SCHEME FOR EXAMINATION
B.TECH (FOUR YEAR) DEGREE COURSE
SECOND YEAR, INFORMATION TECHNOLOGY
SEMESTER IV
EFFECTIVE FROM SESSION 2021-22

SL. NO.	SUBJECT CODE	SUBJECTS	PERIODS/ WEEK			EVALUATION SCHEME			CREDITS
			L	T	P	IA	ESE	TOTAL	
THEORY									
1	IT204TPC01	DISCRETE MATHEMATICS	3	1	0	30	70	100	4
2	IT204TPC02	COMPUTER ORGANIZATION & ARCHITECTURE	3	0	0	30	70	100	3
3	IT204TPC03	OPERATING SYSTEMS	3	0	0	30	70	100	3
4	IT204TPC04	DESIGN & ANALYSIS OF ALGORITHMS	3	0	0	30	70	100	3
5	IT204THS02	MANAGEMENT 1 – MANAGEMENT PROCESS AND ORGANIZATIONAL BEHAVIOUR	3	0	0	30	70	100	3
PRACTICAL									
1	IT204PPC01	COMPUTER ORGANIZATION & ARCHITECTURE LAB	0	0	4	30	20	50	2
2	IT204PPC02	OPERATING SYSTEMS LAB	0	0	4	30	20	50	2
3	IT204PPC03	IT WORKSHOP	1	0	2	30	20	50	2
TOTAL CREDITS									**22**
IA- INTERNAL ASSESSMENT, ESE-END SEMESTER EXAMINATION, L-LECTURE, T-TUTORIAL, P-PRACTICAL									

SCHOOL OF STUDIES OF ENGINEERING & TECHNOLOGY
GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.) (A CENTRAL UNIVERSITY)
B. TECH. INFORMATION TECHNOLOGY

Programme Outcomes: Graduates will be able to:

PO1: Fundamentals: Apply knowledge of mathematics, science and engineering.

PO2: Problem analysis: Identify, formulate and solve real time engineering problems using first principles.

PO3: Design: Design engineering systems complying with public health, safety, cultural, societal and environmental considerations

PO4: Investigation: Investigate complex problems by analysis and interpreting the data to synthesize valid solution.

PO5: Tools: Predict and model by using creative techniques, skills and IT tools necessary for modern engineering practice.

PO6: Society: Apply the knowledge to assess societal, health, safety, legal and cultural issues for practicing engineering profession.

PO7: Environment: Understand the importance of the environment for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics, and responsibilities and norms of the engineering practice.

PO9: Teamwork: Function effectively as an individual and as a member or leader in diverse teams and multidisciplinary settings.

PO10: Communication: Communicate effectively by presentations and writing reports.

PO11: Management: Manage projects in multidisciplinary environments as member or a team leader.

PO12: Life-long learning: Engage in independent lifelong learning in the broadest context of technological change.

Programme Specific Outcomes:

PSO1: To apply knowledge of recent computing technologies, skills and current tools of Information Technology Engineering.

PSO2: To design software systems, components or processes to meet identified needs within economic, environmental and social constraints.

PSO3: To explore research gaps, analyze and carry out research in the specialized/emerging areas.

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204TPC01	3	1	0	4 HOURS	30	70	4

DISCRETE MATHEMATICS

Course Objective:

1. To introduce a number of discrete mathematical structures found to be serving as tools in the Development of theoretical computer science.
2. Course focuses on how discrete structures actually helped computer engineers to solve problems Occurred in the development of programming languages
3. Course highlights the importance of discrete structures towards simulation of a problem in computer science engineering.
4. To apply logical reasoning to solve a variety of problems.
5. Model and analyze computational processes using analytic and combinatorial methods.

Unit 1:

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's Diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.

Unit 2:

Basic counting technique s-inclusion and exclusion, pigeon-hole principle, Permutation and combination, Principle of Mathematical Induction, The Well Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor, Euclidean Algorithm, The Fundamental Theorem of Arithmetic.

Unit 3:

Propositional Logic: Basic Connectives and Truth Tables, Logical Equivalence the Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers.

