Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology
(Autonomous Institute)

Faculty of Science and Technology

Board of Studies

Information Technology

Syllabus

Double Minor Courses

(2023 Pattern)
(w.e.f. AY: 2024-25)

Syllabus: Double Minor Information Technology
w. e. f. AY:2024-2025

Double Minor in Full Stack Development

SEM	Course Code	Courses Name	Teaching Scheme			Examination Scheme and Marks							Credits			
			TH	PR	TUT	Activity	ISE	ESE	TW	PR	OR	Total	TH	PR	TUT	Total
III	IT23262TH	Programming in Java	2	-	-	10	-	60	-	-	-	70	2	-	-	3
	IT23262PR	Programming in Java	-	2	-	-	-	-	30	-	-	30	-	1	-	
IV	IT23272TH	Database Management Systems	2	-	-	10	-	60	-	-	-	70	2	-	-	3
	IT23272PR	Database Management Systems	-	2	-	-	-	-	30	-	-	30	-	1	-	
V	IT23362TH	Fundamentals Web Development	3	-	-	10	30	60	-	-	-	100	3	-	-	4
	IT23362PR	Fundamentals Web Development	-	2	-	-	-	-	30	-	-	30	-	1	-	
VI	IT23372TH	Introduction to UI and UX Design	3	-	-	10	30	60	-	-	-	100	3	-	-	4
	IT23372PR	Introduction to UI and UX Design	-	2	-	-	-	-	30	-	-	30	-	1	-	
VIII	IT23472TH	Software Testing	3	-	-	10	30	60	-	-	-	100	3	-	-	4
	IT23472PR	Software Testing	-	2	-	-	-	-	30	-	-	30	-	1	-	
Total			**13**	**10**	**0**	**50**	**90**	**300**	**150**	**0**	**0**	**590**	**13**	**5**	**0**	**18**

Prof.S.A.Takale
HOD-IT

Dr.A.H. Kolekar
CoE

Prof.S.M.Bhosle
Dean Academics

Prof.S.B.Lande
Principal

Bucket of DOUBLE MINOR DEGREE

DOUBLE MINOR DEGREE **(only for students having CGPA >= 7.5)**
Double Minor: Artificial Intelligence and Data Science
Double Minor: Cloud Computing and Virtualization
Double Minor: Full Stack Development
Double Minor: Embedded Systems and Real-Time OS
Double Minor: Municipal or Urban Engineering
Double Minor: Enterprise Resource Planning
Double Minor: Digital Mfg. and Robotics
Double Minor: Renewable Energy

Programming In Java

Course Code: IT23262	Course Credits: 03	Course type: DM

Teaching Scheme			Evaluation Scheme					
TH	**PR**	**TUT**	**ACTIVITY**	**ISE**	**ESE**	**TW**	**PR**	**OR**
02	**02**	**–**	**10**		**60**	**30**		**-**

Prerequisite Course Mapping:
C, C++ programming

Future Course Mapping:
Advanced Java Programming

Importance of Course: Software industries using java as a pure object oriented programming language for development of the various software. Java provides a secure and platform independent way for developing various applications which are useful in real life for all the people working in different areas.

Course Objectives:
To learn how to implement object-oriented designs with Java.
To learn how to extend Java classes with inheritance and dynamic binding.
To understand how to design applications with threads in Java.

Course Outcomes:
Understand implementation of object-oriented designs with Java.
Develop reusable programs using Inheritance and polymorphism.
Understand the use of abstraction and encapsulation.
Apply the concepts of strings, threading to develop efficient real world applications.

UNIT No.	Syllabus	Hrs.
I	**Fundamentals of Java Programming** Basics of Java, Application, Control statements: conditional statement, looping statement, break, continue, array Java OOPs Concepts, object and class, methods, Access Modifiers in Java, constructor, static and this keyword	6
II	**Inheritance and Polymorphism** **Inheritance:** need of inheritance, types of inheritance, aggregation **Polymorphism :** Method overloading, method overriding, super and final keyword, types of polymorphism	6
III	**Java Abstraction and Encapsulation** Abstract class, interface Package, access modifier, encapsulation	6

IV	**Java string and Multithreading** **Java string:** basics of string, different string methods **Multithreading**: multithreading, advantages of multithreading, multitasking, what is thread, thread class, java thread methods, life cycle of thread, creating thread, sleeping thread, calling run method	6

List of Practical Assignments:

1. Write a java program to print even numbers between 1 to 500
2. Write a java program to print 5 toppers among 10 students.
3. Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.
4. Write a Java program to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, override the calculateSalary() method to calculate and return the salary based on their specific roles.
5. Write a Java program to create an abstract class Animal with abstract methods eat() and sleep(). Create subclasses Lion, Tiger, and Deer that extend the Animal class and implement the eat() and sleep() methods differently based on their specific behavior.
6. Write a Java program to create a class called Person with private instance variables name, age. and country. Provide public getter and setter methods to access and modify these variables.
7. Write a Java program to print the current date and time in the specified format.
8. Write a Java program that creates two threads to find and print even and odd numbers from 1 to 20.

Text Books:

E. Balaguruswamy, “Programming with JAVA: A Primer” McGraw Hill Education, India, 5th Edition.
“Herbert Schildt, “Java: The complete reference”, Tata McGraw Hill, 7th Edition.

Reference Books:

. Y. Daniel Liang (2010), “Introduction to Java programming”, Pearson Education, India, 7th Edition.
. T. Budd, “Understanding OOP with Java”, Pearson Education, 2nd Updated Edition.
. Cay Horstmann, “Core Java Volume 1”, Kindle, 11th Edition.

Online Resources:

NPTEL Course :
1.https://archive.nptel.ac.in/courses/106/105/106105185/#
2. https://archive.nptel.ac.in/courses/106/105/106105163/
Virtual Lab:
simulator Link http://vlabs.iitkgp.ac.in/coa/

Database Management System

Course Code : IT23272	Course Credits: 03	Course type: DM

Teaching Scheme			Evaluation Scheme					
TH	PR	TUT	ACTIVITY	ISE	ESE	TW	PR	OR
02	02	–	10	-	60	30	-	-

Prerequisite Course Mapping:

1. Programming in Java

Future Course Mapping:

1. Advanced Database Management System
2. Data Science and Big Data Analytics

Importance of Course: In software projects to store data, databases are used. The knowledge of DBMS will provide ways of storing and handling these databases .

Course Objectives:

1. To realize the fundamental concepts of Database Systems
2. To discover the various RDBMS Tools
3. To absorb different PL-SQL Types
4. To grow up on recent Database Trends

Course Outcomes:

1. **Understand** fundamental elements of database management systems
2. **Understand and remember** PL/SQL queries.
3. **Formulate** SQL queries on data for relational databases
4. **Apply** ACID properties for transaction management and concurrency control

Unit No.	Syllabus	Hrs.
1	**Introduction to DBMS** 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, ER Diagram: Basic concepts, Entity, attributes, relationships, constraints, keys, EER Diagram-Basic concepts ,characteristics	6
2	**Introduction to SQL and PL/SQL** **SQL:** 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	6

	and Joins, Set membership, Tuple Variables, Set comparison, Ordering of Tuples, Aggregate Functions, SQL Functions, Nested Queries. Web application development using Database connectivity	
3	**Introduction to Relational Database and Query Processing** **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.	6
4	**Database Transaction Management** 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	6

PRACTICAL ASSIGNMENTS

Assign No	Title
1	Study the installation and uninstallation of MySQL and SQLite
2	Choose a database application. Perform requirement analysis in detail for the same. Draw an entity-relationship diagram for the proposed database.
3	**SQL Queries:** Create a database using SQL DDL statements.
4	**Implementing Data Constraints in MySQL** Implement sql queries to provide students with hands-on experience in implementing various data constraints using SQL commands in MySQL. (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, and DEFAULT)
5	**DML Statements :** Populate/modify database applications with the help of DML. i.e using SQL DML Statements. Demonstrate the use of concepts like Insert, Select, Update, Delete with operators, functions, and set operator etc
6	**Exploring SQL Computations done on Table Data and Built-in Functions in MySQL** Implement sql queries for different SQL concepts for SQL computation (arithmetic operators, logical operators, pattern matching, IN and NOT IN predicates, and MySQL built-in functions).
7	**Exploring SQL Grouping, Database Views** Implement SQL queries for different SQL concepts such as grouping data, using the GROUP BY clause, employing the HAVING clause, applying the EXISTS/NOT EXISTS operators,

	Creating and using Database Views.
8	**SQL Queries**: Write equijoin, non equijoin, self join and outer join queries for suitable database applications using SQL DML statements.
9	**Nested Queries** Implement SQL queries to demonstrate the nested queries in SQL using MySql.
10	Write PL/SQL blocks to implement all types of cursor.

