Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology

Vidyanagari, Baramati, Dist. – Pune 413133
An Autonomous Institute Approved by AICTE and affiliated to SPPU,Pune

Department of Computer Engineering

Curriculum Structure and Syllabus of S. Y. B. Tech Computer Engineering (Course 2023)

With effective from Academic Year 2024-25

INSTITUTE VISION AND MISSION

VISION

To achieve Academic Excellence through Persistent and Synergic Collaborations amongst all Stakeholders.

MISSION

1. To ensure holistic development of students as lifelong learners and problem solvers through value based quality education.

2. To motivate faculty to attain the state-of-the-art knowledge and wisdom in their domain and be a facilitator towards co- creation of knowledge

3. To frame and deploy conducive and empowering policies for multifaceted growth of students, faculty and staff to make them contributors towards excellence.

4. To partner with industry for mutually beneficial relations to generate employable and deployable workforce.

5. To fulfill the aspirations of alumni, parents, society, region and nation at large by generating technically competent.

DEPARTMENT VISION AND MISSION

VISION

To achieve excellence in the field of Computer Engineering with consistent and collaborative efforts of every individual

MISSION

1. To develop students with fundamental advanced tools and technologies to work as skilled Computer professionals with ethical values.

2. To promote faculty for higher education and expose them to current trends to enrich educational quality.

3. To provide appropriate environment with required resources to achieve academic excellence.

4. To develop hand-in-hand relations with industries for catering institute-industry needs.

5. To apply collaborative efforts to make students competent to provide solutions to social problems.

Program Specific Outcomes (PSO)

PSO1: Professional Skills

- The ability to understand, analyse and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying.

PSO2: Problem-Solving Skills

- The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3: Successful Career and Entrepreneurship

- The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

Program Educational Objectives (PEO)

1. Students will be able to apply the fundamentals, domain knowledge and modern technology of computer engineering to analyse, design and implement effective solutions to engineering problems

2. Students will be able to identify the needs of society and deals with professional ethics, sense of responsibilities, and understanding of legal, safety, health, cultural and environmental issues

3. Students will be motivated for lifelong learning, investigative approach, multidisciplinary thinking and competitive exams

4. Students will be able to achieve successful career in different roles and responsibilities

5. Students will be nurtured for strong managerial and communication skills to work as an individual and team member

Program Outcomes (POs)		
Learners are expected to know and be able to		
PO1	**Engineering knowledge**	Apply the knowledge of mathematics, science, Engineering fundamentals, and an Engineering specialization to the solution of complex Engineering problems.
PO2	**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.
PO3	**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.
PO4	**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.
PO5	**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.
PO6	**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.
PO7	**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.
PO8	**Ethics**	Apply ethical principles and commit to professional ethics and responsibilities and norms of Engineering practice.
PO9	**Individual and Team Work**	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication Skills**	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.
PO11	**Project Management and Finance**	Demonstrate knowledge and understanding of 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.
PO12	**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.

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology
Board of Studies : Computer Engineering

Syllabus: Second Year (SY B. Tech.) Computer Engineering
w.e.f. AY:2023-2024

SEMESTER-III

Course Code	Courses Name	Teaching Scheme			Examination Scheme and Marks							Credits				
		TH	PR	TUT	Activity	ISE	ESE	TW	PR	OR	Total	TH	PR	OR	TUT	Total
CO23201	Discrete Mathematics	3	-	-	20	20	70	-	-	-	110	3	-	-	-	3
CO23202	Data Structures and Algorithms	3	-	-	20	20	70	-	-	-	110	3	-	-	-	3
CO23203	Digital Electronics and Computer Organization	3	2	-	20	20	70	20	-	20	130	3	-	-	-	4
CO23204	Object Oriented Paradigm	3	-	-	20	20	70	-	-	-	110	3	-	-	-	3
MDCO23201	Multi-disciplinary minor:	2	2	-	20	20	50	20	-	-	110	2	1	-	-	3
[illegible]230XX	Open Elective	2	-	-	-	-	50	-	-	-	50	2	-	-	-	2
CO23205	Data Analytics and Visualization using Python	-	4	-	-	-	-	40	20	-	60		2	-	-	2
CO23206	Data Structures Lab	-	4	-	-	-	-	40	40	-	80		2	-	-	2
Total		16	12	0	100	100	380	120	60	20	780	16	6	0	0	22

SEMESTER-IV

Course Code	Courses Name	Teaching Scheme			Examination Scheme and Marks							Credits				
		TH	PR	TUT	Activity	ISE	ESE	TW	PR	OR	Total	TH	PR	OR	TUT	Total
BS23201	Advanced Mathematics for Computer Engineering	3	-	1	-	20	70	20	-	-	110	3	-	-	1	4
CO23211	Microprocessor	3	2	-	20	20	70	20	-	20	150	3	1	-	-	4
[illegible]O23212	Database Management System	3	2	-	20	20	70	20	20	-	150	3	1	-	-	4
CO23213	Operating System	3	2	-	20	20	70	20	-	20	150	3	1	-	-	4
CO23214	Software Engineering	3	-	-	20	20	70	-	-	-	110	3	-	-	-	3
MDCO23201	Multi-disciplinary minor:	2	2	-	20	20	50	20	-	-	110	2	1	-	-	3
Total		17	8	1	100	120	400	100	20	40	780	17	3	1	1	22

Dept. Autonomy Coordinator
Mr. M. D. Shelar

Academic Coordinator
Dr. P. M. Paithane

Head of Department
Dr. G. J. Chhajed

Dean Autonomy
Dr. C. B. Nayak

Dean Academic
Dr. S. M. Bhosle

Principal
Dr. R. S. Bichkar

VPKBIET BARAMATI
Affiliated to S.P.P.U; Pune
Id No./PU/PN/Engg./152/2000
PUNE (MS.)

Principal
Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering & Technology, Baramati
Vidyanagari, Baramati-413133

Multidisciplinary Minor (MDM) Subjects			
AI23051	AI & Machine Learning	ET23053	Internet of Things
AI23052	Data Science	CE23051	Waste Management
AI23053	Generative AI	CE23052	Green Building & Smart Cities
CO23051	Cloud Computing	ME23051	Introduction to 3D Printing Technologies
CO23052	High Performance Computing	ME23052	Introduction to Robotics & Automation
CO23053	Computer Graphics & Gaming	EL23051	Solar Tech
IT23051	Cyber Security	EL23052	Industrial Automation
IT23052	Full Stack Development	GS23051	Nano Technology
ET23051	Embedded Systems	GS23052	Linear Algebra and Statistics
ET23052	Drone Technology		

Open Electives (OE) Subjects			
OE2301	Digital Marketing	OE2311	Biotechnology
OE2302	Professional Leadership	OE2312	International Relations
OE2303	Organizational Behavior	OE2313	Universal Human Values
OE2304	Industrial Management	OE2314	Education Technology
OE2305	Disaster Management	OE2315	Design Thinking
OE2306	Energy Economic & Management	OE2316	Financial Literacy for Bharat#
OE2307	Operation Research	OE2317	Sustainability & Climate Change
OE2308	Intellectual Property Rights	OE2318	Agriculture Technology
OE2309	Cyber Laws	OE2319	Architectural Technology
OE2310	Bioinformatics		

CO23202: Discrete Mathematics		
Teaching Scheme: TH: 03 Hrs/Week	**Credit: 03**	**Examination Scheme:** Course Activity: 20 Mark In Semester: 20 Mark End Semester: 70 Mark

Prerequisite: Basic Mathematics

Course Objective:

- To introduce students to understand, explain, and apply the foundational mathematical concepts at the core of computer science.
- To apply appropriate set, function, and relation models to analyze practical examples.
- To acquire skills in logic and proof techniques to enhance mathematical maturity.
- To acquire a comprehensive understanding of set theory, graph theory, and algebraic structures.
- To formulate problems precisely, solve the problems, apply formal proof techniques and explain the reasoning clearly

Course Outcomes:

1. To **Acquire** Knowledge of sets and logics for solving the real world problems
2. To **Recognize** and **Analyze** Relations, Functions, and their Characteristics
3. To **apply** advanced counting techniques such as the Inclusion-Exclusion principle and generating functions to solve complex counting problems and optimize problem-solving strategies.
4. To **explore** the concepts of group theory and their applications for solving the advance technological problems.
5. Illustrate the principles & concepts of graph theory for solving problems related to computer science.
6. Utilize tree structures as a modeling tool for solving algorithmic problems, demonstrating the ability to convert real-world problems into tree representations.

Guideline for Couse Activity:

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Projects
2. Industry Visit
3. Seminar
4. Research Paper
5. Group Discussion

Course Contents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Set Theory and Logics**	**07 Hours**

Introduction and Significance of Discrete Mathematics in Computer Engineering, Application areas in Computer Engineering. **Set Theory:** Introduction to Set, Set Representation, Types of Sets, Power set, Set Operations, Principle of Inclusion and Exclusion. **Logics and Proofs:** Propositions, Conditional Propositions, Truth Tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference. Predicate Logic: First order predicate, well-formed formula of predicate, Universal and Existential Quantifiers, Translating English Statements into Propositions, Mathematical Induction.

