MIT-WPU
॥ विद्यावारिधिर्ददाति ॥

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SYLLABUS

DR VISHWANATH KARAD

MIT - WORLD PEACE UNIVERSITY

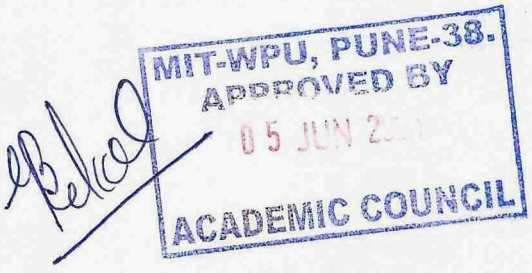
FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING AND TECHNOLOGY

B. Tech Computer Science & Engineering (Artificial Intelligence and Data Science)

BATCH 2024 – 2028



PROGRAMME STRUCTURE

Preamble:

The B. Tech CSE(Artificial Intelligence and Data Science) is the most sought-after branch of Engineering in today world. With the advancements in hardware and software technologies, there is huge scope for development of a wide range of applications. The Internet and allied technologies have connected the world cohesively offering immense opportunities at national and international levels. The students of MITWPU will be tomorrow's global leaders, researchers, entrepreneurs and change-makers. MITWPU has the objective to make them competent for global scenarios.

The B. Tech CSE (Artificial Intelligence and Data Science) curriculum offers a varied range of subjects that fall into the core, specialization and basic sciences categories. The programme also has provisions for pursuing Industry projects, Internships, Foreign and National study tours, Interdisciplinary Projects as a prudential aspect of the course curriculum. The value-based education is ensured by offering Peace related subjects and Yoga practice.

The curriculum is based on the theme of Continuous Evaluation. Theory and Laboratory components are given appropriate importance. The communication skills are enhanced through the component of Seminars. Industry exposure is given through Internships / Projects, and development of latest Technologies is achieved and enhanced through usage of latest Tools.

The curriculum will transform the students into winning personalities.

Dr. Mangesh Bedekar
Professor and Dean,
School of Computer Engineering and Technology

MIT-WPU, PUNE-38.
APPROVED BY
05 JUN 2021
ACADEMIC COUNCIL

Vision and Mission of the Programme

VISION

To be an academic centre of excellence in Computer Science and Engineering to cater to societal needs.

MISSION

- To create conducive environment for nurturing integrity, discipline and technical knowledge in emerging areas of computer science and engineering.
- To encourage students to work in trans-disciplinary domain in collaboration with industry and to inculcate research mindset.
- To develop globally competent graduates to provide solutions for societal problems.

Programme Educational Objectives

The Computer Engineering and Technology Graduate will:

PEO 1 Competent Professionals: Identify and effectively use knowledge of Artificial Intelligence and Data Science to solve real life problems and offer sustainable solutions.

PEO 2 Multifaceted Professionals: Apply research aptitude, innovative mindset and emerging technologies of Artificial Intelligence and Data Science to excel in multidisciplinary domains.

PEO 3 Ethical Professionals: Pursue ethical values, leadership and interpersonal skills during their professional careers for wellbeing of society.

Programme Outcomes (POs)

Computer Science and Engineering Graduates will 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 the 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:** 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 the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

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.

Programme Specific Outcomes (PSOs)

Computer Science and Engineering Graduates will be able to:

PSO 1 Analyse, design and develop computer-based systems to solve real life problems by applying knowledge of Computer Science and Engineering.

