Pimpri Chinchwad Education Trust's

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR NO. 26, PRADHIKARAN, NIGDI, PUNE 411044

An Autonomous Institute Approved by AICTE and Affiliated to SPPU, Pune

DEPARTMENT OF INFORMATION TECHNOLOGY

Curriculum Structure and Syllabus
of
Minor in Software Development
(Course 2020)

Effective from Academic Year 2023-24
(Updated with Minor Changes)

Institute Vision

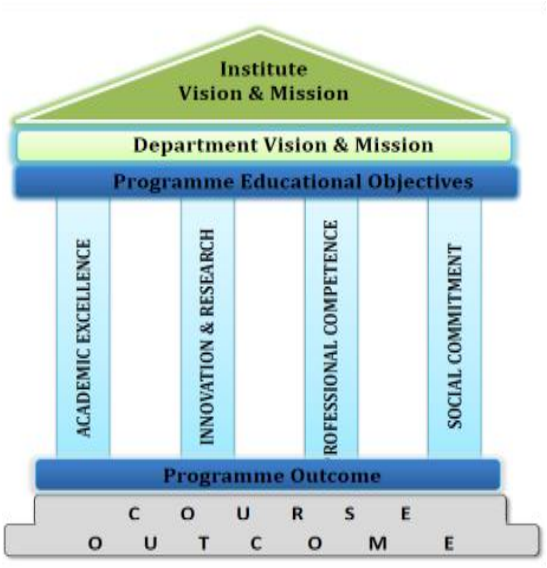
To be one of the top 100 Engineering Institutes of India in coming five years by offering exemplarily Ethical, Sustainable and Value Added Quality Education through a matching ecosystem for building successful careers.

Institute Mission

1. Serving the needs of the society at large through establishment of a state-of-art Engineering Institute
2. Imparting right Attitude, Skills, Knowledge for self-sustenance through Quality Education
3. Creating globally competent and Sensible engineers, researchers and entrepreneurs with an ability to think and act independently in demanding situations

Quality Policy

We at PCCOE are committed to impart Value Added Quality Education to satisfy the applicable requirements, needs and expectations of the Students and Stakeholders. We shall strive for academic excellence, professional competence and social commitment in fine blend with innovation and research. We shall achieve this by establishing and strengthening state-of- the-art Engineering and Management Institute through continual improvement in effective implementation of Quality Management System.



	Pimpri Chinchwad Education Trust's **Pimpri Chinchwad College of Engineering**	
Course Approval Summary		

A) Board of Study-Information Technology

Sr. No.	Name of the Course	Course Code	Page Number	Signature and Stamp of BoS
1	Object-Oriented Programming	MIT5991	5	
2	Object-Oriented Programming Laboratory	MIT5992	7	
3	Data Structure and Algorithms	MIT6991	10	
4	Data Structure and Algorithms Laboratory	MIT6992	12	
5	Database Management System	MIT7991	15	
6	Database Management System Laboratory	MIT7992	17	
7	Software Engineering	MIT7993	18	
8	**Project**	MIT8991	21	

Approved by Academic Council:

Chairman, Academic Council
Pimpri Chinchwad College of Engineering

Approved by Board of Governors:

Chairman, Board of Governors
Pimpri Chinchwad College of Engineering

Preface

Looking at Global Scenario to enhance the employability skills and impart deep knowledge in emerging/ multidisciplinary areas, an additional avenue is provided to passionate learners through the Minors and Honors Degree Scheme in academic structure.

For **Minors degree** program, student has to earn additional 20 credits in multidisciplinary areas of other domains.

Objectives of Minors Degree

- To impart knowledge in multidisciplinary areas.
- To provide effective yet flexible options for students to achieve basic to intermediate level competence in the multidisciplinary area.
- To enhance the employability skills through different combinations in the diverse fields of engineering.
- To provide an academic mechanism for fulfilling multidisciplinary demands of industries.
- To provide a strong foundation to students aiming to pursue research/ higher studies in an interdisciplinary field of study.