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Relation & Function	07 Hours

Relation: Definition of Relation, Properties of Binary Relations, Closure of Relations, Warshall"s Algorithm,Equivalence Relations and Equivalence Classes, Partitions, Partial Ordering Relations (POSET), Hasse Diagrams and Lattices, Properties of lattices – Bounded, Complemented, Distributed, Modular and Complete lattice. **Function**: Function Definition, Composition of Functions, Injective, Surjective and Bijective Function, Inverse of a Function, Growth of Functions.

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Counting Principles	07 Hours

Introduction to Counting- rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Advanced Counting Techniques: Inclusion-Exclusion principle, Pigeonhole principle. Recurrence Relations: Basics of recurrence relations Solving linear recurrence relations.

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Group Theory	07Hours

Definition, Basic Properties, Groups, Semi-group & Monoid, Abelian group, Subgroup, Normal subgroup, Groups and Coding, Group Homomorphism's, Rings, Integral Domain and Field. **Case Study:** Application of Group Theory in Computer Engineering.

Mapping of Course Outcomes for Unit V	CO5

UNIT V	Graph Theory	07 Hours

Graph : Basic Terminology and Special Types of Graphs, Representation of graphs using adjacency matrix and adjacency list, , Paths and Circuits, Hamiltonian and Euler Paths and Circuits, Isomorphic Graphs, Planer Graph,Single source shortest path- Dijkstra's Algorithm, Graph coloring. **Case Study:** Applications of Graph Theory in Computer Engineering.

Mapping of Course Outcomes for Unit VI	CO6

UNIT VI	Trees	07 Hours

Introduction, properties of trees, Rooted Trees, Binary search tree, tree traversal, Prefix Codes, Huffman Algorithm for Optimal Tree, Spanning Trees, Minimum Spanning Trees, Kruskal"s and Prim's Algorithm for Minimum Spanning Tree. **Case Study:** Applications of Trees in Computer Engineering.

Books and Other Resources

Text Books:

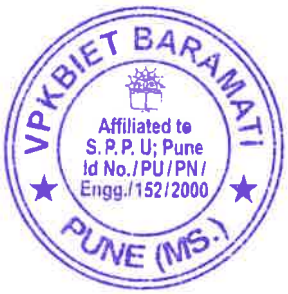
1. "C. L. Liu, "Elements of Discrete Mathematics"‖, TMH, ISBN 10:0-07-066913-9.
2. "N. Biggs, "Discrete Mathematics", 3rd Ed, Oxford University Press, ISBN 0 –19-850717–8.
3. Kenneth H. Rosen, "Discrete Mathematics and its Applications"‖, Tata McGraw-Hill, ISBN 978- 0-07-288008-3
4. Narsingh Deo, "Graph with application to Engineering and Computer Science", Prentice Hall of India, 1990, 0 – 87692 – 145 – 4
5. Eric Gossett, "Discrete Mathematical Structures with Proofs", Wiley India Ltd, ISBN:978-81- 265-2758-8

E Books & Videos:

1. https://www.ebookphp.com/discrete-mathematical-structures-6th-edition-epub-pdf/
2. http://discrete.openmathbooks.org/pdfs/dmoi-tablet.pdf
3. http://home.iitk.ac.in/~arlal/book/mth202.pdf
4. https://web.stanford.edu/class/cs103x/cs103x-notes.pdf
5. http://home.iitk.ac.in/~arlal/book/mth202.pdf

MOOC/ Video Lectures available at:

1. https://www.nptel.ac.in/courses/106/106/106106094/
2. https://nptel.ac.in/courses/106/106/106106183/
3. https://nptel.ac.in/courses/106/103/106103205/
4. https://nptel.ac.in/courses/106/105/106105192/
5. https://nptel.ac.in/courses/111/106/111106050/
6. https://nptel.ac.in/courses/111/106/111106102/



CO23202: Data Structure & Algorithms		
Teaching Scheme: TH: 03 Hrs/Week	**Credit: 03**	**Examination Scheme:** Course Activity: 20 Mark In Semester: 20 Mark End Semester: 70 Mark

Prerequisite: C, C++ Programming

Course Objective:

1. To study data structures, their implementations and applications
2. To study different searching and sorting techniques
3. To develop a logic for graphical modelling of the real life problems
4. To build the logic to use appropriate data structure in logical and computational solutions
5. To choose the appropriate data structure and algorithm design method for a specified application

Course Outcomes:

1. To understand linear data structures and basics of algorithm analysis
2. To implement searching and sorting algorithms and calculate their complexity
3. To develop applications by using stack and queue
4. To design and apply Linked list as a data structures in the application development
5. To design and apply tree and graph as a data structures in the application development
6. To design and apply hashing and multi way trees structures in the application development

Course Activity :

For the assessment of course activity must complete at least one activities out of following
1. Course mini project design using java
2. Implementation of Data Structure assignments using java

Course Contents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Introduction**	**06 Hours**

Introduction to Data Structures -: Concept of data, Data object, Data structure, Concept of Primitive and non-primitive, linear and Nonlinear, static and dynamic, persistent and ephemeral data structures,Definition of ADT. **Analysis of algorithm:** Frequency count and its importance in analysis of an algorithm, Time Complexity & Space complexity of an algorithm Big 'O', 'Ω' and 'Θ' notations. **Sequential Organization:** Single and multidimensional array and address calculation

Mapping of Course Outcomes for Unit II	CO2	
UNIT II	**Searching and Sorting**	**06 Hours**

Searching and sorting: Need of searching and sorting, Concept of internal and external sorting, sort Stability, Searching methods: Linear and binary search algorithms, Fibonacci Search.
Sorting methods: Bubble sort, selection sort, insertion sort, Quick sort, Merge sort, shell sort, Bucket sort and comparison of all sorting methods

Mapping of Course Outcomes for Unit III	CO3	
UNIT III	**Stack &Queue**	**08 Hours**

Stack: Concept of stack, stack as an ADT using sequential and linked organization, Applications of stack: recursion, converting expressions from infix to postfix or prefix form, evaluating postfix or prefix form
Queue: Concept of queues as ADT, Implementation of queue using array and linked organization, Concept of

circular queue, double ended queue, priority queue, Applications of queue.

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Linked List	08 Hours

Types of linked list- Linear and circular linked lists, Doubly Linked List and operations, Circular Linked List, singly circular linked list, doubly circular linked list, Polynomial Manipulations - Polynomial addition, Multiplication of two polynomials using linked list. Generalized Linked List (GLL)

Mapping of Course Outcomes for Unit V	CO5

UNIT V	Tree and Graph	08 Hours

Tree: Trees and binary trees-concept and terminology, Expression tree, Binary tree as an ADT, Binary search tree, Recursive and Non recursive algorithms for binary tree traversals, Binary search tree as ADT (Insert Search Delete, level wise Display), Concept of threaded binary tree. **Height Balanced Tree:** AVL tree.

Graph : Concept and terminologies, Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Breadth First Search traversal, Depth First Search traversal, Prim's and Kruskal' algorithms for minimum spanning tree, Shortest path using Dijkstra's algorithm.

Mapping of Course Outcomes for Unit VI	CO6

UNIT VI	Advanced Data Structures	06 Hours

Hash Table: Hash Table- Concepts-hash table, hash function, bucket, collision, overflow, open hashing, closed hashing, perfect hash function, hash functions- properties of good hash function, division, multiplication, extraction, mid-square, folding and universal, Collision resolution strategies- open addressing and chaining without replacement, open addressing and chaining with replacement. **Multiway Trees:** B-Tree, B+ Tree.