Proof Techniques: Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Prof by Contraposition, Proof of Necessity and Sufficiency.

Unit 4:

Algebraic Structures and Morphism: Algebraic Structures with one binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields, Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form.

Unit 5:

Graphs and Trees: Graphic and their properties, Degree. Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Graph Colouring, Colouring maps and Planar Graphs, Four colour conjecture, trees and rooted trees, binary trees.

Test books:

1. Kenneth II. Rosen, Discrete Mathematics and its Applications, Tau McGraw-Hill.
2. Susanna S. Epp, Discrete Mathematics with Applications, 4th edition. Wadsworth Publishing Co. Inc. J.
3. C L. Liu and D P Mahapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw-Hill.

Reference Book:

1. J. P. Tremblay and R Manor, Discrete Mathematical Structure and its Application to Computer science, TMG Edition, Tata McGraw-Hill.
2. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press, Schaum's Outlines Series, Seymour lipschutz, Mare Lipson.

Course Outcomes: After completion of this course students will be:

1. Able to apply mathematical logic and Boolean algebra in switching circuits & logic circuits.
2. Familiar with set theory, relation and functions.
3. Familiar with algebraic structures, graph theory and combinatorics.
4. Able to solve problems in various fields in computer science, specially networking.
5. Model problems in Computer Science using graphs and trees.

Course Outcomes and their mapping with Programme Outcomes:

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	3	1							3	2	2	2
CO2	3	3	2	1	2							3	3	2	3
CO3	3	3	2	2	1							3	3	2	3
CO4	3	3	3	2	3							3	3	3	3
CO5	3	2	3	3	3							3	3	2	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204TPC02	**3**	**0**	**0**	**3 hours**	**30**	**70**	**3**

COMPUTER ORGANIZATION & ARCHITECTURE

Course Objectives:
CO1: Conceptualize the basics of organizational and architectural,
CO2: Learn about various basic arithmetic operation
CO3: Learn about various control unit design and Input-output subsystems
CO4: Understand the basics pipeline.
CO5: Understand the basics Memory organization and their basic working.

UNIT 1
Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU – registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction sets of some common CPUs.

UNIT 2
Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

UNIT 3
Introduction to x86 architecture. CPU control unit design: hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU. Memory system design: semiconductor memory technologies, memory organization. Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes – role of interrupts in process state transitions, I/O device interfaces – SCII, USB

UNIT 4
Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards. Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

UNIT 5

Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Suggested books:

1. “Computer Organization and Design: The Hardware/Software Interface”, 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
2. “Computer Organization and Embedded Systems”, 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

Suggested reference books:

1. “Computer Architecture and Organization”, 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2. “Computer Organization and Architecture: Designing for Performance”, 10th Edition by William Stallings, Pearson Education.
3. “Computer System Design and Architecture”, 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course Outcomes:

After the course the students are expected to be able to

1: Demonstrate computer organization and architecture concepts of a computer system

2: Describe the Computer arithmetic operation algorithm and hardware

3: Understand the basics of hardwired and micro-programmed control of the CPU, Memory, I/O system

4: Describe fundamentals concepts of pipeline and issues

5: Describe the memory hierarchy and related function,

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	2	1	1							3	3	3	1
CO2	3	3	3	3	3							3	3	3	3
CO3	3	3	2	2	3							3	3	3	3
CO4	3	2	1	2	2							3	3	3	3
CO5	3	2	2	2	2							3	3	3	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204TPC03	3	0	4	3 hours	30	70	3

OPERATING SYSTEMS

Objectives of the course

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management

UNIT I - INTRODUCTION TO OPERATING SYSTEM:

Objective and function of operating system. The evaluation of the operating system, system components operating system services, system structure, batch interactive, time sharing and real time operating system, Protection. File system: File concepts, file organization and access mechanism.

UNIT II - CONCURRENT PROCESS:

Process concepts, principal of concurrency. The producer consumer problem, the critical section problem , semaphore, classical problem in concurrency , inter process communication , process generation , process scheduling .