PSO 2 Apply descriptive and predictive analytics to arrive at actionable insights from data for solving business/engineering problems

PSO 3 Demonstrate critical thinking and research attitude in computing and data-driven technologies to address interdisciplinary problems

For a UG course at MITWPU the actual credit distribution will be as below:

Course Basket	Credits Assigned
Program Major	48
Program Disciplinary Foundation	34
Program Electives	16
Program Capstone Project, Problem Based Learning, Seminar, and Internships	32
University Core	24
University Electives	9
Total	**163**

MIT-WPU, PUNE-38.
APPROVED BY
05 JUN 2021
ACADEMIC COUNCIL

Assessment Scheme :

Abbreviation	L-T-P-J-C : L-Lecture, T-Tutorial, P-Practical, J-Project, C-Total Credits. CCA1 - Class Continuous Assessment 1, MT - Mid Term Test, CCA2 - Class Continuous Assessment 2, LCA1 - Laboratory Continuous Assessment 1, LCA2 - Laboratory Continuous Assessment 2, LCA3 - Laboratory Continuous Assessment 3, TE - Term End Exam

Type of Course	Assessment Scheme Code	Description L-T-P-J-C	CCA1	MT	CCA2	LCA1	LCA2	LCA3	TE	Total
Theory Courses	TT1	All Theory (L, T) Only courses **with** TE exams	15	30	15	-	-	-	40	100
Theory Courses with Continuous Evaluation	TT2	All Theory (L, T) only courses **without** TE exams	35	30	35	-	-	-	-	100
Lab /Projects/ Internship/ Dissertation	PJ	All courses having P and J components Only	-	-	-	33.33	33.33	33.33	-	100
Theory and Lab Course 1	TL1	2-0-2-0-4	7.5	15	7.5	10	10	10	40*	100
Theory and Lab Course 2	TL2	1-0-3-0-4	2.5	10	2.5	15	15	15	40*	100
Theory and Lab Course 3	TL3	3-0-1-0-4	10	25	10	5	5	5	40	100
Theory and Lab Course 4	TL4	2-0-1-0-3	10	20	10	6.67	6.67	6.67	40	100
Theory and Lab Course 5	TL5	1-0-2-0-3	5	10	5	13.33	13.33	13.33	40*	100
Theory and Lab Course 6	TL6	2-1-1-0-4	10	25	10	5	5	5	40	100
Theory and Lab Course 7	TL7	1-1-1-0-3	10	20	10	6.67	6.67	6.67	40	100

* Term End Exams to be conducted anywhere within the MITWPU Campus subject to the following conditions:
1. All eligible students will be taking this exam in the same space and at the same time slot.
2. The time for Term End Exams will be a maximum of 3 hrs.
3. QP will be sent along with Invigilators by DoE.

PLEASE NOTE: IF ANY OF THE ASSESSMENT CODE COMBINATION AS APPLICABLE TO YOUR PARTICULAR PROGRAM IS NOT AVAILABLE IN THE ABOVE GIVEN CODES, PLEASE CONTACT WITH YOUR ASSOCIATE DEAN ACADEMICS TO HAVE IT INCLUDED FROM DEAN ACADEMICS INCORDINATION WITH THE CONTROLLER OF EXAMINATION.

B. Tech. (Second Year) (Computer Science Engineering- Artificial Intelligence and Data Science) (2024-28)

Semester – III

Sr. No.	Course Code	Name of Course	Type	Weekly Workload, Hrs				Credits	Assessment Scheme Code
				L	T	P	J		
1.	EMT22020	Calculus and Numerical Methods	PF	3	1	--		4	TT1
2.	AID20010	Data Structures	PM	3	--	--		3	TT1
3.	AID20020	Data Structures Laboratory	PR	--	--		3	1	PJ
4.	AID20030	Project Based Learning – I	PR	--	--		3	1	PJ
5.	AID20040	Database Management Systems	PM	3	--			3	TT1
6.	AID20050	Database Management Systems Laboratory	PF			2		1	PJ
7.		University Elective- I	UE	3	--			3	
8.		University Elective- II	UE	3	--			3	
9.	PCE10020	Spiritual and Cultural Heritage: Indian Experience	UC	2	--	--		2	TT1
10.	UNC10110	Research Innovation Design Entrepreneurship	PF	--	--		3	1	I
				17	1	2	9	22	

MITWPU, PUNE-38. APPROVED BY 05 JUN 2024 ACADEMIC COUNCIL

****Assessment Marks are valid only if Attendance criteria are met**

Weekly Teaching Hours: 29

L-Lecture, T-Tutorial, P-Practical, J-Project.