Books and Other Resources

Text Books:

1. "Fundamentals of Data Structures in C", E. Horowitz, S. Sahni, S. Anderson-freed, Second Edition, 2008, University Press, ISBN 978-81-7371-605-8

2. "The C Programming Language", B. Kernighan, D. Ritchie, Second Edition, 2015, Pearson Education India; ISBN 81-203-0596-5

Reference Books:

1. "Data Structures using C", Y. Langsam, M. Augenstin and A. Tannenbaum, First Edition, 2002, Pearson Education Asia, ISBN 978-81-317-0229-1
   "Computer organization" , Hamacher and Zaky, Fifth Edition
2. "Fundamentals of Data Structures in C++", Ellis Horowitz, S. Sahni, D. Mehta, 2nd Edition, 2008, University Press, ISBN-10: 8173716064
3. "An introduction to data structures with Applications", Jean-Paul Tremblay, Paul. G. Soresan, 2nd Edition, 1984, Tata Mc-Graw Hill International Editions, ISBN-0-07-462471-7

CO23203: Digital Electronics and Computer Organization		
Teaching Scheme: TH: 03 Hrs/Week PR: 02 Hrs/Week	**Credit: 04** **TH Credit :03** **PR Credit : 01**	**Examination Scheme:** Course Activity: 20 Mark In-Semester : 20 Mark End-Semester: 70 Mark Termwork : 20 Mark Oral : 20 Mark

Prerequisite:

Basis Electronics Engineering

Course Objective:

- To understand the Boolean Logic and its simplification
- To learn the implementation of basic combinational and sequential circuits
- To understand the fundamentals of Computer Organization
- To understand the Processor and Instructions
- To learn the basics of memory system in Computer

Course Outcomes:

1. Perform basic binary arithmetic & simplify logic expressions.
2. Implementation of Combinational Logic Functions
3. Use of Flip-flop for implementation of Counters and Registers
4. Understand the organization of Computer and its structure
5. Perform Computer Arithmetic and production of microinstructions
6. Describe an assortment of memory types and I/O devices

Guideline for Couse Activity:

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Projects
2. Industry Visit
3. Seminar
4. Research Paper
5. Group Discussion

Course Contents

Mapping of Course Outcomes for Unit I		**CO1**
UNIT I	**Number System, Logic Gate and Boolean Algebra**	**08 Hours**

Number system: Number Conversion, Representation of Binary Numbers: Sign Magnitude, 1's complement, 2's Complement, Binary Codes. **Logic Gates**: Positive and Negative Logic, Truth table, Logic Gates **Boolean Algebra**: Theorems, Simplification Techniques: Sum of Product, Product of Sum, K-Map, Quine–McCluskey Tabular Method

Mapping of Course Outcomes for Unit II		**CO2**
UNIT II	**Combinational Circuits**	**06 Hours**

Introduction, Clock Signal, Combinational Circuit Implementation: Half Adder-Subtractor, Full Adder-Subtractor, BCD Adder, Parity Generator and Checker, Comparator, Multiplexures and Demultiplexures		
Mapping of Course Outcomes for Unit III	**CO3**	
UNIT III	**Sequential Circuits**	**06 Hours**
Introduction to Flip-Flop, Types of Flip-Flop, Flip-Flop Conversion: JK to SR, JK to D, JK to T, SR to JK. **Registers**: Types, Shift Registers Counters: Synchronous and Asynchronous		
Mapping of Course Outcomes for Unit IV	**CO4**	
UNIT IV	**Basic Structure of Computer and Computer Arithmetic**	**08 Hours**
Von Neumann Architecture, Functional units of Computer, Basic Operational Concept – Address, Data, Control Bus structure, Memory location and Address, Memory operations **Arithmetic**: Addition and Subtraction of Signed Numbers, Design of Fast Adder, Multiplication of Unsigned Numbers, Multiplication of signed Numbers -- Booth Multiplication, Integer Division, Floating-Point Numbers and Operations.		
Mapping of Course Outcomes for Unit V	**CO5**	
UNIT V	**Control Processing Unit**	**08 Hours**
Processor: Registers and Its types **Instructions**: Elements, Format and representation Addressing modes **Basic Processing Unit**: Execution of a Complete Instruction, Hardware Components, Instruction cycle, Instruction Pipelining, Pipelining Hazards, Control Signals, Hardwired Control and Micro-program Control, **Single Bus Organization and Micro Instructions**		
Mapping of Course Outcomes for Unit VI	**CO6**	
UNIT VI	**Memory and I/O System**	**08 Hours**
Memory Systems: Characteristics of Memory Systems, Memory Hierarchy, Memory read & write cycle, Semiconductor RAM Memories, Read-only Memories **Cache Memory** – Principle of Locality, Organization, Mapping functions, write policies, Replacement policies, Multilevel Caches, Cache Coherence **Input / Output Systems**: Accessing I/O devices, I/O Module-Programmed I/O, Interrupt Driven I/O, Direct Memory Access (DMA).		
Books and Other Resources		
Text Books: 1. "Modern Digital Electronics", R.P. Jain, Tata McGraw-Hill, Third Edition 2. "Computer organization and architecture, designing for performance" by William Stallings , Prentice Hall ,Eighth edition **Reference Books:** 1. "Digital Design", M Morris Mano, Prentice Hall, Third Edition 2. "Computer organization" , Hamacher and Zaky, Fifth Edition		
Guidelines for Students Journal : The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment		

grade/marks and assessor's sign, Theory- Concept, circuit diagram, pin configuration, conclusion/analysis). As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided.

Guidelines for Laboratory /TW Assessment :

- Continuous assessment of laboratory work is done based on overall performance and Laboratory performance of student.
- Each Laboratory assignment assessment should assign grade/marks based on parameters with appropriate weightage.
- Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students.

Practical Assignments

1. Design and Implement Full Adder using IC74138
2. Design and implement Code Converters-Binary to Gray.
3. Design and Realization of BCD Adder using 4-bit Binary Adder (IC 7483).
4. Design & Implement Parity Generator and checker using EX-OR.
5. Design of Synchronous 3 bit Up and Down Counter using MSJK Flip Flop / D Flip Flop
6. Design and Realization: Flip Flop conversion
7. Write a C++/Python/Java Program to implement booth algorithm.
8. Case Study: Single Bus Organization and Microinstructions

CO23204: Object Oriented Paradigm

TeachingScheme:	Credit:03	Examination Scheme:
TH:03 Hrs/Week		Course Activity : 20Mark Mark In Semester:20Mark End Semester : 70Mark

Prerequisite: Object oriented programming (OOP)

Course Objective:

- To learn the basic concept of Java Programming.
- To learn Object Oriented Programming(OOP) principles using Java Programming Language
- To learn Exception handling concepts of Java Programming
- To learn Applet and Multithreading concepts of Java Programming
- To learn AWT concepts of Java Programming

Course Outcomes:

Students will be able to

1. Develop programs using Java, an Object Oriented Programming language.
2. Develop application using inheritance, encapsulation, and polymorphism.
3. Demonstrate Packages and Interfaces
4. Demonstrate Exception handling and file operations
5. Develop application using Multithreading for robust application development.
6. Develop application using Applet and AWT.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Project using Java Language
2. Industry Visit
3. Seminar

CourseContents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Introduction to Java**	**06 Hours**

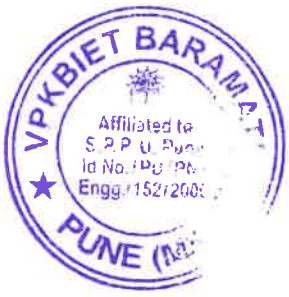
Features of Java ,JDK Environment ,OOPs Concepts Class, Abstraction , Encapsulation, Inheritance, Polymorphism, Difference between C++ and JAVA, Structure of Java program ,Data types ,Variables ,Operators , Keywords , Decision Making (if, switch), Looping(for, while) ,Type Casting ,Array Creating an array Types of Array - One Dimensional arrays - Two Dimensional array ,String - Arrays , Methods.

Mapping of Course Outcomes for Unit II	CO2	
UNIT II	**Classes, Object and Inheritance**	**06 Hours**

Creating Classes and objects , Memory allocation for objects , Constructor , Implementation of Inheritance Simple, Multilevel, Using super: Using super to Call Superclass Constructors, , Implementation of Polymorphism Method Overloading, Method Overriding ,Modifiers and Access Control. Using final with Inheritance: Using final to Prevent Overriding , Using final to Prevent Inheritance

Mapping of Course Outcomes for Unit III	CO3	
UNIT III	**Packages and Interfaces**	**06 Hours**

Packages : Defining a Package , Finding Packages and CLASSPATH, A Short Package Example Packages and Member Access : Example, Importing Packages, **Interfaces :** Defining an Interface, Implementing Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces, Interfaces Can Be Extended **Default Interface Methods:** Default Method Fundamentals, Example.		
Mapping of Course Outcomes for Unit IV	**CO4**	
UNIT IV	**Exception and File Handling**	**06 Hours**
Exception: Exception types, Using try catch and multiple catch, Nested try, throw , throws and finally, Creating user defined Exceptions **File Handling :**Stream, Byte Stream Classes, Character Stream Classes, File IO basics, File operations ,Creating file, Reading file (character, byte) ,Writing file (character, byte).		
Mapping of Course Outcomes for Unit V	**CO5**	
UNIT V	**Multithreading in Java**	**06 Hours**
Concurrency and Synchronization, Java Thread Model: Thread priorities, Synchronization, Messaging, Main Thread, Creating thread: Implementing Thread using thread class and Runnable interface. Creating multiple threads using is Alive() and join().		
Mapping of Course Outcomes for Unit VI	**CO6**	
UNIT VI	**Applet and AWT Programming**	**06 Hours**
Applet : Introduction, Types applet, Applet Life cycle, Creating applet, Applet tag, Applet Classes, Color, Graphics , Font. **AWT :** Components and container used in AWT, Layout managers, Listeners and Adapter classes, Event Delegation model		
Books and Other Resources		
TextBooks: 1. Programming with JAVA - E Balgurusamy 2. Herbert Schildt, "The Complete Reference Java", 9th Ed, TMH,ISBN: 978-0-07-180856-9. **ReferenceBooks:** 1. The Complete Reference – JAVA Herbert Schildt 2. "Maureen Spankle, "Problem Solving and Programming Concepts", Pearson; 9th edition, ISBN-10: 9780132492645, ISBN-13: 978- 0132492645		