UNIT III - CPU SCHEDULING:

Scheduling concepts, performance criteria scheduling algorithms. Algorithm evaluation, multiprocessor scheduling. I/O management and Disk scheduling I/O devices and organization of the I/O functions. I/O buffering disk I/O operating system design issues.

UNIT IV - DEAD LOCKS:

System models, deadlock characterization, prevention, avoidance and detection recovery from deadlock, combined approach.

UNIT V - MEMORY MANAGEMENT:

Base machine , Residence monitor , multiprogramming with fixed partition , multiprogramming with variable partitions, multiple base register, paging, segmentation, paging segmentation, virtual memory concepts , demand paging performance , page replacement algorithms, allocation of frames, thrashing , cache memory organization impact on performance .

Course Outcomes

1. Create processes and threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.
3. Specification of memory organization develops the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement file management system.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

Reference Books

Milenkovic M. , “Operating System concepts”, MGH

Tanenbaubm A. S. “Operating System design and implementation” , PHI

Silberschartz A.and Patterson J.I. , “ Operating system concepts”, Wisley.

Stilling William “ Operating System “, Maxwell McMillan International Edition 1992.

Dectel H.N. , “An introduction to operating system “, Addision Wisley.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	3	2	1	2							3	2	3
CO2	2	1	2	3	1	2							3	3	2
CO3	2	1	2	2	2	3							3	3	3
CO4	2	1	2	2	2	2							3	3	3
CO5	1	2	2	2	2	3							2	3	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204TPC04	3	0	0	3 hours	30	70	3

DESIGN & ANALYSIS OF ALGORITHMS

Course Objectives

1. To develop proficiency in problem solving and programming.
2. To be able to carry out the Analysis of various Algorithms for mainly Time and Space Complexity.
3. To get a good understanding of applications of Data Structures.
4. To develop a base for advanced study in Computer Science.
5. To teach various advanced design and analysis techniques such as greedy algorithms, dynamic programming & Know the concepts of tractable and intractable problems and the classes P, NP and NP-complete problems.

Unit 1:
Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

Unit 2:
Fundamental Algorithmic Strategies: Brute-Force, Greedy, Dynamic Programming, Branchand-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving , Bin Packing, Knap Sack TSP. Heuristics – characteristics and their application domains.

Unit 3:
Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

Unit 4:
Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.

Unit 5:
Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE

Suggested books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.

2. Fundamentals of Algorithms – E. Horowitz et al.

Suggested reference books

1. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Course Outcomes

1. For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.
2. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming.
5. To understand an analyses approximation algorithms, Randomized algorithms, NP and P SPACE

Course Outcomes and their mapping with Programme Outcomes:

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	2	3								3	2	2
CO2	2	3	2	3	2								2	3	3
CO3	3	2	2	3	3								2	3	2
CO4	3	2	3	2	2								3	2	3
CO5	2	2	3	2	2								2	3	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204THS02	3	0	0	3 hours	30	70	3

MANAGEMENT PROCESS AND ORGANIZATIONAL BEHAVIOUR

Course Objectives:

1. To help the students to develop cognizance of the importance of Management processes.
2. To enable students to describe how people behave under different conditions and understand why people behave as they do.
3. To provide the students to analyse specific strategic human resources demands for future action.
4. To enable students to synthesize related information and evaluate options for the most logical and optimal solution.
5. To learn how to face the challenges of an Organization.

Course Outcomes (Cos):

On completion of this course, the students will be able to

1. To learn the basic concepts of Organizational Behaviour and its applications in contemporary organizations.
2. To understand how individual, groups and structure have impacts on the organizational effectiveness and efficiency.
3. To appreciate the theories and models of organizations in the workplace.
4. To creatively and innovatively engage in solving organizational challenges.
5. To learn and appreciate different cultures and diversity in the workplace.

UNIT -I

School of Management Thought: Evolution of Management thought, Systems and Contingency approach of management, Decision Theory School.

UNIT -II

Managerial processes, functions, skills and roles in an organization. Nature, process and technique of planning, Organizing, Staffing, Directing, Coordinating, Control.