Total Credits: 22

Academic Coordinator | Program Director/HoS | Associate Dean Academics | Dean | Dean Academics MITWPU | Registrar MITWPU

COURSE STRUCTURE

Course Code	AID20010				
Course Category	Program Major				
Course Title	Data Structures				
Teaching Scheme	Lectures	Tutorials	Laboratory / Practical	Project	Total
Weekly load hours	3	--	--	--	3
Credits	3	--	--	--	3
Assessment Schema Code	TT1				

Prerequisites:

- Foundations of Programming

Course Objectives:

By participating in and understanding all facets of this Course a student will be able:

1. **Knowledge:**
   i. Learn the data structure and its fundamental concept
2. **Skill:**
   i. Understand the different Linear and nonlinear data structures such as Arrays, Stacks, Queues Trees and Graph.
   ii. Study the different Sorting Techniques and concepts of Heap and Trees
3. **Attitude:**
   i. Learn how to apply Data Structures to solve Real-life Problems

Course Outcomes:

On completion of course, students will be able to:

1. To demonstrate the use of sequential data structures
2. To analyse different sorting algorithms so as to understand their applications
3. To develop skills for writing and analysing algorithms to solve domain problems
4. To choose appropriate non-linear data structures to solve a given problem

Course Contents:

UNIT 1 Introduction to Data Structures:

Data, Data Objects, Abstract Data types (ADT) and Data Structures, Types of data structures (Linear and Non-linear, Static and dynamic) Introduction to algorithms Analysis of Algorithms-Space complexity, Time complexity.

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

Sorting: Types of Sorting-Internal and External Sorting, General Sort Concepts-Sort Order, Stability, Efficiency, and Number of Passes, Comparison Based Sorting Methods-Bubble Sort, Insertion Sort, Selection Sort, Shell Sort, Comparison of all sorting algorithm

UNIT 2 Linear Data Structures: Arrays

Array as an Abstract Data Type, Sequential Organization, Storage Representation and their Address Calculation: Row major and Column Major, Multidimensional Arrays: Concept of Ordered List, Single Variable Polynomial: Representation using arrays, Polynomial as array of structure, Polynomial addition, Polynomial evaluation and Polynomial multiplication. Sparse Matrix: Sparse matrix representation using array, Sparse matrix addition, Transpose of sparse matrix- Simple and Fast Transpose, Time and Space trade-off.

UNIT 3: Stacks and Queues:

Stacks: Stack as an Abstract Data Type, Representation of Stack Using Sequential Organization, stack operations, Applications of Stack- Expression Conversion and Evaluation, Linked Stack and Operations, Recursion.

Queues: Queue as Abstract Data Type, Representation of Queue Using Sequential Organization, Queue Operations Circular Queue, Advantages of Circular queues, Linked Queue and Operations Deque-Basic concept, types (Input restricted and Output restricted), Application of Queue: Job scheduling.

UNIT 4: Linked List:

Linked List as an Abstract Data Type, Representation of Linked List Using Sequential Organization, Representation of Linked List Using Dynamic Organization, Types of Linked List: singly linked, Circular Linked Lists, Doubly Linked List, Primitive Operations on Linked List, Polynomial Manipulations-Polynomial addition. Generalized Linked List (GLL) concept, Representation of Polynomial using GLL.

Case Study: Garbage Collection

UNIT 5: Tree and Heap:

Tree- Basic Terminology, Binary Tree- Properties, Converting Tree to Binary Tree, Representation using Sequential and Linked organization, Binary tree creation and Traversals, Operations on binary tree. Binary Search Tree (BST) and its operations, threaded binary tree-Creation and Traversal of In-order Threaded Binary tree.

Heap- Heap as a priority queue, Heap sort.