MDCO23201: Computer Graphics and Gaming

Teaching Scheme:	Credit: 03	Examination Scheme:
TH:02Hrs/Week PR:02Hrs/Week	TH Credit :02 PR Credit :01	Course Activity :20Marks In Semester :20Marks End Semester :50Marks Term-Work :20Marks

Prerequisite: C++ programming (CPP)

Companion Course, if any: Computer Graphics Laboratory

Course Objective:

1. Remembering: To acquaint the learner with the basic concepts of Computer Graphics
2. Understanding: To learn the various algorithms for generating and rendering graphical figures.
3. Applying: To get familiar with mathematics behind the graphical transformations
4. Understanding: To understand and apply various methods and techniques regarding animation.
5. Creating: To generate Interactive graphics using OpenGL

Course Outcomes:

Students will be able to

1. Be familiar with the graphics designing concepts and devices.
2. Construct a mathematical design using the development process.
3. Recognize the design principles of animation and gaming application.
4. Implement the use of gaming tools in application design.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Active participation in Gaming Competition
2. Poster Presentation
3. Video Presentations
4. Survey on various Animation making tools
5. Visit to Animation Business Schools

Course Contents

Mapping of Course Outcomes for Unit I		CO1
UNIT I	**Basics of Computer Graphics**	**03 Hours**

Introduction, What is computer Graphics? Area of Computer Graphics, Design and Drawing, Animation Multimedia applications, Simulation, How are pictures actually stored and displayed, Difficulties for displaying pictures. Graphics Devices Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, The Shadow - Mask CRT, Direct View Storage Tube, Tablets, The light Pen, Three Dimensional Devices.

Mapping of Course Outcomes for Unit II		CO2
UNIT II	**Transformations**	**03 Hours**

VPKBIET BARAMATI Affiliated to S.P.P.U; Pune Id No./PU/PN/Engg./152/2000 PUNE (MS.)

Simple line drawing methods, Introduction Point Plotting Techniques Qualities of good line drawing algorithms The Digital Differential Analyzer (DDA), Bresenham's Algorithm Generation of Circles Introduction, what is transformation? Matrix representation of points Basic transformation, Translation, Rotation, Scaling, Need for 3-Dimensional Imaging Techniques for 3-Dimesional displaying, Translation, Rotation, Scaling

Mapping of Course Outcomes for Unit III	**CO3**

UNIT III	**Animation**	**03 Hours**

Animation: Introduction, Conventional and computer-based animation, Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. Design of animation sequences, Animation languages, Key- frame, Morphing, Motion specification. Gaming: Introduction, Gaming platform (NVIDIA, i8060), Advances in Gaming.

Mapping of Course Outcomes for Unit IV	**CO4**

UNIT IV	**Gaming**	**03Hours**

Principles of game design, Game Design Theory,MDA,8 type of Fun in Game, Visual style, Gameplay, Generate ideas for a game concept Idea Development Process, Stimulus, Genre Market Research, Target platform ,Creating Prototype Creating physical Games: Board Game, Card Game, Party Games and etc....

Books and Other Resources

TextBooks:

1. Computer Graphics, Multimedia and Animation ,2010, Pakhira Malay K.
2. Donald D. Hearn and Baker- Computer Graphics with OpenGL, 4th Edition, ISBN-13: 9780136053583

Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, —Computer Graphics Principles and Practice‖, 2nd Edition,Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Rogers, J. Adams, —Mathematical Elements for Computer Graphics‖, 2nd Edition, Tata McGrawHill Publication, 2002, ISBN 0 – 07 – 048677 – 8.

Guidelines for Term Work Assessment :

Termwork assessment will be based on overall performance of Laboratory assignments performed by a students.

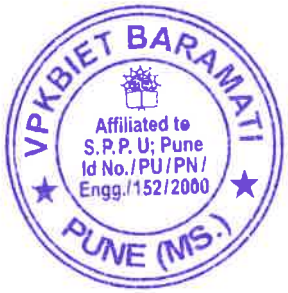
Guideline for Practical Conduction :

Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.
Operating System recommended :- 64-bit Open source Linux or its derivative, Windows
Programming tools recommended: - Open Source C++ Programming tool like G++/GCC, OPENGL, DEV C++.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation.

2.	Write C++ program to draw a 4X4 chessboard.
3.	Write C++ program to draw 2-D object and perform following basic transformations, a) Scaling b) Translation c) Rotation.
4.	Write C++ program to draw Man Walking in the Rain with an Umbrella.
5.	Write a C++ Program to make puzzle game.
6.	Write a C++ Program to make Tic Tac Toe game.
7.	Write a C++ Program to draw a car in motion.



CO23205: Data Analytics and Visualization using Python		
TeachingScheme: PR:02Hrs/Week	**Credit: 02**	**Examination Scheme:** Term-Work: 40 Marks Practical : 20 Marks

Prerequisite: Programming and Problem Solving (PPS)

Course Objective:

- To understand principles of Data Analytics and Visualization for the analysis of real time problems
- To develop in depth understanding and implementation of the key technologies in Data Science and data analytics To understand the Processor and Instructions
- To develop ability to analyse and demonstrate knowledge of statistical data analysis techniques for decision-making
- To gain practical, hands-on experience with statistics programming languages.

Course Outcomes:

On completion of this course students will be able to

1. Apply principles of Data Science for the analysis of real time problems
2. Implement data representation using statistical methods
3. Use various data analytics libraries
4. Perform pre-processing on the dataset
5. Implement data visualization techniques

Guidelines for Term Work Assessment :

Term work assessment will be based on overall performance of Laboratory assignments performed by a students.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. .

Guidelines for Laboratory Conduction :

Operating System recommended :- 64-bit Open source Linux or its derivative
Programming tools recommended: - Python

Practical Assignments

1. Data Wrangling I

Perform the following operations using Python on any open source dataset (e.g., data.csv)

- Import all the required Python Libraries.
- Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a cleardescription of the data and its source (i.e., URL of the web site).
- Load the Dataset into pandas dataframe.
- Data Preprocessing: check for missing values in the data using pandas isnull(), describe()function to get some initial statistics. Provide variable descriptions. Types of variables etc.Check the dimensions of the data frame.

- Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
- Turn categorical variables into quantitative variables in Python.

In addition to the codes and outputs, explain every operation that you do in the above steps andexplain everything that you do to import/read/scrape the data set.

2. Data Wrangling II

Create an "Academic performance" dataset of students and perform the following operations usingPython.

- Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.
- Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.
- Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.

3. Data Wrangling III

Perform the following statistic operations on any open source dataset (e.g., data.csv)

- Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset(age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable
- Scan Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor'of iris.csv dataset.

Provide the codes with outputs and explain everything that you do in this step

Use dataset https://www.kaggle.com/c/boston-housing for 4-6 assignments

4. Write a Python program to demonstrate how to draw a Bar Plot using Matplotlib/Seaborn and analyse the data patterns if any.

5. Write a Python program to demonstrate how to draw a Scatter Plot using Matplotlib/Seaboarn and analyse the data patterns if any.

6. Write a Python program to demonstrate how to draw a Histogram Plot using Matplotlib/Seaboarn and analyse the data patterns if any.

7. Write a Python program to demonstrate how to draw a Pie Chart using Matplotlib/Seaboarn Using iris.csv dataset and analyse the data patterns if any.

8. Download the Iris flower dataset into a DataFrame.(e.g.,https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as:
   - List down the features and their types (e.g., numeric, nominal) available in the dataset.
   - Create a histogram for each feature in the dataset to illustrate the feature distributions.
   - Create a boxplot for each feature in the dataset.
   - Compare distributions and identify outliers.

9. Use the dataset 'titanic',
   - Plot a box plot for distribution of age with respect to each gender along with the information about whether they survivedor not. (Column names : 'sex' and 'age')
   - Write observations on the inference from the above statistics.

10. Write a Python program to draw 3D Plots using Plotly Libraries.

11. Write a Python program to draw Time Series and creating map using Plotly Libraries.

12. **Mini Project:**

Mini project aims at giving students a hands on experience of data analytics and visualization.

- Collect the dataset
- Pre-process the data
- Analyze the data.
- Visualize the dataset using above visualization methods and tools.