UNIT -III

Organizational Behavior: Concept, Significance, Understanding and Managing individual behavior – Personality, Perceptions, Values, Attitudes, Learning, Work-motivation, Individual Decision Making and Problem solving.

UNIT -IV

Understanding and Managing Group Processes: Interpersonal and Group dynamics. Applications of emotional intelligence in organizations. Group decision making. Leadership and Influence Process : Concept, styles and Theories.

UNIT – V

Understanding and Managing Organizational Systems, Organizational Conflict – sources, pattern levels and types of conflict. Organizational design and structure. Work stress.

Suggested Readings

1. Koontz, Harold, Cyril O'Donnell, and Heinz, Whelrich. Essentials of Management. New Delhi: Tata Mc Graw Hill.
2. Robbins, S.P. Organizational Behaviour. New Delhi: PHI.
3. Luthans, F. Organisational Behaviour. NewYork: Mc Graw Hill.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	2	2	2	2	3	1	2	3	-	1	1	1	2
CO2	2	3	2	2	2	2	3	3	2	3	-	1	1	3	3
CO3	2	3	3	3	3	2	1	2	3	3	-	2	1	3	3
CO4	2	3	3	3	3	3	3	2	3	3	-	2	1	2	3
CO5	2	3	3	2	3	2	2	3	3	3	-	2	1	2	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204PPC01	0	0	4	4 HOURS	30	20	2

COMPUTER ORGANIZATION & ARCHITECTURE LAB

Course Objectives:

1. Conceptualize the basics of organizational and architectural.
2. Learn about implementation of various operation in assembly code.
3. Learn about various basic arithmetic operation
4. Learn about various digital circuit design
5. Design and simulation in Logisim, GNUsim8085 or another simulator.

List of Practical's

1. Write the working of 8085 simulators GNUsim8085 and basic architecture of 8085 along with small introduction.
2. Study the complete instruction set of 8085 and write the instructions in the instruction set of 8085 along with examples.
3. Write an assembly language code in GNUsim8085 to implement data transfer instruction.
4. Write an assembly language code in GNUsim8085 to store numbers in reverse order in memory location.
5. Write an assembly language code in GNUsim8085 to implement arithmetic instruction.
6. Write an assembly language code in GNUsim8085 to add two numbers using lxi instruction.
7. Write an assembly language code in GNUsim8085 to add two 8 bit numbers stored in memory and also storing the carry.
8. Write an assembly language code in GNUsim8085 to find the factorial of a number.
9. Write an assembly language code in GNUsim8085 to implement logical instructions.
10. Write an assembly language code in GNUsim8085 to implement stack and branch instructions.
11. Design Ripple Carry Adder using simulator
12. Design Carry-Look-Ahead Adder using simulator
13. Design Registers and Counters using simulator
14. Design Combinational Multipliers using simulator
15. Booth's Multiplier using simulator
16. Arithmetic Logic Unit using simulator
17. Memory Design using simulator
18. Write a C program to implement Booth's algorithm for multiplication.
19. To write a C program for multiplication of two binary numbers
20. To write C program for sum of two binary number
21. Write a program to implement Division Algorithm.

22. To Study the Logisim Tool
23. Design of Half-adder circuit
24. Design of Full-adder circuit
25. Design of Ripple -adder circuit

Course Outcomes:

After the course the students are expected to be able to

1: Demonstrate computer organization and architecture concepts of a computer system

2: Understand assembly code implementation.