.

Preface of Minor in software Development

In the era of Rapid Automation the knowledge of programming becomes critical skill to meet industry demands and fetch a job. In engineering streams-

Mechanical automates the **Civil** by the invention of Wheel, Gears, Pulleys, IC Engine, Etc.

Electrical Automates the Mechanical by Motors.

Electronics Automates Electrical by Control circuits.

Programming Automates Electronics by Automating the Circuits.

So we can combine the following-

Programming -> Electronics -> Electrical -> Mechanical.

So every engineering student should learn programming for TWO purposes.

1. Computation/Simulation.
2. Real Time Control -> Robotics, etc.

Core engineering courses provide a strong foundation to the students, but growing technology requires new methods and ways to progress, prosperity and to inculcate the **Problem-Solving Techniques**. Multidisciplinary activity with professionals from many different backgrounds interacting together to build systems.

With the motivation to develop the course that a bridge gap between Interdisciplinary Engineering Programs and put in the work, Information Technology Departments floated the Minor Course as- "Software Development". The software development minor provides students with skill sets that will enable them to apply the problem-solving skills to the problem they can solve using the concepts they learned through this minor. Software Development focuses on discovering, creating, and designing a practical solution to a problem with a system. Software Development provides students with the knowledge and skills in software development using programming languages. This study includes the - Object Oriented Programming, knowledge of Algorithms and Data Structures in practice, Database Management System, and Software Engineering.

Objectives:

This program aims to:

- Learn fundamentals of object-oriented concepts and programming
- Apply the concepts of object-oriented paradigm.
- Apply linear and non-linear data structures to the applications.
- Implement abstract properties of various linear and non-linear data structures
- Understand the fundamental concepts of database systems.
- Understand the software engineering fundamentals.

Learning Outcomes:

At the successful completion of this Minor program, students will able to:

- Develop programs that utilize the concept of operator overloading, inheritance, exception handling
- Compare and select appropriate searching and sorting techniques in the application development.
- Apply fundamental elements of database management systems.
- Apply software engineering principles to develop software.
- Develop projects applying object-oriented concepts.
- Practice acquired knowledge within the chosen technology area for project development.

INDEX

LIST OF ABBREVIATIONS IN CURRICULUM STRUCTURE

Sr. No.	Abbreviation	Expansion
1.	L	Lecture
2.	P	Practical
3.	T	Tutorial
4.	H	Hours
5.	Cr	Credits
6.	IE	Internal Evaluation
7.	ETE	End Term Evaluation
8.	TW	Term Work
9.	OR	Oral
10.	PR	Practical
11.	PROJ	Project

Curriculum Structure

Minor in Software Development

CURRICULUM STRUCTURE

Structure for Minor in Software Development. (Information Technology)

Semester – V		Teaching Scheme					Evaluation Scheme					
Course Code	**Course Name**	**L**	**T**	**P**	**H**	**Cr**	**CE**	**ETE**	**TW**	**PR**	**OR**	**Total**
MIT5991	Object-Oriented Programming	3	-	-	3	3	50	50	-	-	-	100
MIT5992	Object-Oriented Programming Laboratory	-	-	2	2	1	-	-	-	-	25	25
Total		**3**	**-**	**2**	**5**	**4**	**50**	**50**	**-**	**-**	**25**	**125**
Semester – VI		**Teaching Scheme**					**Evaluation Scheme**					
Course Code	**Course Name**	**L**	**T**	**P**	**H**	**Cr**	**CE**	**ETE**	**TW**	**PR**	**OR**	**Total**
MIT6991	Data Structure and Algorithms	3	-	-	3	3	50	50	-	-	-	100
MIT6992	Data Structure and Algorithms Laboratory	-	-	4	4	2	-	-	25	25	-	50
Total		**3**	**-**	**4**	**7**	**5**	**50**	**50**	**25**	**25**	**-**	**150**
Semester – VII		**Teaching Scheme**					**Evaluation Scheme**					
Course Code	**Course Name**	**L**	**T**	**P**	**H**	**Cr**	**CE**	**ETE**	**TW**	**PR**	**OR**	**Total**
MIT7991	Database Management System	3	-	-	3	3	50	50	-	-	-	100
MIT7992	Database Management System Laboratory	-	-	2	2	1	-	-	-	25	-	25
MIT7993	Software Engineering	3	-	-	3	3	50	50	-	-	-	100
Total		**6**	**-**	**2**	**8**	**7**	**100**	**100**	**-**	**25**	**-**	**225**
Semester – VIII		**Teaching Scheme**					**Evaluation Scheme**					
Course Code	**Course Name**	**L**	**T**	**P**	**H**	**Cr**	**CE**	**ETE**	**TW**	**PR**	**OR**	**Total**
MIT8991	**Project**	-	-	8	8	4	-	-	50	-	50	100
Total		**-**	**-**	**8**	**8**	**4**	**-**	**-**	**50**	**-**	**50**	**100**
Total Credit for Semester V+VI+VII+VIII = 20												