Reference Books :

1. Chirag Shah, "A Hands-On Introduction To Data Science", Cambridge University Press,(2020), ISBN : ISBN 978-1-108-47244-9.
2. Wes McKinney, "Python for Data Analysis", O' Reilly media, ISBN : 978-1-449-31979-3.
3. "Scikit-learn Cookbook", Trent hauk,Packt Publishing, ISBN: 9781787286382
4. R Kent Dybvig, "The Scheme Programming Language", MIT Press, ISBN 978-0-262-51298-5.
5. Jenny Kim, Benjamin Bengfort, "Data Analytics with Hadoop", OReilly Media, Inc.
6. Jake VanderPlas, "Python Data Science Handbook" https://tanthiamhuat.files.wordpress.com/2018/04/pythondatasciencehandbook.pdf
7. Gareth James, "An Introduction to Statistical Learning" https://www.ime.unicamp.br/~dias/Intoduction%20to%20Statistical%20Learning.pdf
8. Cay S Horstmann, "Scala for the Impatient", Pearson, ISBN: 978-81-317-9605-4, Alvin Alexander, "Scala Cookbook", O'Reilly, SPD,ISBN: 978-93-5110-263-2

References :

- https://www.simplilearn.com/data-science-vs-big-data-vs-data-analytics-article
- https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
- https://www.edureka.co/blog/hadoop-ecosystem

- https://www.edureka.co/blog/mapreduce-tutorial/#mapreduce_word_count_example
- https://github.com/vasanth-mahendran/weather-data-hadoop
- https://spark.apache.org/docs/latest/quick-start.html#more-on-dataset-operations

https://www.scala-lang.org/

MOOCs Courses link:

- https://nptel.ac.in/courses/106/106/106106212/
- https://onlinecourses.nptel.ac.in/noc21_cs33/preview
- https://nptel.ac.in/courses/106/104/106104189/

https://onlinecourses.nptel.ac.in/noc20_cs92/preview

Virtual Laboratory:

- "Welcome to Virtual Labs - A MHRD Govt of india Initiative"
- http://cse20-iiith.vlabs.ac.in/List%20of%20Experiments.html?domain=Computer%20Science

CO23206: Data Structures Lab		
Teaching Scheme: PR: 04 Hrs/Week	**Credit: 02**	**Examination Scheme:** Term Work: 40 Mark Practical: 40 Mark

Guidelines for Practical Examination:

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student 's understanding of the fundamentals, effective and efficient implementation.

Guidelines for Laboratory Conduction:

Assignments should be implemented in C++/JAVA programming language. Use of open source software is encouraged. Based on the concepts learned. Operating System recommended: - 64-bit Open source Linux or its derivative Programming Programming tools recommended: Open source programming tool like G++/GCC, Jupyter Notebook, Pycharm, Spyder etc.

Guidelines for Term Work Assessment:

Term work assessment will be based on overall performance of Laboratory assignments performed by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, efficient codes, and punctuality.

Practical Assignments

1. Implement a program to store marks of N students in Fundamental of Data Structures and calculate: avg. score of the class, highest and lowest score, count absent students, display marks with highest frequency
2. Implementation of following matrix operations: addition of two matrices, subtraction of two matrices, multiplication of two matrices, transpose of a matrix
3. Implement a program to store roll numbers of student in array who attended training program in random order. Write function for- a) Searching whether particular student attended training program or not using linear search b) Searching whether particular student attended training program or not using binary search
4. Implement a program to store percentage of students using an array. Write function for sorting array of in ascending order using a) Selection Sort b) Bubble sort
5. A Implement a program for expression conversion as infix to postfix
6. Implement a program for postfix/prefix expression evaluation
7. Implement a program to simulate the system using circular queue using array.
8. Implement a program for manipulation of a singly link list(insert at front, insert at middle insert at end, delete at front, delete at middle delete at end,display,reverse)
9. Starting with an Empty Binary Search Tree(BST),create a BST by reading the values in the given Order and perform following operations on it: 1. Insert a new node 2. Perform Inorder, Preorder And Postorder Traversals 3. Search 4. Delete 5. Height 6. Mirror Image 7. Find Smallest and Largest Element
10. Implement a program to create Threaded Binary Tree and perform inorder traversal on it.
11. Implement a program to represent a graph using adjacency matrix and adjacency list, compute

DFS and BFS of the same

12. Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number.
13. Implement a program to find minimum spanning tree using prims/kruskal's algorithm
14. A Implement a program to find shortest path using Dijkstra's algorithm

Books and Other Resources

Text Books:

1. "Fundamentals of Data Structures in C", E. Horowitz, S. Sahni, S. Anderson-freed, Second Edition, 2008, University Press, ISBN 978-81-7371-605-8

2. "The C Programming Language", B. Kernighan, D. Ritchie, Second Edition, 2015, Pearson Education India; ISBN 81-203-0596-5

Reference Books:

1. "Data Structures using C", Y. Langsam, M. Augenstin and A. Tannenbaum, First Edition, 2002, Pearson Education Asia, ISBN 978-81-317-0229-1
   "Computer organization" , Hamacher and Zaky, Fifth Edition
2. "Fundamentals of Data Structures in C++", Ellis Horowitz, S. Sahni, D. Mehta, 2nd Edition, 2008, University Press, ISBN-10: 8173716064
3. "An introduction to data structures with Applications", Jean-Paul Tremblay, Paul. G. Soresan, 2nd Edition, 1984, Tata Mc-Graw Hill International Editions, ISBN-0-07-462471-7

GS23201: Advanced Mathematics for Computer Engineering

Teaching Scheme:	Credit: 04	Examination Scheme:
Theory : 3 Hours/Week Tutorial :1 Hour/Week		In-Semester : 20 Marks End-Semester :70 Marks Term Work : 20 Marks

Prerequisite: Differential & Integral calculus, Taylor series, Differential equations of first order and first degree, Fourier series, Collection, Classification & Representation of data.

Course Objective:

To provide the students with concepts and techniques in Linear differential equations, Fourier transform, Statistical methods, and Probability theory. The aim is to equip them with the techniques to understand advanced-level mathematics and its applications that would be useful in their discipline and enhance their thinking power.

Course Outcomes:

1. Solve higher-order linear differential equations using appropriate techniques useful for modeling in their field.
2. Understand the concepts of Fourier transform
3. Understand and apply the various concepts of statistical methods of correlation, and regression and Apply them in their field.
4. Apply the concepts of appropriate Probability and Probability distribution for data analysis and predictions in multiple data sets.
5. Solve Algebraic, Transcendental equations and System of linear equations using numerical techniques.
6. Compute Interpolating polynomials, numerical differentiation, and integration, numerical solutions of ordinary differential equations used in modern scientific computing.

Course Contents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Linear Differential Equations (LDE) and Applications**	**07 Hours**

Introduction, Solution of LDE, General method, short-cut method, Method of variation of parameters, Cauchy's, Legendre's DE, Simultaneous DE.

Mapping of Course Outcomes for Unit II	CO2	
UNIT II	**Fourier Transform and Statistics**	**07 Hours**

Fourier Transform: General Fourier, Fourier Sine, Cosine, and inverse transforms.

Statistics: Measures of dispersion, Moments, Skewness and Kurtosis, Correlation and Regression analysis.

Mapping of Course Outcomes for Unit III	CO3	
UNIT III	**Regression Models**	**07 Hours**

Importance of Regression in Data Mining, Simple Linear Regression, Model: $Y = \beta0 + \beta1X + \varepsilon$ Assumptions, Estimation of $\beta0$ and $\beta1$ by the method of least squares, Multiple linear regression model $Y = \beta0 + \beta1X1 + ... + \beta pXp + \varepsilon$, residuals, Least-Squares Estimation of the Regression Coefficients, obtaining normal equations, Solutions of normal equations, Generalized linear models, and applications.

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Probability and Probability Distributions	07Hours
Theorems on probability, Random variables, Probability Mass function, Probability Density function, Mathematical Expectation. Binomial, Poisson, and Normal distribution and applications.		
Mapping of Course Outcomes for Unit V		**CO5**
UNIT V	**Numerical methods for Algebraic and System of Equations**	**07 Hours**
Numerical Solution of Algebraic and Transcendental Equations: Bisection, Secart, Regula-Falsi, Newton–Raphson and Successive Approximation Methods. Numerical Solutions of System of linear equations: Gauss elimination, LU Decomposition, Cholesky, Jacobi and Gauss-Seidel Methods		
Mapping of Course Outcomes for Unit VI		**CO6**
UNIT VI	**Numerical methods in calculus**	**07 Hours**
Interpolation: Finite Differences, Newton's and Lagrange's Interpolation formula, Numerical Differentiation. Numerical Integration: Trapezoidal and Simpson's rules. Solution of Ordinary differential equations: Euler's, Modified Euler's, Runge-Kutta 4th order.		

Books and Other Resources

Text Books:

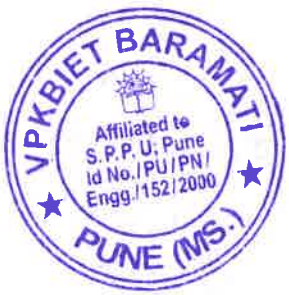
1. Higher Engineering Mathematics by B.V. Ramana (Tata McGraw-Hill).
2. Higher Engineering Mathematics by B. S. Grewal (Khanna Publication, Delhi).

E Books & Videos:

1. Erwin Kreyszig, "Advanced Engineering Mathematics", 10ed, Wiley India
2. M. D. Greenberg, "Advanced Engineering Mathematics", 2nd e Pearson Education
3. Peter V. O'Neil, "Advanced Engineering Mathematics", 7ed, Cengage Learning
4. S. L. Ross, "Differential Equations", 3e, Wiley India
5. Sheldon M. Ross, "Introduction to Probability and Statistics for Engineers and Scientists", 5e, Elsevier Academic Press
6. M. K. Jain, S. R. K. Iyengar, and R. K. Jain, "Numerical Methods for Scientific and Engineering Computation", 5e, (New Age International Publication).
7. Draper, N. R. and Smith, H. "Applied Regression analysis", (1998) (John Wiley) Third Edition.
8. S.P. Gupta, Sultan Chand and Sons, "Statistical Methods", New Delhi, 2009.

Guidelines for Term Work Assessment :

1. Tutorials for the subject shall be engaged in a minimum of three batches (batch size of 22 students) per division.
2. Term work shall consist of six assignments on each unit and is based on performance and continuous internal assessment.



CO23211: Microprocessor

Teaching Scheme:	Credit: 04	Examination Scheme:
TH: 03 Hrs/Week PR: 02 Hrs/Week	TH Credit :03 PR Credit : 01	Course Activity:20 Marks In Semester :20 Marks End Semester :70 Marks Termwork :20 Marks Oral : 20 Marks

Prerequisite:
Digital Electronics and Logic Design

Course Objectives:
The course is intended to provide practical exposure to the students on microprocessors, design and coding knowledge on 8086 and 80836 and introduction to microcontrollers.

- To learn and distinguish the architecture and programmer's model of advanced processor.
- To acquire the logic to build assembly language programs.
- To identify the memory management features and processes of advanced processors.
- To study processor modes and protection methods.
- To demonstrate the use of virtual mode.
- To differentiate Microprocessor and microcontroller.

Course Outcomes:
After successful completion of the course, the learner will be able to-

1. **Exhibit** skill of assembly language programming for the application.
2. **Classify** Processor architectures and bus cycles.
3. **Illustrate** advanced features of 80386 Microprocessor.
4. **Compare** and **contrast** different processor modes and protection methods.
5. **Use** virtual mode mechanism in applications.
6. **Differentiate** between Microprocessors and Microcontrollers.

Guideline for Couse Activity:
The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Active participation in Hackthon related to Microprocessor Assembly Language
2. Survey on uses of Microprocessor with emerging technology
3. Industry Visit
4. Seminar
5. Research Paper
6. Intellectual Property Rights

Mapping of Course Outcomes for Unit I		CO1
UNIT I	Introduction to Microprocessor and Assembly Language	08 Hours

Introduction: Brief History of Intel Processors, Basics of 8086, Need of microprocessors, Applications of microprocessor, Addressing modes and data types.

Instruction Set: 80386 Instruction Set, types of instructions of 80386

Introduction to assembly language programming- Basic Arithmetic and Loop Instructions, Basic Syntax, Procedures and Parameters, Macro, String Representations and Array Representations and Processing.

Mapping of Course Outcomes for Unit II	**CO1, CO2**

UNIT II	**System Architecture**	**08 Hours**

Systems Architecture- 80386 Architecture, Programmers Model, Operating modes, Systems Registers (Systems flags, Memory Management registers, Control registers, Debug registers, Test registers), System Instructions. Functional pin Diagram, functionality of various pins, I/O Organization, Memory Organization (Memory banks), Basic memory read and writes cycles with timing diagram, Processor State after Reset.

Mapping of Course Outcomes for Unit III	**CO3**

UNIT III	**Memory Management**	**06Hours**

Global Descriptor Table, Local Descriptor Table, Interrupt Descriptor Table, GDTR, LDTR, IDTR. Formats of Descriptors and Selector, Segment Translation, Page Translation, Combining Segment and Page Translation.

Mapping of Course Outcomes for Unit IV	**CO4**

UNIT IV	**Protection**	**08 Hours**

Need of Protection, Overview of 80386DX Protection Mechanisms: Protection rings and levels, Privileged Instructions, Concept of DPL, CPL, RPL, EPL.Inter privilege level transfers using Call gates, Conforming code segment, Privilege levels and stacks. Page Level Protection, Combining Segment and Page Level Protection.

Mapping of Course Outcomes for Unit V	**CO5**

UNIT V	**Multitasking and Virtual 8086 Mode**	**08 Hours**

Multitasking- Task State Segment, TSS Descriptor, Task Register, Task Gate Descriptor, Task Switching, Task Linking, Task Address Space.

Interrupts, Exceptions: Error handling in microprocessor

Virtual Mode – Features, Memory management in Virtual Mode, Entering and leaving Virtual mode.

Mapping of Course Outcomes for Unit VI	**CO6**

UNIT VI	**Introduction to Microcontrollers**	**06 Hours**

Introduction to Microcontrollers: Architecture of typical Microcontroller, In diagram of microcontroller, Difference between Microprocessor and Microcontroller, Characteristics of microcontrollers, Types of microcontroller, Applications of Microcontrollers.

Books and Other Resources

Textbooks:

1. A.Ray, K.Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing", Tata McGraw Hill,2004 ISBN 0-07-463841-6
2. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, Revised 2 Edition, 2006 ISBN 0- 07-

100462-9

Reference Books:

1. Walter A. Triebel, "The 80386Dx Microprocessor: Hardware", Software, and Interfacing, Pearson Education, ISBN: 0137877307, 9780137877300.
2. Brey, Barry B, "8086/8088, 80286, 80386 and 80486 Assembly Language Programming", Prentice Hall, ISBN: 13: 9780023142475.
3. Mohammad Rafiquzzaman, "Microprocessors: Theory and Applications: Intel and Motorola", Prentice Hall, ISBN:-10:0966498011, 13:978:0966498011.
4. Introduction to 64 bit Intel Assembly Language Programming for Linux, 2nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012.Assembly Language Step-by-step: Programming with Linux, 3rd Edition, Jeff Duntemann, Wiley ISBN:-10 0470497025, ISBN-13: 978-0470497029, 2009.

Virtual Laboratory:

http://209.211.220.205/vlabiitece/mi/MI3.php

MOOC/ Video Lectures available at:

https://nptel.ac.in/courses/106/108/106108100/

https://nptel.ac.in/courses/108/107/108107029/

Guideline for Practical Conduction:

Continuous assessment of laboratory work is based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

Operating System: 64-bit Open source Linux or its derivative.

Programming Tools: Preferably using Linux equivalent or MASM/TASM/NASM/FASM.

Guidelines for Term Work Assessment :

Term work assessment will be based on overall performance of Laboratory assignments performed by a students.

Guidelines for Practical Examination :

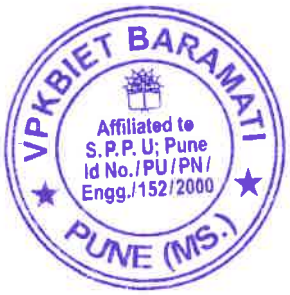
Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation

Practical Assignments

1. Write an X86/64 ALP to accept a string and to display its length.
2. Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers
3. Write an X86/64 ALP to count the number of positive and negative numbers from the array.
4. Write an X86/64 ALP to multiply two 8 and 16 bit signed and unsigned numbers.
5. Write an X86/64 ALP to compare two strings using string instructions.
6. Write x86 ALP to find the factorial of a given integer number on a command line by using recursion.
7. Write x86 ALP using macro for a) Addition b) Subtraction c) Multiplication

d) Division

8. Write an X86 ALP to accept five Hexadecimal numbers from user and store them in an array and display them.
9. Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5- digit BCD number into its equivalent HEX number.
10. Write X86 ALP to find, a) Number of Blank spaces b) Number of lines c) Occurrence of a particular character. Accept the data from the text file.



CO23212: Database Management Systems

Teaching Scheme:	Credit: 04	Examination Scheme:
TH: 03 Hrs/Week	TH Credit :03	Course Activity: 20 Mark
PR: 02 Hrs/Week	PR Credit :01	In Semester: 20 Mark
		End Semester: 70 Mark
		Term Work : 20 Mark
		Practical: 20 Mark

Prerequisite: Students are expected to have a good understanding of Discrete Mathematics, Data Structures and Algorithms

Course Objective:

- To understand the fundamental concepts of Database Management Systems
- To acquire the knowledge of database query languages and transaction processing
- To understand systematic database design approaches
- To acquire the skills to use a powerful, flexible, and scalable general-purpose databases to handle Big Data
- To be familiar with advances in databases and applications

Course Outcomes:

1. Analyze and design Database Management System using ER model
2. Implement database queries using database languages
3. Normalize the database design using normal forms
4. Apply Transaction Management concepts in real-time situations
5. Use NoSQL databases for processing unstructured data
6. Use advanced database Programming concepts

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Database Mini Project
2. Survey on uses of Advanced Database with emerging technology presentation
3. Industry Visit
4. Seminar
5. Research Paper in database domain

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction to Database Management Systems and ER Model	07 Hours

Introduction, Purpose of Database Systems, Database-System Applications, View of Data, Database Languages, Database System Structure, Data Models. **Database Design and Model**: Entity, Attributes, Relationships, Constraints, Keys, Design Process, Entity-Relationship Model, ER Diagram, Design Issues, Extended E-R Features, converting ER and EER diagram into tables.