3: Describe the Computer arithmetic operation algorithm and hardware

4: Understand the basics of digital circuit design

5: learn to use various tools

References:

1. https://cse.iitkgp.ac.in/~chitta/coldvl/comp.html

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3							3	3	3	2
CO2	3	3	3	3	3							3	3	3	2
CO3	3	3	3	3	3							3	3	3	2
CO4	3	3	3	3	3							3	3	3	2
CO5	3	3	3	3	3							3	3	3	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204PPC02	**0**	**0**	**4**	**4 HOURS**	**30**	**20**	**2**

OPERATING SYSTEMS LAB

Course Objectives

1. To learn the fundamentals of Operating Systems.

2. To learn the mechanisms of OS to handle processes and threads and their communication

3. To learn the mechanisms involved in memory management in contemporary OS

4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols

5. To know the components and management aspects of concurrency management

List of Experiments

1. Simulate the following CPU scheduling algorithms.
a)FCFS b) SJF c) Round Robin d) Priority.

2. Write a C program to simulate producer-consumer problem using Semaphores

3. Write a C program to simulate the concept of Dining-philosophers problem.

4. Write a C program to simulate the following contiguous memory allocation Techniques
Worst fit b) Best fit c) First fit.

5. Simulate all page replacement algorithms a)
FIFO b) LRU c) OPTIMAL

6. Simulate all File Organization Techniques
a) Single level directory b) Two leveldirectory

7: Simulate Bankers Algorithm for Dead Lock Avoidance and dead lock prevention.

8. Write a program to simulate disk scheduling algorithms.
a) FCFS b) SCAN c) C-SCAN

Course Outcomes

1. Create processes and threads.

2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.

3. Specification of memory organization develops the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.

4. Design and implement file management system.

5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	3	2	1	2							3	2	3
CO2	2	1	2	3	1	2							3	3	2
CO3	2	1	2	2	2	3							3	3	3
CO4	2	1	2	2	2	2							3	3	3
CO5	1	2	2	2	2	3							2	3	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT204PPC03	1	0	2	2 HOURS	30	20	2

IT WORKSHOP

Course Learning Objectives:

1. To illustrate the concept of java and familiar and aware with basics.
2. To describe the mathematical logics about programing language.
3. To relate theoretical concepts with problem solving approach.
4. To provides the basic knowledge of use appropriate
5. To integrate classroom learning into different case study.

1. Write a program to perform multiplication of two matrices.

2. Write a program to find the volume of a box having its side w,h,d means width ,height and depth. Its volume is v=w*h*d and also find the surface area given by the formula s=2(wh+hd+dw). use appropriate constructors for the above.

3. Develop a program to illustrate a copy constructor so that a string may be duplicated into another variable either by assignment or copying.

4. Create a base class called shape. Apart from Constructors, It contains two methods getxyvalue() and showxyvalue() for accepting co-ordinates and to display the same. Create the subclass Called Rectangle which contains a method to display the length and breadth of the rectangle called showxyvalue().Illustrate the concepts of Overriding and Constructor call sequence.

5. Write a program that creates an abstract class called dimension, creates two subclasses, rectangle and triangle. Include appropriate methods for both the subclass that calculate and display the area of the rectangle and triangle.

6. Write a program, which throws Arithmetic Exception. Write another class (in a different file) that handles the Exception.

7. Write a program to sort a stream of Strings.

8. Create a user defined Exception class which throws Exception when the user inputs the marks greater than 100 Catch it and again rethrow it.

9. Write a program in which a Mythread class is created by extending the Thread class. In another class, create objects of the Mythread class and run them. In the run method print “GGVIT” 10 times. Identify each thread by setting the name.

10. Write a program to illustrate various thread methods.

11. Write a Program to implement Bank Account Class which illustrates the concept of Synchronization.

12. Write Program to illustrate the use of Vector Class and Iterator Interface.

REFERENCES BOOKS:

1. Java2 complete reference - Herbert schildt (TMH)
2. Java programming – E Balagurusamy
3. Java 2 Black book – Steven Holzner
4. Java Examples in a nutshell – O' Reilly

Course Outcomes:

1. Ability to learn about programming language.
2. Ability to understand the java programming language.
3. Be able to implement, compile, test and run Java programs comprising more than one class, to address a particular software problem.
4. Demonstrate the ability to use simple data structures like arrays in a Java program.

Course Outcomes and their mapping with Programme Outcomes:

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3								3	3	2
CO2	3	3	3	3	3								3	3	2
CO3	3	3	3	3	3								3	3	2
CO4	3	3	3	2	2								3	3	3
CO5	3	3	3	2	2								3	3	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly