	-	-	-	-	2	1	3	2	2	-	2		1	1
CO3	-	-	-	-	-	2	1	3	3	3	-	1		2	1

S – Strong (3) M – Medium (2) L – Low (1)

Course Structure

Nos.	Module Name	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: Plantation /Heritage Walk				
1.1	Plantation and adoption of a tree: Plantation of a tree that will be adopted for four years by a group of B.E. students. They will also make an excerpt either as a documentary or a photoblog describing the plant's origin, its usage in daily life, and its appearance in folklore and literature	01	0	04
OR				
1.1	Heritage walk and crafts corner: Heritage tour, knowing the history and culture of the city, connecting to people around through their history, knowing the city and its craftsmen, photo blog and documentary on evolution and practice of various craft forms.	01	0	04
Module – 2: Organic farming/Food walk				
2.1	Organic farming and waste management: usefulness of organic farming, wet waste management in neighboring villages, and implementation in the campus.	01	0	04
OR				
2.2	Food Walk; City's culinary practices, food lore, and indigenous materials of the region used in cooking.	01	0	04
Module – 3: Water Conservation				
3.1	Water Conservation: knowing the present practices in the surrounding villages and implementation in the campus, documentary or photo blog presenting the current practices	01	0	04
	Total No. of Hours	**15**		

Activities:

Jamming session, open mic, and poetry: Platform to connect to others. Share the stories with others. **Share the experience of Social Connect**: Exhibit the talent like playing instruments, singing, one-act play, art-painting, and fine art.

Pedagogy:

The pedagogy will include interactive lectures, inspiring guest talks, field visits, social immersion and a course project.

Applying and synthesizing information from these sources to define the social problem to address and try to arrive at the solution through the course project, of your group.

Social immersion with NGOs/social sections will be a key part of the course.

Guideline for Assessment Process: Continuous Internal Evaluation (CIE)

After completion of the social connect, the student shall prepare, with daily diary as reference, a comprehensive report in consultation with the mentor/s to indicate what he has observed and learnt in the social connect period. The report should be signed by the mentor.

The course shall be evaluated on the basis of the following criteria and/or other relevant criteria pertaining to the activity completed / based on the rubrics approved by the DC.

Components	**Marks**
Marks allotted for the diary	10
Planning and scheduling the social connect	05
Information/Data collected during the social connect-	10
Analysis of the information/data and report writing	15
Final presentation	10
Total	**50**

ABILITY ENHANCEMENT COURSE

Code: BCS358A
Credits: 1
L:T:P - 1:0:0
SEE Hour: 1

Course: Data Analytics with Excel
CIE: 50 Marks
SEE: 50 Marks
Total Marks: 100

Prerequisites if any	Basic Computer Skills, Fundamental Excel Skills
Learning objectives	The learning objectives for a data analytics course with Excel typically revolve around equipping students with the knowledge and skills to effectively analyze and interpret data using Excel as a tool

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	To familiarize oneself with Excel's Basic features	Understanding
CO2	To gain skills on data visualization for data analysis using MS Excel.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	-	-	2	-	-	-	-	-	-	-		1	-
CO2	3	3	3	2	3	-	-	1	-	-	-	2		3	2

Mapping Strength: **Strong– 3** **Medium – 2** **Low – 1**

Course Structure

Sl. No.	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: I Basics of MS Excel				
1.1	Features of MS	2	-	-
1.2	Excel Worksheets and Workbooks: Labeling and Naming Worksheets and Workbooks, Adding, Deleting and Saving Worksheets and Workbooks, Reposition Worksheets, Inserting, Deleting, and Renaming Worksheets, Copy Worksheets, Printing a Workbook, Formatting a Worksheet, Adding Elements to a Workbook, Protecting Worksheet and Workbook.	2	-	-
Module – 2: Data Representation using MS Excel				
2.1	Import externa data, Creating a Table, Sorting Data into a Table, Data Validation, Consolidation	1	-	-
2.2	Defining Names in MS Excel, Macros: View Macros, Record Macros	2	-	-
2.3	Formulas and Functions: Creating a Formula, Formula Auditing, Meaning and Advantages of functions, Insert function, Use relative References, Mathematical Functions, Statistical Functions, Date & Time Functions	2	-	-
Module – 3: Data Visualization and Analysis				
3.1	Charts: Chart elements: Titles, legend, data labels, creating a New Chart, Formatting the Chat, Types of charts, Using Chart Templates	2	-	-
3.2	PivotTables: Creating a PivotTable, Filtering and Sorting a PivotTable, Using Slicers to manipulate PivotTables, Creating a PivotChart	2	-	-
3.3	Filtering Data: Creating a Custom AutoFilter, Using an Advanced Filter. Data Group, Ungroup and Subtotals	2	-	-
Total No. of Lecture Hours		**15**	**-**	**-**
Total No. of Tutorial Hours			**00**	**-**
Total No. of Practical Hours				**00**

Textbook:

1. "Microsoft Excel 2019 Step by Step" by Curtis Frye
2. "Excel 2019 Bible" by Michael Alexander, Richard Kusleika, and John Walkenbach

Reference Book:

1. "Microsoft Excel Data Analysis and Business Modeling" by Wayne L. Winston:
2. "Excel 2019 All-in-One For Dummies" by Greg Harvey:

Online Resources:

1. https://www.academia.edu/42074058/Excel_2019_BIBLE
2. https://ptgmedia.pearsoncmg.com/images/9780735681019/samplepages/9780735681019.pdf

Code: BCS358B
Credits: 1
L:T:P - 1:0:0
SEE Hour: 1

Course: Data Analytics with R
CIE: 50 Marks
SEE: 50 Marks
Total Marks: 100

Prerequisites if any	Basics of any programming language and knowledge of statistics and mathematics.
Learning objectives	1. Understand the basics of data analytics and its importance in decision-making processes. 2. Utilize the R programming language for data manipulation and analysis tasks.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply the R programming language to perform data analysis tasks.	Analysis
CO2	Apply statistical concepts and techniques to analyze data.	Analysis

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	-	-	-	2	-	-	-	-	-	-	3		3	3
CO2	2	-	-	-	2	-	-	-	-	-	-	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: Introduction to R				
1.1	Introduction, Downloading and Installing Data Types in R.	1	-	-
1.2	RIDEs and Text Editors, Handling Packages in R, Working with Directory.	1	-	-
1.3	Few Commands for Data Exploration.	2	-	-
Module – 2: Loading and Handling Data in R				
2.1	Introduction, Challenges of Analytical Data Processing, Expression, Variables and Functions.	2	-	-
2.2	Missing Values Treatment in R, Using the 'as' Operator to Change the Structure of Data.	2	-	-
2.3	Vectors, Few Common Analytical Tasks, Methods for Reading Data.	1	-	-
Module-3:Exploring Data in R				
3.1	Introduction, Data Frames, R Functions for Understanding Data in Data Frames.	2	-	-
3.2	Load Data Frames, Exploring Data.	2	-	-
3.3	Descriptive Statistics, Spotting Problems in Data with Visualization.	2	-	-
	Total No. of Lecture Hours	**15**	**-**	**-**
		Total No. of Tutorial Hours	**00**	**-**
			Total No. of Practical Hours	**00**

Textbook:

1. **Data Analytics Using R by Seema Acharya, MC GRAW HILL INDIA,** © Published: April 20, 2018.

Reference Book:

1. Jared P Lander, R for everyone: advanced analytics and graphics, Pearson Education, 2013

Online Resources:

1. https://dokumen.pub/data-analytics-using-r-paperback-jan-01-2018-seema-acharya-9352605241-9789352605248.html

Code: BCS358C
Credits: 01
L:T:P -1:0:0
SEE Hour: 1

Course: Project Management with Git
CIE: 50 Marks
SEE: 50 Marks
Total Marks: 100

Prerequisites if any	NIL
Learning objectives	1. To make the students understand the basic concepts related to Version controller with Git. 2. To provide knowledge of Installation procedure and basic commands in Git.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Describe basic concepts related to Version controller with GiT.	Understanding
CO2	Explain Installation procedure and basic commands in GiT.	Understanding

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	2	-	-	1	-	-	-	-	-	-	-		2	1
CO2	2	2	-	-	1	-	-	-	-	-	-	-		2	1

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

Sl. No.	Module	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1				
1.1	Setting Up and Basic Commands Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.	1	-	-
1.2	Creating and Managing Branches Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."	1	-	-
1.3	Creating and Managing Branches Write the commands to stash your changes, switch branches, and then apply the stashed changes.	1	-	-
1.4	Collaboration and Remote Repositories Clone a remote Git repository to your local machine.	1	-	-
1.5	Collaboration and Remote Repositories Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.	1	-	-
1.6	Collaboration and Remote Repositories Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge	2	-	-
Module – 2				
2.1	Git Tags and Releases Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.	1	-	-
2.2	Advanced Git Operations Write the command to cherry-pick a range of commits from "source-branch" to the current branch.	1	-	-
2.3	Analysing and Changing Git History Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?	1	-	-
2.4	Analysing and Changing Git History Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."	2	-	-
2.5	Analysing and Changing Git History Write the command to display the last five commits in the repository's history.	2		
2.6	Analysing and Changing Git History Write the command to undo the changes introduced by the commit with the ID "abc123".	1		
	Total No. of Lecture Hours	**15**	-	-
	Total No. of Tutorial Hours		**00**	-
	Total No. of Practical Hours			**00**

Textbook:

1. Git: Version Control for Everyone Beginner's Guide Copyright © 2013 Packt, Publishing Ravishankar Somasundaram.

Reference Book:

1. Version Control with Git by Jon Loeliger, Copyright © 2009 Jon Loeliger. All rights reserved. Printed in the United States of America. ISBN: 978-0-596-52012-0 [M]1242320486

Online Resources:

1. https://www.udemy.com/course/introduction-to-version-control-with-git-and-github/

Code: BCS358D
Credits: 1
L:T:P - 1:0:0
SEE Hour: 1

Course: Data Visualization with Python
CIE: 50 Marks
SEE: 50 Marks
Total Marks: 100

Prerequisites if any	Introduction to python programming
Learning objectives	To understand the basics of data visualizations techniques To understand the different plotting libraries and apply in real world scenarios.

Course Outcomes:
On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Understand the basics of data visualization techniques and libraries	Understanding
CO2	Apply the data visualization techniques in the real world problems	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	-	-	-	1	2	-	-	-	-	-	-	-		2	1
CO2	-	-	-	-	3	-	-	-	-	-	-	-		2	1

Mapping Strength: **Strong– 3** **Medium – 2** **Low – 1**

Course Structure

		No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module – 1: The Importance of Data Visualization in Business Intelligence				
1.1	Why Is Data Visualization Important? Why Do Modern Businesses Need Data Visualization? The Future of Data Visualization, How Data Visualization Is Used for Business Decision-Making, Introducing Data Visualization Techniques	3	-	-
1.2	Data Gathering and Cleaning, Cleaning Data, Checking for Missing Values, Handling the Missing Values, Reading and Cleaning CSV Data, Merging and Integrating Data	2	-	-
Module – 2: Data Exploring and Analysis				
2.1	Series Data Structures, Data Frame Data Structures	2	-	-
2.2	Data Analysis, Statistical Analysis, Data Grouping, Iterating Through Groups, Aggregations, Transformations, Filtration	3	-	-
Module – 3: Data Visualization				
3.1	Direct Plotting, Seaborn Plotting System	3	-	-
3.2	Matplotlib Plot	2	-	-
Total No. of Lecture Hours		**15**	**-**	**-**
Total No. of Tutorial Hours			**00**	**-**
Total No. of Practical Hours				**00**

Textbook:

1. Data Analysis and Visualization Using Python, Dr. Ossama Embarak, 2018
2. Python for Data Analysis by Wes McKinney, 2nd Edition, October 2017, O'Reilly Media Publisher

Online resources

1. https://doi.org/10.1007/978-1-4842-4109-7
2. https://www.oreilly.com/library/view/python-for-data/9781491957653/
3. https://www.oreilly.com/library/view/interactive-data-visualization/9781449340223/