Mapping of Course Outcomes for Unit II	CO2

UNIT II	SQL and PL/SQL	07 Hours

SQL: Characteristics and Advantages, SQL Data Types and Literals, DDL, DML, DCL, TCL, SQL Operators. **Tables**: Creating, Modifying, Deleting, Updating. **SQL DML Queries**: SELECT Query and clauses, Index and Sequence in SQL. **Views**: Creating, Dropping, Updating using Indexes, Set Operations, Predicates and Joins, Set membership, Tuple Variables, Set comparison, Ordering of Tuples, Aggregate Functions, SQL Functions, Nested Queries. **PL/SQL**: Concept of Stored Procedures and Functions, Cursors, Triggers, Assertions, Roles and Privileges.

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Relational Database Design	07 Hours

Relational Model: Basic concepts, Attributes and Domains, CODD's Rules. **Relational Integrity:** Domain, Referential Integrities, Enterprise Constraints. **Database Design:** Features of Good Relational Designs, Normalization, Atomic Domains and First Normal Form, Decomposition using Functional Dependencies, Algorithms for Decomposition, 2NF, 3NF, BCNF.

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Database Transaction Management	07Hours

Introduction to Database Transaction, Transaction states, ACID properties, Concept of Schedule, Serial Schedule. **Serializability:** Conflict and View, Cascaded Aborts, Recoverable and Non- recoverable Schedules. **Concurrency Control**: Lock-based, Time-stamp based Deadlock handling. **Recovery methods**: Shadow-Paging and Log-Based Recovery, Checkpoints. **Log-Based Recovery**: Deferred Database Modifications and Immediate Database Modifications.

Mapping of Course Outcomes for Unit V	CO5

UNIT V	NoSQL Databases	07 Hours

Introduction to Distributed Database System, Advantages, disadvantages, CAP Theorem. **Types of Data:** Structured, Unstructured Data and Semi-Structured Data. **NoSQL Database:** Introduction, Need, Features.
Types of NoSQL Databases: Key-value store, document store, graph, wide column stores, BASE Properties, Data Consistency model, ACID Vs BASE, Comparative study of RDBMS and NoSQL. **Mongo DB** (with syntax and usage): CRUD Operations, Indexing, Aggregation, MapReduce, Replication, Sharding.

Mapping of Course Outcomes for Unit VI	CO6

UNIT VI	Advances in Databases	07 Hours

Emerging Databases: Active and Deductive Databases, Semantic Databases. **Complex Data Types:** Semi-Structured Data **Nested Data Types**: JSON, XML. **Spatial Data:** Geographic Data, Geometric Data. Introduction to Big Data, **HADOOP:** HDFS, Map Reduce.

Books and Other Resources

Text Books:

1. "Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", McGraw Hill Publishers, ISBN 0-07-120413-X, 6th edition
2. "Connally T, Begg C., "Database Systems", Pearson Education, ISBN 81-7808-861-4
3. "Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled, Addison Wesley", ISBN- 10: 0321826620, ISBN-13: 978-0321826626

Reference Books:

1. "C J Date, "An Introduction to Database Systems", Addison-Wesley, ISBN: 0201144719
2. "S.K.Singh, "Database Systems: Concepts, Design and Application", Pearson Education, ISBN 978-81-317-6092-5
3. "Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O'Reilly Publications, ISBN: 978-

1 449 34468 9

4. "Kevin Roebuck, "Storing and Managing Big Data - NoSQL, HADOOP and More", Emereopty Limited, ISBN: 1743045743, 9781743045749

5. "Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications ISBN: 9788176569644, 9788176569644

Guidelines for Term Work Assessment :

Term work assessment will be based on overall performance of Laboratory assignments performed by a students. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, efficient codes, and punctuality.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation..

Guidelines for Laboratory Conduction :

Use of open source software is encouraged. Based on the concepts learned.
Operating System recommended :- 64-bit Open source Linux or its derivative Programming
Tools recommended: - MYSQL/Oracle, MongoDB, ERD plus, ER Win

Practical Assignments

1. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym. Design and implement Code Converters-Binary to Gray.
2. Write at least 10 SQL queries for suitable database application using SQL DML statements.
3. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 1 to 10. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns, radius and area using procedure and function.
4. Write a PL/SQL block of code to calculate grades of students and separate all students grades wise using Cursor.
5. Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library Audit table
6. Write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit and Display etc.)
7. Design and Develop Mongo DB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.).
8. Design and Develop Mongo DB Queries using aggregation and indexing with suitable example using Mongo DB.
9. Implement Map reduces operation with suitable example using Mongo DB.
10. Write a program to implement Mongo DB database connectivity with any front end language to implement Database navigation operations (add, delete, edit, Display)

CO23213: Operating Systems		
Teaching Scheme: TH: 03 Hrs/Week PR: 02 Hrs/Week	**Credit: 04** TH Credit :03 PR Credit :01	**Examination Scheme:** Course Activity: 20 Mark In-Semester: 20 Mark End-Semester: 70 Mark Term work: 20 Mark Oral: 20 Mark

Prerequisite: Digital Electronics and Computer Organization, Programming Languages

Course Objective:

- To learn the basic concepts operating system
- To learn processes life cycle, process states and scheduling algorithms
- To learn Inter Process synchronization, and communication between processes
- To explore memory management, and virtual memory management policies..
- To understand the organization and management of file systems,
- To understand device management

Course Outcomes:

1. To Build the basic knowledge of Operating System
2. To identify process management strategies and processor scheduling algorithms
3. To apply concepts of Inter Process synchronization, and communication
4. To apply memory management, and virtual memory management strategies
5. To make use of concepts of File management and free space management
6. To make use of concepts of device management

Course Activity:

Course teacher will plan the course activity

Course Contents

Mapping of Course Outcomes for Unit I		CO1
UNIT I	**Operating System Structures**	**06 Hours**

Introduction to Operating System: Evolution of OS, Functions of OS, Operating System Components, O.S. Services, Types of OS, Kernel and types of Kernel, System Calls, Virtual Machines, Boot Sequence. Case study of UNIX operating System

Mapping of Course Outcomes for Unit II		CO2
UNIT II	**Processes Management**	**08 Hours**

Process concept: Creation, Termination of process, Process states, Context Switching, Process Control Block, **Thread:** Concept of a Thread, Thread libraries, Multithreading, Comparison of thread

Process Scheduling, Scheduling criterion, Scheduling algorithms: FCFS, SJF, RR, Priority ,Process System calls. Case Study of Unix Process Management

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Inter process Communication & Synchronization	08 Hours
Synchronization: Critical section problem, Hardware support for mutual exclusion, Semaphores, Monitors, Classical Problems in Synchronization: Producer-consumer, Reader-writer, **Deadlock:** Deadlock-principle, Deadlock prevention Deadlock avoidance, Deadlock detection and recovery, Case Study of Unix IPC		
Mapping of Course Outcomes for Unit IV	**CO4**	
UNIT IV	**Memory management**	**08 Hours**
Memory Management: Continuous and Non Contiguous memory management, Swapping, Paging, Segmentation. **Virtual Memory Management**: Demand Paging, Page replacement algorithms- FIFO, LRU, Optimal, Thrashing, Allocation method, Case Study of Unix Memory management		
Mapping of Course Outcomes for Unit V	**CO5**	
UNIT V	**File Management**	**06 Hours**
File Organization: Concept of files, File Attributes, File operations, File types, Directories and types of directories, Free space management. **File System Implementation:** Data structures like Inode and super block, data block and boot block. Case study of Unix file system		
Mapping of Course Outcomes for Unit VI	**CO6**	
UNIT VI	**I/O System**	**08 Hours**
I/O Management: I/O devices, Organization of I/O functions, Design issues related to I/O devices, I/O Buffering. **Disk Scheduling**- FCFS, SCAN, SSTF, LOOK		

Books and Other Resources

Text Books:

1. " Abranhan Silberschatz, Peter B Galvin, Greg Gagne; Operating System Concepts, Wiley India Students Edition, 8th Edition, ISBN: 978-81-265-2051-0
2. Andrew S. Tanenbaum; Modern Operating Systems; Prentice Hall of India Publication; 3rd Edition. ISBN: 978-81-203-3904-0

Reference Books:

1. Milan Milenkovic; Operating Systems; Tata McGraw Hill; Second Edition. ISBN: 0-07-044700-4
2. Maurice J. Bach; The Design of the Unix Opearating System; Prentice Hall of India; ISBN: 978-81-203-0516-8
3. Uresh Vahalia; Unix Internals, The New Frontiers; Prentice Hall; ISBN: 0-13-101908-2

Guidelines for Term Work Assessment:

Term work assessment will be based on overall performance of Laboratory assignments performed by students.