Abbreviations: Course Abbreviation; **L**- Lecture; **P**- Practical; **H**- Hours; **CR**- Credits; **IE**– Internal Evaluation-1; **MTE** – Mid Term Examination **ETE** – End Term Examination; **TW** – Term Work; **OR** – Oral Exam.

CourseSyllabus

Semester-V

Minor in Software Development

Program:	Minor in Software Development				Semester: V	
Course:	Object Oriented Programming				Code:MIT5991	
Teaching Scheme				Evaluation Scheme		
Lecture	Tutorial	Credit	Hours	CE	ETE	Total
3	-	3	3	50	50	**100**

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Learn fundamentals of object-oriented concepts and programming
2. Apply the concepts of object-oriented paradigm.
3. Design and implement applications using object oriented programming.

Outcomes:

After learning the course, the students will be able to:

1. Construct classes for given real life data and operations.
2. Apply operator overloading to develop programs
3. Design hierarchy of classes using inheritance.
4. Make use of polymorphism using virtual function and exception handling to develop programs
5. Apply generic programming using templates.
6. Develop an application using file handling

Detailed Syllabus

Unit	Description	Duration (Hours)
1.	**FUNDAMENTALS OF OBJECT ORIENTED PROGRAMMING** objects, classes, data members, member functions, data encapsulation, data abstraction and information hiding, inheritance, polymorphism. Access Control, constructors, destructors, operators, static data members, array of objects, 'const' keyword and reference variables.	6
2.	**FUNCTIONS AND OPERATOR OVERLOADING** Function prototypes, default and constant arguments, inline functions, static member functions, Function overloading, friend function, Operator Overloading : Need, rules,overloading the unary and binary operators using member and friend function, overloading relational and and assignment operator	8
3.	**INHERITANCE** Need of inheritance, base and derived classes, member access control, types of inheritance, derived class constructor, constructors in multiple inheritance, overriding member functions, ambiguity in multiple inheritance and virtual base class.	7
4.	**VIRTUAL FUNCTIONS AND EXCEPTION HANDLING** Pointers to objects, 'this' pointer, Pointers to derived class, virtual function, pure virtual function, abstract class, virtual destructors, early and late binding Exception Handling: try, catch and throw, MultipleExceptions, Exceptions with arguments.	8

5.	**TEMPLATES** Function template and class template, overloading function template,member function templates and template arguments, Introduction to Standard Template Library (STL), containers, iterators and algorithms. Template class-List, Vector	8
6.	**FILE I/O** Classes for file stream operations, file operations (open, close, read, write,detect end of file), file modes, File pointers and their manipulations (sequential and random access), error handling during file operations.	8
	Total	**45**

Text Books:

1. E. Balaguruswamy, "Object-oriented Programming with C++", Tata McGraw Hill, 7th edition.
2. Robert Lafore, "Object-Oriented Programming in C++", SAMS Techmedia

Reference Books:

1. Herbert Schildt, "C++: The Complete Reference", McGraw-Hill.
2. Kogent, "Object-Oriented Programming Methodology", Wiley, ISBN-9789351191841

Program:	Minor in Software Development					Semester: V	
Course:	Object Oriented Programming Laboratory					Code:MIT5992	
Teaching Scheme				Evaluation Scheme			
Practical	Tutorial	Credit	Hours	TW	Oral	PR	Total
2	-	1	2	-	25	-	**25**

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Apply the concepts of object oriented paradigm.
2. Develop object oriented programming skills.
3. Design and implement applications for real life problems using object oriented programming.

Outcomes:

After learning the course, the students will be able to:

1. Construct classes for given real life data and operations.
2. Develop programs which utilizes the concept of operator overloading.
3. Design hierarchy of classes using inheritance.
4. Implement programs using virtual function and exception handling
5. Construct a program using templates.
6. Develop an application using file handling

Detailed Syllabus

Assignment	Suggested List of Assignments
1	Create a class named weather report that holds a daily weather report with data members day_of_month, hightemp, lowtemp, amount_rain and amount_snow. Use different types of constructors to initialize the objects. Also include a function that prompts the user and sets values for each field so that you can override the default values. Write a menu driven program in C++ with options to enter data and generate monthly report that displays average of each attribute
2	Design a class 'Distance' with data members distance in kilometers, meters and centimeters Provide default and parameterized constructors. Write a program using operator overloading 1. Addition using friend function Subtraction using member function
3	An organization has to maintain information of employees for calculation of salary. An employee can be a waged or fixed salaried employee. A waged employee is paid on an hourly basis. If a salesperson is a waged employee who is paid commission on sales made. Design all the above classes and implement member functions to accept employee details, display and compute salary of an employee

4	Create a base class shape with two double type values and member functions to input the data and compute_area() for calculating area of figure. Derive two classes' triangle and rectangle. Make compute_area() as a virtual function and redefine this function in the derived class to suit their requirements. Write a program that accepts dimensions of triangle/rectangle and displays calculated area. Display error codes and messages by using appropriate try and catch block to handle the exceptions thrown
5	Write a program in C++ using function/class template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.
6	Write a program in C++ to implement a sequential file. Design a base class with name, date of birth, blood group and another base class consisting of the data members such as height and weight. Design one more base class consisting of the insurance policy number and contact address. The derived class contains the data members' telephone numbers and driving license number. Write a menu driven program to carry out the following things: 1. Build a master table 2. Display 3. Insert a new entry 4. Delete entry 5. Edit Search for a record

Text Books:

1. E. Balaguruswamy, "Object-oriented Programming with C++", Tata McGraw Hill, 5th edition.
2. Robert Lafore, "Object-Oriented Programming in C++", SAMS Techmedia

Reference Books:

1. Herbert Schildt, "C++: The Complete Reference", McGraw-Hill.
2. kogent, "Object-Oriented Programming Methodology", Wiley, ISBN-9789351191841

CourseSyllabus

Semester-VI

Minor in Software Development

Program:	Minor in Software Development				Semester: VI	
Course:	Data Structures and Algorithms				Code:MIT6991	
Teaching Scheme				Evaluation Scheme		
Lecture	Tutorial	Credit	Hours	CE	ETE	Total
3	-	3	3	50	50	**100**

Objectives:
After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Analyze algorithms with respect to time and space complexity.
2. Select appropriate searching and sorting techniques in the application development.
3. Apply linear and non-linear data structures to the applications.

Outcomes:
After learning the course, the students will be able to:

1. Analyze algorithms with respect to time and space complexity.
2. Compare and select appropriate searching and sorting techniques in the application development.
3. Construct Linear data structures using arrays and linked lists.
4. Make use of abstract properties of stack and queue data structures.
5. Apply algorithms to create and traverse non-linear data structure trees.
6. Solve the concept of graphs for various problems.

Detailed Syllabus

Unit	Description	Duration (Hours)
2	**Introduction to Data Structures and Analysis of Algorithms** Basics of Data structure, Types, Analysis of algorithm: frequency count, Time & Space complexity, Big 'O', 'Ω' and 'Θ' notations, Best, Worst, and Average case.	6
2.	**Linear Data Structures** Sequential organization: One dimensional and Multidimensional arrays and their storage representation: row-major and column-major form and address calculation. Linked organization: Types of linked list, its operations, applications	8
3.	**Stacks and Queue** Stack as an ADT, implementation of the stack, applications. Queue as an ADT, implementation of the queue, types and applications.	8
4.	**Trees** Trees terminology, and binary trees. Expression tree. Conversion of the general tree to binary tree. Binary tree as an ADT. Binary tree traversals, Binary search trees, Applications.	8
5.	**Graphs** Graph terminology, Representation of graphs, Depth First Search and Breadth-First Search traversal. Algorithms for minimum spanning tree, shortest path algorithm, applications.	8

6	**Searching and Sorting** Searching and sorting algorithms and its Complexity analysis.	7
	Total	45

Text Books:

1. E. Horowitz, S. Sahani, S. MEhata, Dinesh, "Fundamentals of Data Structures in C ++", Universities Press, (2008)
2. Aho, Alfred V., Hopcroft, John. E., Ullman Jeffrey D, " Data Structures And Algorithms", Pearson Prentice Hall.

Reference Books:

1. G. A.V. Pai, "Data Structures and Algorithms ", McGraw Hill, ISBN -13: 978-0-07-066726-6
2. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
3. J. Tremblay, P. Soresan, "An Introduction to Data Structures with Applications", 2nd edition, Tata McGraw Hill International Editions, 1984, ISBN-0-07-462471-7.

Program:	Minor in Software Development					Semester: VI	
Course:	Data Structures and Algorithms Laboratory					Code: MIT6992	
Teaching Scheme				Evaluation Scheme			
Lecture	Practical	Tutorial	Credit	TW	OR	PR	Total
-	4	-	2	25	-	25	50

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Analyze algorithms with respect to time and space complexity.
2. Implement abstract properties of various linear and non-linear data structures

Outcomes:

After learning the course, the students will be able to:

1. Select appropriate searching and sorting techniques in the application development.
2. Make use of abstract properties of various data structures such as stacks, queues.
3. Implement non-linear data structures such as trees, graphs.

Assignment No.	Suggested List of Assignments (Write any 6 programs in C++)
1.	Represent sets using one-dimensional arrays and implement functions to perform i. Union ii. Intersection iii. Difference iv. Symmetric difference of two sets
2.	Implement a Polynomial with operations like creates a polynomial, insert an entry into a polynomial, add two polynomials and evaluate a polynomial.
3.	Implement any database using a doubly-linked list with the following options ● Insert a record ● Delete a record ● Modify a record ● Display list forward ● Display list backward
4.	Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix, and evaluation of postfix and prefix expression.

5.	Implement Circular Queue using Array. Perform following operations on it. 1. Insertion (Enqueue) 2. Deletion (Dequeue) 3. Display
6.	Implement binary search tree and perform the following operations: a. Insert b. Delete c. Search d. Mirror image e. Display f. Display level-wise (algorithmic analysis)
7.	Represent any real-world graph using adjacency list /adjacency matrix find minimum spanning tree using Prim's algorithm.
8.	Write a Menu driven program for a cricket player's display board. The information of the cricketer can be (not limited to) Name, Age, Country, Category (Batsman, Bowler, Wicket keeper, All-rounder), Number of ODI's played, Number of International 20-20's played. Display following. (Any four) 1. Number of batsman of a particular country 2. Sort the Batsman as per the average batting score(Any sort) 3. Batsman with highest average score (Binary search) 4. Number of bowlers of a particular country 5. The bowler that has taken a maximum no of wickets 6. Show particular players the entire "Display board information" Delete/Modify the record

Reference Books:

1. G. A.V. PAI, "Data Structures and Algorithms ", McGraw Hill, ISBN -13: 978-0-07-066726-6
2. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
3. J. Tremblay, P. Soresan, "An Introduction to Data Structures with Applications", 2nd edition, Tata McGraw Hill International Editions, 1984, ISBN-0-07-462471-7.
4. ds1-iiith.vlabs.ac.in/exp/stacks-queues/index.html
5. ds1-iiith.vlabs.ac.in/exp/poly-arithmetic/index.html

CourseSyllabus

Semester-VII

Minor in Software Development

Program:	**Minor in Software Development**			Semester: **VII**		
Course: **Database Management Systems (DBMS)**				Code: **MIT7991**		
Teaching Scheme				**Evaluation Scheme**		
Lecture	**Tutorial**	**Credit**	**Hours**	**CE**	**ETE**	**Total**
3	-	3	3	50	50	**100**

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Understand the fundamental concepts of database systems.
2. Understand the concept of database design.
3. Learn the foundation of database implementation using SQL.
4. Provide knowledge of transaction processing and concurrency control.

Outcomes:

After learning the course, the students will be able to :

1. Apply fundamental elements of database management systems.
2. Design ER-models to represent simple database application scenarios.
3. Formulate SQL queries on data for relational databases.
4. Remove the duplicate data and database anomalies using normalization.
5. Apply ACID properties for transaction management and concurrency control.
6. Analyze various emerging trends in databases.

Detailed Syllabus

Unit	Description	Duration (Hours)
1	**Introduction:** Basic concepts, Advantages of DBMS over file processing systems, Data abstraction, Database languages, Data models, Data independence, Components of a DBMS, Overall structure of DBMS, Multi-user DBMS architecture, System catalogs, Codd's Rule.	6
2	**Data Modeling**: Basic concepts, Entity, attributes, relationships, constraints, keys. **Relational Model:** The Entity-Relationship Model, Converting ER diagrams into table & EER Concepts, Data modeling constraints.	8
3	**Introduction to SQL:** Characteristics and advantages SQL Data Types, Literals, DDL, DML, SQL Operators, Views, Indexes, Nulls, Set operations, Aggregate Functions, Nested Queries, Stored Procedure, Triggers, Programmatic, Embedded SQL, ODBC.	8
4	**Relational Databases Design:** Data Redundancy and Insert, update and delete Anomalies, Functional Dependencies, Normalization.	8
5	**Transaction Management:** Basic concept of a Transaction, Properties of Transactions, Database Architecture, Concept of Schedule. **Concurrency Control**: Need, Locking methods, Dead locks, Time stamping Methods.	8

6	**Emerging Trends in databases:** NoSQL databases, Internet Databases, Cloud databases, Mobile Databases, SQLite database, XML databases.	7
	Total	**45**

Text Books:

1. AviSilberschatz, Henry F. Korth, S. Sudarshan, " Database System Concepts", Seventh Edition, McGraw-Hill,ISBN 9780078022159.
2. G. K. Gupta “Database Management Systems” , Tata McGraw Hill

Program:	**Minor in Software Development**						Semester: **VII**
Course: **Database Management System Laboratory**							Code**: MIT7992**
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Credit**	**Hours**	**TW**	**OR**	**PR**	**Total**
--	2	1	2	-	-	25	25

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Learn Entity relational diagram (ERD) design notations.
2. Create tables with primary and foreign key.
3. Learn SQL queries using arithmetic and boolean operators.
4. Learn SQL DDL/DML statements.

Outcomes:

After learning the course, the students will be able to :

1. Install and configure MySQL.
2. Study SQLite and install SQLite.
3. Design Entity relational diagram (ER/EER) diagram..
4. Implement SQL tables with primary and foreign key.
5. Execute SQL queries using arithmetic and boolean operators.
6. Execute SQL DDL/DML statements.

Detailed Syllabus:

Assignment	**Suggested List of Assignments** **Group A (Study Assignments)**
1	Install and configure client and server of MySQL.(Show all commands and necessary steps for installation and configuration).
2	Study of SQLite: What is SQLite?Uses of SQlite. Building and installing SQLite.
Assignment	**Group B (MySQL)**
3	Design any database with at least 3 entities and relationships between them. Draw suitable ER/EER diagram for the system.
4	Design and implement a database (for assignment no 1) using DDL statements and apply normalization on them.
5	Create Table with primary key and foreign key constraints. a. Alter table with add n modify b. Drop table.
6	Perform following SQL queries on the database created in assignment 1. • Implementation of relational operators in SQL • Boolean operators and pattern matching • Arithmetic operations and built in functions • Group functions • Processing Date and Time functions • Complex queries and set operators
7	Execute DDL/DML statements which demonstrate the use of views. Update the base table using its corresponding view. Also consider restrictions on updatable views and perform view creation from multiple tables.

Reference Books:

1. Ivan Bayross, “SQL, PL/SQL: The Programming Language of Oracle”, BPB Publication, ISBN10: 8176560723; ISBN-13: 978-8176560726

Program:	**Minor in Software Development**				Semester: **VII**	
Course: Software Engineering					Code:**MIT7993**	
Teaching Scheme				**Evaluation Scheme**		
Lecture	**Tutorial**	**Credit**	**Hours**	**CE**	**ETE**	**Total**
3	-	3	3	50	50	**100**

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Understand the software engineering fundamentals.
2. Introduce principles of agile software development, the SCRUM process, and agile practices.
3. Understand project management through the life cycle of the project.
4. Understand current and future trends and practices in the IT industry.

Outcomes:

After learning the course, the students will be able to:

1. Apply software engineering principles to develop software.
2. Analyze software requirements and formulate design solutions for a software.
3. Explain concepts of design engineering , planning and scheduling.
4. Understand risk management and software configuration management.
5. Illustrate software testing strategies.

Detailed Syllabus:

Unit	Description	Duration (Hours)
1	**Software Engineering Fundamentals**: The Nature of Software, Defining Software, Software Engineering Practice. Software Process: A Generic Process Model, defining a Framework Activity, Identifying a Task Set. **Process Models:** Software Process Models, Evolutionary Process Models, Concurrent Model. **Agile software development**: Agile manifesto, agility principles, Agile methods agile development, SCRUM.	6
2	**Requirements Engineering:** User and system requirements, Functional and non-functional requirements, requirements engineering (elicitation, specification, validation, negotiation). **Software Requirements Specification (SRS):** software requirements Specification document, structure of SRS, writing a SRS, structured SRS for online shopping. **Requirements Analysis**: Analysis Model, data modeling, scenario based modeling, class based modeling, Flow oriented modeling, behavioral modeling-Introduction to UML diagrams.	8

3	**System modeling:** Interaction models, Structural models, Behavioral models, Model-driven engineering. **Design Engineering** : Design Process & quality, Design Concepts, design Model, Pattern-based Software Design. **Design and implementation:** Object-oriented design using the UML.	8
4	**Project Planning:** Project initiation, Creating the Work Breakdown Structure. **Scheduling:** Importance of Project Schedules, Developing the Schedule using Gantt Charts, PERT/ CPM. **Project Management:** The Management Spectrum, People, Product, Process, Project, The W5HH Principle, Metrics in the Process and Project Domains, Metrics for Project.	8
5	**Quality Concepts:** Quality, software quality, Quality Metrics, achieving software quality. **Software Testing:** Principles of Testing, Test plan, Test case, Types of Testing, Verification & Validation, Testing strategies, Defect Management, Defect Life Cycle.	8
6	**Project Estimation:** Software Project Estimation, Decomposition Techniques, Cost Estimation Tools and Techniques., Typical Problems with IT Cost Estimates. Case Study : CASE Tools.	**7**
Total		**45**

Text Books:

1. Ian Sommerville, 'Software Engineering', Addison-Wesley, 9th Edition, 2010, ISBN-13: 978-0137035151.
2. Roger Pressman, "Software Engineering:A Practitioner's Approach", McGraw Hill,ISBN 0-07- 337597-7.

Reference Books:

1. Marchewka, "Information Technology Project Management",Willey India, ISBN: 9788-1265- 4394-6

CourseSyllabus

Semester-VIII

Minor in Software Development

Program:	**Minor in Software Development**			Semester: **VIII**			
Course: Project				Code: **MIT8991**			
Teaching Scheme				**Evaluation Scheme**			
Practical	**Tutorial**	**Credit**	**Hours**	**TW**	**OR**	**PR**	**Total**
8	-	4	8	50	50	-	**100**

Objectives:

After completion of the course, students will have adequate background, conceptual clarity and knowledge related to:

1. Develop problem solving ability using core computing concepts.
2. Understand basics of IT Project management
3. Develop critical thinking for solving problems using core knowledge of Information Technology.
4. Emphasizes learning activities that are long-term, inter-disciplinary and student-centric.
5. Work in TEAM and learn professionalism.

Outcomes:

After learning the course, the students will be able to:

1. Develop projects applying object-oriented concepts.
2. Practice acquired knowledge within the chosen area of technology for project development.
3. Organize and Maintain large collection of related data.
4. Organize information in digital space.

General Guidelines for Project Work

Project is one of the significant contributory team works that has to be completed based on the required number of credits as per academic regulations. It is necessary to explore the domain of interest / research/ thrust area/ society needs.

1. **Project teams:** 3-4 students can form a team within the same or different discipline and their area of interest is to be registered with project Coordinator.
2. **Selection of Project**: a. Student shall identify the area or topics based on the knowledge they gain from all the courses offered to them.
3. **Guide Allotment:**
   a. Considering registered teams area of interest/domain and expertise of guide, the Project coordinator in consultation with panel of experts allots Project guides.
   b. Guide should be allotted from the same program.
4. **Project Review:**
   a. The Project coordinator with the Head of the department shall constitute a review committee for project groups.

b. A review committee will approve the project group and title.

c. The Project Review committee will evaluate the timely progress of the projects.

6. Report:

a. Report should be prepared using Word/Latex only as per the template provided by the department.

b. Project reports shall be submitted in softcopy and hardcopy form (Hard bound reports could be avoided).

Project Presentation

The evaluation will be based on the following criteria:

1. Depth of knowledge and skills
2. Communication & presentation Skills
3. Team work and creativity
4. Planning & organizational skills
5. Societal understanding and Ethics
6. Regularity and punctuality
7. Attendance record

Project Work Evaluation:

Review 1: -

Deliverables:

1. The precise problem statement/title based on literature survey and feasibility study.
2. Purpose, objectives and scope of the project.
3. List of required hardware, software or other equipment for executing the project, test Environment/tools, cost and human efforts in hours.
4. System overview- proposed system and proposed outcomes.
5. Architecture and initial phase of design (DFD).
6. Project plan 1.0.

Review 2: –

Deliverables:

1. SRS and High level design
2. Detail architecture/System design/algorithms/techniques
3. Complete Project and Testing
4. Project plan 2.0

Report

Project reports shall be submitted in softcopy form/ (Hard bound reports could be avoided).

The project report contains the details.

1. Problem definition
2. Requirement specification
3. System design details (UML diagrams)
4. System implementation – code documentation – dataflow diagrams/ algorithm, protocols used.
5. Test result and procedure
6. Conclusions.
7. Appendix

a. Tools used
b. References
c. Papers published/certificates
d. Plagiarism Report of paper and project report from any open source tool

Credit distributions

1. TW- Review, presentation Report, project competitions, and deliverables
2. Oral- presentation, project execution [with external examiner

Vision and Mission of Information Technology Department

Department Vision
To develop Information Technology professionals through Quality Education with dedicated faculty.

Department Mission
Foster the development of Information Technology professionals with focus on Excellent Academics, Research Aptitude, Overall Personality Development and Social Awareness.