Learning Resources:

Text Books:

1. Fundamentals of Data Structures, E. Horowitz, S. Sahni, S. A-Freed, Universities Press
2. Data Structures and Algorithms, A. V. Aho,. J E. Hopcroft, J D. UIlman, Pearson.

Reference Books:

1. The Art of Computer Programming: Volume 1: Fundamental Algorithms, Donald E. Knuth.

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

2. Introduction to Algorithms, Thomas, H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press.
3. Open Data Structures: An Introduction (Open Paths to Enriched Learning), (Thirty First Edition), Pat Morin, UBC Press.

Supplementary Reading:

1. Aaron Tanenbaum, "Data Structures using C", Pearson Education.
2. R. Gilberg, B. Forouzan, "Data Structures: Apseudo code approach with C", Cenage Learning.

Web Resources:

Web links:

1. https://www.tutorialspoint.com/data_structures_algorithms/index.htm

MOOCs:

1. https://nptel.ac.in/courses/106103069

Pedagogy:

1. Power point presentations
2. Videos
3. Co-teaching
4. Systematic use of group work and project-based learning

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

COURSE STRUCTURE

Course Code	AID20020				
Course Category	PR				
Course Title	Data Structures Laboratory				
Teaching Scheme	Lectures	Tutorials	Laboratory / Practical	Project	Total
Weekly load hours	--	--	3	--	3
Credits	--	--	1	--	1
Assessment Schema Code	PJ				

Prerequisites:

Foundations of Programming

Course Objectives:

1. **Knowledge**
   i. Learn to implement Data structures concepts.

2. **Skills**
   i. Understand the implementation of different Linear and nonlinear data structures such as Arrays, Stacks, Queues Trees and Graph.
   ii. To Implement Different Sorting Techniques and concepts of Heap and Trees

3. **Attitude**
   i. Learn how to implement Data Structures to solve Real-life Problems

Course Outcomes:

After completion of this course students will be able to:

1. To Implement basic data structures such as arrays, linked lists, stacks, and queues in a programming language.
2. To Apply data structures to solve a real-world problem and present the solution
3. Demonstrate the ability to implement data structures in a chosen programming language

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

Course Contents:

1. Write a program to create a student database using an array of structures. Apply sorting techniques (Bubble sort , Insertion Sort, Selection Sort).
2. Write a C program for sparse matrix realization and operations on it- Simple Transpose, Fast Transpose.
3. Implement stack as an ADT and apply it for different expression conversions (infix to postfix or infix to prefix (Any one), prefix to postfix or prefix to infix, postfix to infix or postfix to prefix (Any one)).
4. Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write a program for simulating job queue. Write functions to add job and delete job from queue.
5. Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request. Similarly, one may cancel the membership of club. First node is reserved for president of club and last node is reserved for the secretary of the club. Write C program to maintain club member 's information using singly linked list. Store student PRN and Name. Write functions to: a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) sorting of two linked list e) merging of two linked list f) Reversing using three pointers
6. Implement binary tree and perform following operations: Creation of binary tree and traversal recursive and non-recursive.
7. Implement a dictionary using a binary search tree where the dictionary stores keywords & its meanings. Perform following operations:
   - Insert a keyword
   - Delete a keyword
   - Create mirror image and display level wise
   - Copy
8. Implement a threaded binary tree. Perform inorder traversal on the threaded binary tree.
9. Read the marks obtained by students of second year in an online examination of a particular subject. Find the maximum and minimum marks obtained in that subject. Use heap data structure and heap sort.

Learning Resources:

Text Books:

1. Fundamentals of Data Structures, E. Horowitz, S. Sahni, S. A-Freed, Universities Press
2. Data Structures and Algorithms, A. V. Aho,. J E. Hopcroft, J D. UIlman, Pearson.

Reference Books:

1. The Art of Computer Programming: Volume 1: Fundamental Algorithms, Donald E. Knuth.

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

2. Introduction to Algorithms, Thomas, H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press.
3. Open Data Structures: An Introduction (Open Paths to Enriched Learning), (Thirty First Edition), Pat Morin, UBC Press.