Guidelines for Practical Examination :

- Internal and External Examine will take jointly oral examination of not more than 2 students at a time.

- Understanding of the fundamentals, effective and efficient implementation.

Guidelines for Laboratory Conduction :

- Operating System recommended :- 64-bit Open source Linux or its derivative
- Programming tools recommended: - C++/ Java/ Python

Practical Assignments

1. Demonstration of Installation of Linux Operating System and Exploration of Unix/Linux Commands (File, Directory and Process commands).
2. Write a program to implement operations on processes using fork and join system calls.
3. Simulation of the scheduling algorithms. For example: First Come First Serve (FCFS), Shortest Remaining Time Next (SRTN).
4. Simulation of scheduling algorithms. For example: Round-Robin (RR), Pre-emptive Priority scheduling.
5. Write a program to implement Reader-Writer problem using semaphores
6. Write a program to implement Producer-Consumer problem
7. Write a program to implement Banker's Algorithm for deadlock handling
8. Simulation of Page replacement algorithms. For example: First-In-First-Out, Least Recently Used, optimal page replacement.
9. Simulation of memory allocation strategies. For example: First Fit, Best Fit and Worst Fit.

CO23214: Software Engineering

Teaching Scheme:	Credit: 03	Examination Scheme:
TH: 03 Hrs/Week		Course Activity: 20 Mark In-Semester: 20 Mark End-Semester: 70 Mark

Prerequisite:

Programming and Problem Solving

Course Objective:

- To learn and understand the principles of Software Engineering.
- To be acquainted with methods of capturing, specifying, visualizing and analyzing software requirements.
- To apply design and testing principles to software project development.
- To understand project management through life cycle of the project

Course Outcomes:

1. Identify and compare the various Software development models.
2. Describe and organize the software requirements.
3. Prediction of tentative estimate required for software development.
4. Design applicable solutions in one or more application domains using software engineering
5. Identify and handle risk management and software configuration management
6. Utilize knowledge of software testing approaches, approaches to verification and validation

Guideline for Couse Activity:

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Projects
2. Industry Visit
3. Seminar
4. Research Paper
5. Group Discussion

Course Contents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Software Engineering and Software Process Models**	**08 Hours**

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

Mapping of Course Outcomes for Unit II	CO2	
UNIT II	**Software Requirements Engineering and Analysis**	**07 Hours**

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

Mapping of Course Outcomes for Unit III	**CO3**

UNIT III	**Estimation and Scheduling**	**06 Hours**

Estimation for Software Projects: The Project Planning Process, Defining Software Scope and Checking Feasibility, Resources management, Reusable Software Resources, Environmental Resources,
Software Project Estimation: Decomposition Techniques, Software Sizing, Problem-Based Estimation, LOC-Based Estimation, FP-Based Estimation, Object Point (OP)-based estimation, Process-Based Estimation, Use-Case–Based Estimation. **Estimation Models**: Reconciling Estimates, Empirical Estimation Models, The Structure of Estimation Models, The COCOMO II Mode, Preparing Requirement Traceability Matrix **Project Scheduling**: Project Scheduling, Defining a Task for the Software Project, Scheduling.

Mapping of Course Outcomes for Unit IV	**CO4**

UNIT IV	**Design Engineering**	**06 Hours**

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

Mapping of Course Outcomes for Unit V	**CO5**

UNIT V	**Risks and Configuration Management**	**06 Hours**

Risk Management: Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan. **Software Configuration Management**: Software Configuration Management, The SCM Repository The SCM Process, Configuration Management for any suitable software system.

Mapping of Course Outcomes for Unit VI	**CO6**

UNIT VI	**Testing Strategies**	**08 Hours**

A Strategic Approach to Software Testing, Verification and Validation, Organizing for Software Testing,
Test strategies for conventional software: black-box and white-box testing, validation testing, system testing, the art of debugging.
Test Strategies for Object-Oriented Software: Unit Testing in the OO Context, Integration Testing in the OO Context, Test Strategies for WebApps, Validation Testing, Validation-Test Criteria, Configuration Review.

Books and Other Resources

Text Books:

1. Roger Pressman, "Software Engineering: A Practitioner's Approach"‖, McGraw Hill, ISBN 007–337597–7
2. Ian Sommerville, "Software Engineering"‖, Addison and Wesley, ISBN 0-13-703515

Reference Books

1. Carlo Ghezzi, "Fundamentals of Software Engineering", PHI, ISBN-10: 0133056996
2. Rajib Mall, "Fundamentals of Software Engineering"‖, PHI, ISBN-13: 978-8120348981
3. Pankaj Jalote, "An Integrated Approach to Software Engineering"‖, Springer, ISBN 13: 9788173192715.

MDCO23201: Computer Graphics and Gaming

Teaching Scheme:	Credit: 03	Examination Scheme:
TH:02Hrs/Week		Activity :20Marks
PR:02Hrs/Week		MarkInSemester:20Marks
		End Semester :50Marks
		Term Work :20Marks

Prerequisite: C++ programming (CPP)

Companion Course, if any: Computer Graphics Laboratory

Course Objective:

- Remembering: To acquaint the learner with the basic concepts of Computer Graphics
- Understanding: To learn the various algorithms for generating and rendering graphical figures.
- Applying: To get familiar with mathematics behind the graphical transformations
- Understanding: To understand and apply various methods and techniques regarding animation.
- Creating: To generate Interactive graphics using OpenGL

Course Outcomes: Students will be able to

1. Be familiar with the graphics designing concepts and devices.
2. Construct a mathematical design using the development process.
3. Recognize the design principles of animation and gaming application.
4. Implement the use of gaming tools in application design.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Active participation in Gaming Competition
2. Poster Presentation
3. Video Presentations
4. Survey on various Animation making tools
5. Visit to Animation Business Schools

Course Contents

Mapping of Course Outcomes for Unit I		CO1
UNIT I	**Basics of Computer Graphics**	**03 Hours**

Introduction, What is computer Graphics? Area of Computer Graphics, Design and Drawing, Animation Multimedia applications, Simulation, How are pictures actually stored and displayed, Difficulties for displaying pictures. Graphics Devices Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, The Shadow - Mask CRT, Direct View Storage Tube, Tablets, The light Pen, Three Dimensional Devices.

Mapping of Course Outcomes for Unit II		CO2
UNIT II	**Transformations**	**03 Hours**

VPKBIET BARAMATI Affiliated to S.P.P.U; Pune Id No./PU/PN/Engg./152/2000 PUNE (MS.)

Simple line drawing methods, Introduction Point Plotting Techniques Qualities of good line drawing algorithms The Digital Differential Analyzer (DDA), Bresenham's Algorithm Generation of Circles Introduction, what is transformation? Matrix representation of points Basic transformation, Translation, Rotation, Scaling, Need for 3-Dimensional Imaging Techniques for 3-Dimesional displaying, Translation, Rotation, Scaling

Mapping of Course Outcomes for Unit III	**CO3**

UNIT III	**Animation**	**03 Hours**

Animation: Introduction, Conventional and computer-based animation, Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. Design of animation sequences, Animation languages, Key- frame, Morphing, Motion specification. Gaming: Introduction, Gaming platform (NVIDIA, i8060), Advances in Gaming.

Mapping of Course Outcomes for Unit IV	**CO4**

UNIT IV	**Gaming**	**03Hours**

Principles of game design, Game Design Theory,MDA,8 type of Fun in Game, Visual style, Gameplay, Generate ideas for a game concept Idea Development Process, Stimulus, Genre Market Research, Target platform ,Creating Prototype Creating physical Games: Board Game, Card Game, Party Games and etc....

Books and Other Resources

TextBooks:

1. Computer Graphics, Multimedia and Animation ,2010, Pakhira Malay K.
2. Donald D. Hearn and Baker- Computer Graphics with OpenGL, 4th Edition, ISBN-13: 9780136053583

Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, —Computer Graphics Principles and Practice‖, 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Rogers, J. Adams, —Mathematical Elements for Computer Graphics‖, 2nd Edition, Tata McGrawHill Publication, 2002, ISBN 0 – 07 – 048677 – 8.

Guideline for Practical Conduction :

Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.
Operating System recommended :- 64-bit Open source Linux or its derivative, Windows
Programming tools recommended: - Open Source C++ Programming tool like G++/GCC, OPENGL, DEV C++.

Practical Assignments

1.	Write C++ program to draw the line styles using DDA and Bresenham's algorithm.
2.	Write C++ program to draw a 4X4 chessboard.
3.	Write C++ program to draw 2-D object and perform following basic transformations, a) Scaling b) Translation c) Rotation.
4.	Write C++ program to draw Man Walking in the Rain with an Umbrella.
5.	Write a C++ Program to make puzzle game.
6.	Write a C++ Program to make Tic Tac Toe game.
7.	Write a C++ Program to draw a car in motion.