Text Books :

1."Database System Concepts" by Silberschatz A., Korth H., Sudarshan S. , 6th edition
2.Data Mining :Concepts and Techniques ,Jiawei Han, Micheline Kamber, Jian Pei

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. Joy A. Kreibich, "Using SQLite", O'REILLY, ISBN: 13:978-93-5110-934-1
4. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications ISBN: 9788176569644, 9788176569644
5. Reema Thareja, Data warehousing, Oxford University Press. ISBN 0195699610

Online Resources:

1.NPTEL Course "**Introduction to DBMS**"
https://nptel.ac.in/courses/106/105/106105175/

Fundamentals Web Development								
Course Code: IT23362			**Course Credits: 04**			**Course type: Double Minor**		
Teaching Scheme			**Evaluation Scheme**					
TH	**PR**	**TUT**	**ACTIVITY**	**ISE**	**ESE**	**TW**	**PR**	**OR**
3	**2**	**–**	**10**	**30**	**60**	**30**	**-**	**-**

Prerequisite Course Mapping:

1. C++, Java.

Future Course Mapping:

1. Computer Network, Distributed System.

Course Objectives:

1. To familiarize students with Web Programming basic concepts.
2. To learn and understand Web scripting languages.
3. To explore the Front end & Backend web programming skills.
4. To understand and learn Mobile web development.
5. To understand and learn Web application deployment.

Course Outcomes:

1. Understand & Develop Static and Dynamic website using technologies like HTML, CSS, Bootstrap.
2. Demonstrate the use of web scripting languages.
3. Understand the basics of Front End Technologies.
4. Develop web application with Front End & Back End Technologies.
5. Develop mobile website using JQuery Mobile.
6. Deploy web application on cloud using AWS.

Unit No.	Syllabus	Hrs.
I	**INTRODUCTION TO WEB TECHNOLOGIES** **HTML:** Getting started with HTML, Why HTML, Tags and Elements, Attributes, Properties, Headings list, Links, Tables, Images, HTML Form, Media (Audio, Video), Semantic HTML5 Elements. **CSS:** Why CSS, Types of CSS, how to use CSS, Properties, Classes, Child-Class (Nested CSS), Colors, Text, Background, Border, Margin, Padding, Positioning (flex, grid, inline, block), Animation, Transition. **BOOTSTRAP:** Why Bootstrap, CSS over Bootstrap, how to Use Bootstrap, Bootstrap Grid System, Bootstrap Responsive, Bootstrap Classes, Bootstrap Components (i.e., Button, Table, List, etc.), Bootstrap as a Cross Platform. **W3C:** What is W3C, How W3C handles/Supports Web Technologies.	7
II	**WEB SCRIPTING LANGUAGES** **JavaScript:** Introduction to Scripting languages, Introduction to JavaScript (JS), JS Variables and Constants, JS Variable Scopes, JS Data Types, JS Functions, JS Array, JS Object, JS Events. **Advanced JavaScript:** JSON - JSON Create, Key-Value Pair, JSON Access, JSON Array, JS Arrow Functions, JS Callback Functions, JS Promises, JS Async-Await Functions, JS Error Handling. **AJAX:** Why AJAX, Call HTTP Methods Using AJAX, Data Sending, Data Receiving, AJAX Error Handling. **JQUERY :**Why JQuery, How to Use, DOM Manipulation with JQuery, Dynamic Content Change with JQuery, UI Design Using JQuery	7
III	**FRONT END TECHNOLOGIES** **Front-End Frameworks:** What is web framework? Why Web Framework? Web Framework Types.	6

	MVC: What is MVC, MVC Architecture, MVC in Practical, MVC in Web Frameworks. **TypeScript:** Introduction to TypeScript (TS), Variables and Constants, Modules in TS. **AngularVersion 10+:** Angular CLI, Angular Architecture, Angular Project Structure, Angular Lifecycle, Angular Modules, Angular Components, Angular Data Binding, Directives and Pipes, Angular Services and Dependency Injections (DI), Angular Routers, Angular Forms. **ReactJS:** Introduction to ReactJS, React Components, Inter Components Communication, Components Styling, Routing, Redux- Architecture, Hooks- Basic hooks, useState() hook, useEffect() hook useContext() hook.	
IV	**BACK END TECHNOLOGIES** **Node.JS:** Introduction to Node.JS, Environment Setup, Node.JS Events, Node.JS Functions, Node.JS Built- in Modules, File System, NPM, Install External Modules, Handling Data I/O in Node.JS, Create HTTP Server, Create Socket Server, Microservices- PM2. **ExpressJS:** Introduction to ExpressJS, Configure Routes, Template Engines, ExpressJS as Middleware, Serving Static Files, REST HTTP Method APIs, Applying Basic HTTP Authentication, Implement Session Authentication. **MongoDB:** NoSQL and MongoDB Basics, MongoDB-Node.JS Communication, CRUD Operations using Node.JS, Mongoose ODM for Middleware, Advanced MongoDB.	7
V	**MOBILE WEB DEVELOPMENT** **Mobile-First:** What is Mobile-First? What is Mobile Web? Understanding Mobile Devices and Desktop. **JQuery Mobile:** Introduction to the jQuery Mobile Framework, Set-up jQuery Mobile, Pages, Icons, Transitions, Layouts Widgets, Events, Forms, Themes, Formatting Lists, Header and Footer, CSS Classes, Data Attributes, Building a Simple Mobile Webpage.	6
VI	**WEB APPLICATION DEPLOYMENT** **Cloud:** AWS Cloud, AWS Elastic Compute, AWS Elastic Load Balancer and its types, AWS VPC and Component of VPC, AWS storage, Deploy Website or Web Application on AWS, Launch an Application with AWS Elastic Beanstalk.	6

List of Practical Assignments:

1. Create a responsive web page which shows the ecommerce/college/exam admin dashboard with sidebar and statistics in cards using HTML, CSS and Bootstrap.
2. Write a JavaScript Program to get the user registration data and push to array/local storage with AJAX POST method and data list in new page.
3. Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component.
4. Create a Node.js application that serves a static website and implements four CRUD APIs using Express.js and MongoDB.
5. Create a simple Mobile Website using jQuery Mobile.
6. Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.

Text Books:

1. Kogent Learning Solutions Inc, Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Blackbook, Dreamtech Press, Second Edition, ISBN: 9788177228496.
2. Raymond Camden, Andy Matthews, JQuery Mobile Web Development Essentials, Packet Publishing, Second Edition, 9781782167891.

Reference Books:

1. Steven M. Schafer, "HTML, XHTML and CSS", Wiley India Edition, Fourth Edition,978- 81- 265-1635-3.
2. Steven M. Schafer, "HTML, XHTML and CSS", Wiley India Edition, Fourth Edition,978- 81-265-

1635-3.
3. Ivan Bayross, “Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP, BPB Publications,4th Edition, ISBN:978-8183330084.
4. Adam Bretz& Colin J Ihrig, Full Stack Javascript Development with MEAN, SPD, First Edition, ISBN:978-0992461256.
5. JavaScript: The Definitive Guide - Master The World's Most-Used Programming Language, Seventh Edition.
6. Programming Typescript: Making Your JavaScript Applications Scale, Boris Cherny.

Online Resources:

NPTEL Course:

1. https://archive.nptel.ac.in/courses/106/106/106106222/
2. https://archive.nptel.ac.in/courses/106/106/106106156/

Introduction to UI and UX Design

Course Code : IT23372	Course Credits: 04	Course type: Double Minor

Teaching Scheme			Evaluation Scheme					
TH	**PR**	**TUT**	**CAA**	**ISE**	**ESE**	**TW**	**PR**	**OR**
3	**2**	**-**	**10**	**30**	**60**	**30**	**-**	**-**

Prerequisite Course Mapping:
1.Basics of Web Technology
2.Basic Design Concepts

Future Course Mapping:
1.Web Design and Development
2.Mobile Application Design

Course Objective:

1. Introduce the fundamental concepts of User Interface (UI) and User Experience (UX) design and their role in developing effective digital products.
2. Develop an understanding of user-centered design principles and the importance of understanding user needs through research methods such as interviews, surveys, and personas.
3. Familiarize students with information architecture, wireframing, and prototyping techniques used in designing websites and mobile applications.
4. Enable students to apply visual design principles including layout, typography, color theory, and interaction design for creating intuitive interfaces.
5. Provide knowledge of usability testing, accessibility standards, and UX evaluation methods for improving the usability and effectiveness of digital interfaces.
6. Introduce students to industry-standard UI/UX design tools used for prototyping, collaboration, and design presentation.

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

1. **Understand** the fundamental concepts of UI and UX design and their importance in digital products.
2. **Apply** user research methods such as interviews, surveys, personas, and journey mapping to understand user needs.
3. **Design** effective information architecture, wireframes, and user flows for websites and applications.
4. **Develop** UI designs and interactive prototypes using standard design tools.
5. **Evaluate** usability and accessibility of interfaces using testing methods and UX evaluation techniques.

UNIT No.	Syllabus	Hrs
I	**Introduction to UI/UX Design** Definition of User Interface (UI) and User Experience (UX) Design, Importance of UI/UX in the digital world, Difference between UI and UX, Real-world examples of good and bad UI/UX, Introduction to User-Centered Design principles, Overview of the UX Design Process, Basic design principles (Alignment, Contrast, Repetition, Proximity, Balance), Introduction to Color Theory and Typography.	6
II	**User Research and Visual Design** Understanding users and user-centered research, User personas and empathy in design, User research methods: interviews, surveys and questionnaires, User journey mapping: touchpoints and pain points, Competitive analysis: SWOT and benchmarking, Introduction to visual design principles (balance, hierarchy, contrast), Visual identity and branding.	6

III	**Information Architecture and UX Design Process** Design Thinking framework: Empathize, Define, Ideate, Prototype, Test, Information architecture: content organization and categorization, Navigation systems for websites and applications, Card sorting and sitemap creation, User scenarios and user flows, Introduction to wire framing and low-fidelity prototypes, Overview of design tools (e.g., Figma).	7
IV	**User Interface (UI) and Interaction Design** Visual design principles: rule of thirds, grid systems, negative space, UI components: buttons, forms, navigation bars and icons, Interaction design principles: affordance, feedback and consistency, Responsive design: mobile-first approach and breakpoints, Micro-interactions and basic animations, Introduction to UI design tools: Figma, Sketch, Adobe XD.	7
V	**Prototyping and User Testing** Prototyping and testing using UI/UX tools, Usability testing and collecting user feedback, Iterating and improving designs based on testing results, Developing a basic style guide for visual consistency, Design for different platforms: web and mobile interfaces, Touch-based interaction design, Design presentation techniques and portfolio development.	7
VI	**Usability, Accessibility and UX Testing** Usability principles and heuristic evaluation, Usability evaluation techniques, Accessibility in UI/UX design, Accessibility standards (WCAG guidelines), UX testing methods: A/B testing, usability testing and heatmaps, Feedback collection and iterative design validation.	6
Total Teaching Hours		**39**

List of Practical Assignments:

1. Study and analyze the User Interface (UI) and User Experience (UX) of two different websites or mobile applications. Identify good and bad design practices based on usability, navigation, layout, and visual design principles.
2. Design and conduct a user research survey for a selected application (e.g., e-learning, food delivery, or shopping app). Collect responses and analyze user needs, expectations, and common usability problems.
3. Develop user personas for a chosen application based on collected user research data. Include user demographics, goals, frustrations, and behavioural patterns.
4. Create a User Journey Map for a selected website or mobile application. Identify user touchpoints, actions, pain points, and opportunities for improving the overall user experience.
5. Design the Information Architecture for a selected website or application by creating a sitemap and organizing content using card sorting techniques.
6. Design low-fidelity wireframes for the main screens of a selected application such as login page, dashboard, and navigation page using paper sketches or UI/UX design tools.
7. Develop a high-fidelity user interface for selected screens of an application by applying visual design principles such as typography, color theory, grid systems, and visual hierarchy using UI design tools.
8. Create an interactive prototype for a selected application by linking multiple screens and demonstrating user interaction using prototyping tools.
9. Perform usability testing for the designed prototype with a group of users, collect feedback, apply heuristic evaluation, and suggest improvements for better usability and accessibility**.**

Text Books:

1. UX for Beginners: A Crash Course in 100 Short Lessons – Joel Marsh, O'Reilly Media.
2. The Design of Everyday Things – Don Norman, Basic Books.
3. The Elements of User Experience: User-Centered Design for the Web and Beyond – Jesse James Garrett, New Riders.