Supplementary Reading:

1. Aaron Tanenbaum, "Data Structures using C", Pearson Education.
2. R. Gilberg, B. Forouzan, "Data Structures: Apseudo code approach with C", Cenage Learning.

Web Resources:

Web links:

1. https://www.tutorialspoint.com/data_structures_algorithms/index.htm

MOOCs:

1. https://nptel.ac.in/courses/106103069

Pedagogy:

1. Power point presentations
2. Videos
3. Laboratory Implementation

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

COURSE STRUCTURE

Course Code	AID20030				
Course Category	PR				
Course Title	Project Based Learning – I				
Teaching Scheme	Lectures	Tutorials	Laboratory / Practical	Project	Total
Weekly load hours				3	3
Credits				1	1
Assessment Schema Code	PJ				

Prerequisites:

- Students should be familiar with how computers work, basic hardware components, and the concept of software will be helpful.
- Students should have a grasp of fundamental programming concepts like variables, data types, operators, control flow

Course Objectives:

By participating in and understanding all facets of this Course a student will be able:

1. **Knowledge:**
   i. Understand the fundamental principles of Object-Oriented Programming (OOP):
   ii. Grasp the advantages of OOP over procedural programming paradigms.
   iii. Apply OOP concepts to design and implement well-structured C++ programs.
   iv. Be familiar with advanced OOP features like templates, exceptions, and STL
2. **Skill:**
   i. Design and create classes to encapsulate data and behavior
   ii. Implement constructors, destructors, and member functions for object initialization, memory management, and functionality.
   iii. Write well-structured, maintainable, and efficient C++ code using OOP principles
   iv. Employ advanced OOP features like templates, exceptions, and the STL to solve complex problems.
3. **Attitude:**
   i. Develop a problem-solving approach that leverages the benefits of OOP for modularity, maintainability, and reusability.
   ii. Appreciate the importance of code clarity, documentation, and testing in OOP development.
   iii. Foster a desire for continuous learning and exploration of advanced OOP concepts

Course Outcomes:

On completion of course, students will be able to:

1. Identify the core OOP concepts

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

2. Describe the purpose of classes in C++ for representing real-world entities.
3. Implement and utilize inherited members within derived classes.
4. Implement concept of polymorphism and exception handling to solve real world problems
5. Demonstrate good coding practices in developing efficient and maintainable C++ programs by using advanced c++ concepts

Laboratory Exercises / Practical:

Paradigms of Programming: Procedural vs. Object-Oriented Programming (OOP) - Key differences and benefits of OOP.OOP Concepts: Encapsulation, Abstraction, Inheritance, and Polymorphism - Understanding their roles in OOP design.Benefits of OOP: Code reusability, maintainability, modularity, and real-world problem modelling.C++ Basics: Data types, operators, control flow statements, functions, arrays

Write programs to solve a simple problem (e.g., calculating area) using both procedural and object-oriented approaches. Compare the code structure, maintainability, and benefits of object-oriented programming.
Accept User Choice for Various Shape-Area Calculations.

Write a C++ Code to use Abstraction, Inheritance, and Polymorphism

Defining Classes: Creating blueprints for objects with member variables (data) and member functions (behaviour). Object Creation and Member Access: Using Constructor and destructor create/delete objects, use of dot operator (.) and arrow operator. Encapsulation: Data hiding and member access control (public, private, protected) for information protection.Methods (Member Functions): Defining member functions within classes to operate on object data, creation of friend class and friend function.

Design a class to represent a bank account with attributes (balance, account number) and methods (deposit, withdraw) with proper access control (public, private, protected) to demonstrate encapsulation.

Define classes for Circle and Rectangle with constructors to initialize their dimensions and member functions to calculate area and perimeter. Create objects and demonstrate their usage. Create Friend Class and Friend Function with appropriate use of it.

Concept of Inheritance: Deriving new classes (derived classes) from existing classes (base classes) for code reuse. Types of Inheritance: Single inheritance, multiple inheritance, hierarchical inheritance - Understanding their usage and potential complexities. Base Class and Derived Class Relationships: Accessing base class members, constructor chaining, and polymorphism introduction. Polymorphism: Definition, Benefits,Types of Polymorphism: Compile -Time Polymorphism, Function Overloading, Operator Overloading, Runtime Polymorphism, Virtual Functions, Dynamic Binding,Exception Handling: Introduction, Exception Handling Mechanism - try, catch and throw, Multiple Exceptions. Abstract Classes and Interfaces. File Handling

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

Create functions for arithmetic operations (+, -, *) that work with both integers and user-defined classes (e.g., Point with X and Y coordinates) demonstrating function overloading.

Develop a base Shape class with a virtual function for drawing. Create derived classes for Circle and Rectangle that override the draw function. Write a program that uses polymorphism to call the appropriate draw function for different shapes.

Write a program that reads data (e.g., student information) from a text file and stores it in objects. Implement functionality to write data from objects back to a text file.

Write Exception Handling Code for various Exceptions in C++

Advanced OOP Concepts:Templates: Introduction to Template, types of templates, Function Template, overloading Function templates, Class Template, Introduction to Standard Template Library (STL): Containers (vectors) and iterators (concepts)

Utilize a vector container from the STL to store student grades. Implement functions to calculate average, minimum, and maximum grades using iterators to access elements in the vector.

Project

Develop an application that simulates a real-world scenario (based on your chosen domain) using object-oriented principles. Focus on designing classes, objects, and their interactions to model the system.

Note:

1. Each student is required to perform all practical exercises based on a specified list. The students are expected to document their activities in a journal. The submitted journal will be evaluated as part of continuous assessment.
2. A group of four students will collaborate on a Project-Based Learning (PBL) activity. This activity is based on the eighth practical exercise. The group is required to submit a report along with a presentation. PBL activity will be assessed as part of the Lab Continuous Assessment.

Learning Resources:

Text Books:

1. Robert Lafore, 'Object-Oriented Programming in C++', Fourth Edition, Sams Publishing, ISBN: 0672323087
2. Deitel, 'C++ How to Program',10th Edition, Pearson Education, ISBN 13:9780134448237
3. Yashavant Kanetkar, 'Let Us C++', BPB Publications, ISBN: 9388176642

Reference Books:

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT.WPU, PUNE-38.
...WED BY

1. Herbert Schildt, 'C++ The Complete Reference', Eighth Edition, McGraw Hill Professional, 2011, ISBN-13:978-0072226805
2. Bjarne Stroustrup, 'The C++ Programming language', Seventh Edition, Pearson Education, ISBN: 9788131705216
3. K. R. Venugopal, Rajkumar Buyya, T. Ravishankar, 'Mastering C++', Tata McGraw-Hill, ISBN 13:9780074634547
4. E. Balguruswamy, 'Object-oriented Programming with C++", 7th edition, ata McGraw-Hill, ISBN 10:9352607996

Supplementary Reading:

1. Power Point Slides
2. Practice Assignments

Web Resources:

1. https://link.springer.com/book/10.1007/978-1-349-23304-5
2. https://dl.ebooksworld.ir/motoman/Apress.Beginning.Cplusplus.17.From.Novice.to.Professional.5th.Edition.www.EBooksWorld.ir.pdf
3. https://link.springer.com/chapter/10.1007/978-3-030-28346-9_4?fromPaywallRec=true

Web links:

1. https://www.w3schools.com/cpp/
2. https://www.javatpoint.com/cpp-tutorial
3. https://www.geeksforgeeks.org/cpp-tutorial/
4. https://www.learncpp.com/

MOOCs:

1. https://archive.nptel.ac.in/courses/106/105/106105151/
2. https://archive.nptel.ac.in/courses/106/105/106105234/
3. https://www.coursera.org/specializations/hands-on-cpp
4. https://www.my-mooc.com/en/mooc/introduction-c-microsoft-dev210x-0/

Pedagogy:

1. PPTs
2. Practical Demos
3. Video ,Expert Lecture

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY

COURSE STRUCTURE

Course Code	AID20040				
Course Category	Project Major				
Course Title	Database Management Systems				
Teaching Scheme	Lectures	Tutorials	Laboratory / Practical	Project	Total
Weekly load hours	3	--	--	--	3
Credits	3	--	--	--	3
Assessment Schema Code	TT1				

Prerequisites

- Fundamentals of Data Structures
- Discrete Mathematics and Graph Theory
- Software Engineering and Project Management
- Basics of DBMS

Course Objectives:

By participating in and understanding all facets of this Course a student will be able:

1. Knowledge

i. Understand the fundamental concepts of Database Management System

ii. To provide a strong formal foundation of Database concepts, DBMS architectures, recent technologies and best industry practices

2. Skills

i. To learn the SQL Database system

ii. To program PL/SQL including stored procedures, stored functions, cursors and packages

3. Attitude

i. To design the Database system for real world applications

ii. To access, modify, program and authenticate the Database system

Course Outcomes:

On completion of course, students will be able to:

1. Gain a good understanding of the architecture and functioning of database management systems
2. Analyze and apply the principles and practices of good database design.
3. Make use of SQL and construct relevant queries to retrieve the data
4. Implement relational Database design from any data model
5. Handle basic issues of transaction processing and concurrency control

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.

Course Contents:

Unit I Introduction to DBMS and Data Modelling: DBMS Vs File Systems, Database System Architecture, Database Architectures: Centralized, Client-Server, Parallel, Distributed, Data Abstraction, Data Independence, Data Definition and Data Manipulation Languages, Data Models, E-R diagram: Components of E-R Model, Conventions, Keys, EER diagram Components, Reduce E-R diagram into Tables.

Unit II Relational Database Design and Normalisation: Relational Model: Attributes, Tuple, Domain, CODD's rule Relational Integrity, Referential Integrities, Enterprise Constraints, Normalization: 1NF, 2NF, 3NF, BCNF, Functional dependency, Decomposition Query Processing: Overview, Measures of Query cost, Selection and Join operations, Evaluation of Expressions.

Unit III Relational Algebra and Querying Database: Relational Algebra, Basic Operations, Relational calculus: Tuple Calculus, Domain Calculus, Introduction to SQL, Characteristics and advantages of SQL, SQL Data Types, DDL Commands, DCL Commands. SQL Queries: DML Queries with Select Query Clauses, Creating, Modifying, Deleting. Views: Creating, Dropping, Updating, Indexes, SQL DML Queries, Set Operations, Predicates and Joins, set membership, Grouping and Aggregation, Aggregate Functions, Nested Queries.

Unit IV Introduction Procedural Languages in SQL(PL/SQL): PL/SQL Concepts: Basic PL/SQL block, PL/SQL Functions and Procedures, Different modes in procedure, Difference between functions and procedures, Cursors, Database Triggers. Exception Handling in PL/SQL, Application of DBMS.

Unit V Transaction and Recovery Management: ACID properties, transactions, schedules and concurrent execution of transactions, serializability: View, Conflict. Concurrency control lock-based protocol (simple, 2 phases: Rigorous 2 phase, Strict 2 phase), Cascade-less Schedule, Recoverable Schedule, Deadlocks: Prevention Techniques (Wait Die, Wound Wait), Detection Techniques, Database Recovery: Failure classification Recovery and atomicity: Log-based recovery, Shadow paging.

Learning Resources:

1. Ivan Bayross, "SQL, PL/SQL: The Programming Language of Oracle", 3rd Edition BPB Publication
2. Reese G., Yarger R., King T., Williums H, "Managing and Using MySQL", O'Reilly Media, Inc., ISBN: 9780596002114, 2nd Edition

Web Resources:

1. https://www.mysql.com/
2. http://www.oracletutorial.com/
3. https://www.oreilly.com/library/view/managing-using/0596002114/

Weblinks:

1. https://www.db-book.com/db6/slide-dir/
2. https://www.mysqltutorial.org/

MOOCs:

1. https://nptel.ac.in/courses/106105175
2. https://www.coursera.org/learn/Database-management

Pedagogy:

Associate Dean Academics (School)

Dean Academics MITWPU

Registrar MITWPU

MIT-WPU, PUNE-38. APPROVED BY

1. White Board
2. PowerPoint Presentations
3. Expert Lecturers
4. Blended Teaching Learning
5. Flipped Classroom, Think Pair Share
6. Project Based Learning

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.

COURSE STRUCTURE

Course Code	AID20050				
Course Category	PF				
Course Title	Database Management Systems Laboratory				
Teaching Scheme	Lectures	Tutorials	Laboratory / Practical	Project	Total
Weekly load hours	--	--	2	--	2
Credits	--	--	1	--	1
Assessment Schema Code	PJ				

Prerequisites

- Fundamentals of Data Structures
- Discrete Mathematics and Graph Theory
- Software Engineering and Project Management
- Basics of DBMS

Course Objectives:

By participating in and understanding all facets of this Course a student will be able:

1. Knowledge

i. Understand the fundamental concepts of Database Management System

ii. To provide a strong formal foundation of Database concepts, DBMS architectures, recent technologies and best industry practices

2. Skills

i. To learn the SQL Database system

ii. To program PL/SQL including stored procedures, stored functions, cursors and packages

3. Attitude

i. To design the Database system for real world applications

ii. To access, modify, program and authenticate the Database system

Course Outcomes:

On completion of course, students will be able to:

1. Ability to construct an ER model and reduce it to the relational schemas from the model.
2. Ability to grant and revoke privileges and comprehend database recovery techniques.
3. Make use of SQL to construct efficient SQL queries to retrieve and manipulate data as required.
4. Construct PL/SQL blocks for handling procedure, functions and triggers for real world applications
5. Develop Database system from scratch for handling real world challenges

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.

Course Contents:

Laboratory Exercises / Practical:

1. Case Study on ER and Conversion of ER into schemas
2. SQL- DDL commands (Create, Alter, Drop, Truncate Rename, Describe), DCL (Grant, Revoke)
3. SQL- DML (Insert, Update, Delete), SQL Select- Logical IN, Negation, NULL, Comparison Operators. Where Clause, Between AND, Exists, ALL, LIKE
4. SQL Queries on: Functions-Single Row, Aggregate Functions, Data Sorting, Subquery, Group by-Having, Set Operations, View. TCL Commands
5. SQL Queries on Joins (Inner, Outer, Natural, Self)
6. PL/SQL Procedures and Functions
7. PL/SQL Triggers
8. PLSQL Cursors

Learning Resources:

1. Ivan Bayross, "SQL, PL/SQL: The Programming Language of Oracle", 3rd Edition BPB Publication
2. Reese G., Yarger R., King T., Williums H, "Managing and Using MySQL", O'Reilly Media, Inc., ISBN: 9780596002114, 2nd Edition

Web Resources:

1. https://www.mysql.com/
2. http://www.oracletutorial.com/
3. https://www.oreilly.com/library/view/managing-using/0596002114/

Weblinks:

1. https://www.db-book.com/db6/slide-dir/
2. https://www.mysqltutorial.org/

MOOCs:

1. https://nptel.ac.in/courses/106105175
2. https://www.coursera.org/learn/Database-management

Pedagogy:

1. White Board
2. PowerPoint Presentations
3. Expert Lecturers
4. Blended Teaching Learning
5. Flipped Classroom, Think Pair Share
6. Project Based Learning

Associate Dean
Academics
(School)

Dean Academics
MITWPU

Registrar
MITWPU

MIT-WPU, PUNE-38.
APPROVED BY