Reference Books:

1. Don't Make Me Think – Steve Krug, New Riders Press.
2. Designing Interfaces – Jenifer Tidwell, Charles Brewer, Aynne Valencia, O'Reilly Media.
3. Lean UX – Jeff Gothelf and Josh Seiden, O'Reilly Media.
4. 100 Things Every Designer Needs to Know About People – Susan Weinschenk, New Riders.
5. Universal Principles of Design – William Lidwell, Kritina Holden, Jill Butler.

Online Resources:

https://www.coursera.org/specializations/ui-ux-design
https://nptel.ac.in/courses/106106173
https://www.interaction-design.org
https://www.nngroup.com

Software Testing

Course Code : IT23472	Course Credits: 04	Course type: DM

Teaching Scheme			Evaluation Scheme					
TH	PR	TUT	CAA	ISE	ESE	TW	PR	OR
3	2	-	10	30	60	30	-	-

Prerequisite Course Mapping:
1. Software Engineering

Future Course Mapping:
1. Software Project Management

Course Objective:
1. To understand fundamental concepts of software testing, process and methods.
2. To understand designing of test project, test cases, test data, test environment for conducting tests.
3. A keen awareness of the software testing solutions.
4. To learn test life cycle and defect life cycle.
5. To learn in detail about various quality assurance models.
6. To understand the automated testing tool to achieve quality.

Course Outcomes: On completion of the course, learner will be able to
1. Apply fundamental concepts of software testing, process and methods.
2. Design test project, test cases, test data, test environment for conducting tests.
3. Propose software testing solutions.
4. Generate test report by making use of manual testing techniques
5. Devise solutions towards defects solving and software failures by making use of object- oriented and component-based software testing methods.
6. Understand the use of automated testing tool in software testing to generate reports.

Syllabus

UNIT No.	Syllabus	Hrs
I	**Introduction to Software Testing** Definition, Need of testing, Testing Principles, Basic concepts – errors, faults, defects, Failures, Verification and validation activities , Functional testing, Non functional testing, Levels of testing, Types of Testing, White box testing, Black box testing, continuous integration and continuous deployment (CICD),Regression testing, Acceptance testing, smoke and sanity testing, , Performance testing, Recovery testing, Application of statistics and probability distribution in Testing.	7
II	**Black -box testing Techniques:** Need of black box testing, Black box testing Concept, Requirement Analysis, Test case design criteria, Testing methods, Requirement based testing, Positive & negative testing, Boundary value analysis, Equivalence Partitioning, Decision table testing, State transition testing, Use case testing, Cause effect graph based, Error guessing, Documentation testing & domain testing, Design of test cases.	7
III	**White-box testing Techniques:** Need of white box testing, Testing types, Test adequacy criteria, Static white-box testing, Dynamic white-box testing, Structure - logic coverage criteria, Basis path testing, Graph metrics, Loop Testing, Data flow testing, Mutation Testing, Design of test cases, Testing of Object oriented systems, Challenges in White box testing.	7

IV	**Test Management:** Testing life cycle – Roles and activities, Test Planning –Master Test Plan-test environment, test suite, test bed, test data, test schedule, forming a test team, Develop test plan review, Test Cases design strategies, Test adequacy criteria, Coverage and control flow graphs, Paths, Loop testing, Test execution, Life cycle of defect, Defect tracking, Defect detection stages, defect types, defect severity, defect analysis and prevention.	6
V	**Quality Assurance:** Quality concepts – quality, quality control, quality assurance, difference between QC ad QA, Software quality assurance – SQA activities, Software reviews, Inspections, Audits, Software quality attributes - Correctness, reliability, usability, maintainability, interoperability, Ishikawa's seven quality tools, six sigma.	6
VI	**Testing Tools:** Manual testing, manual testing Vs automated testing, automation frameworks, Automated testing tools and case studies, Study of testing tools –Test Execution tool, Selenium basics , Selenium IDE, Web-driver, Test NG, Postman, Performance, Scalability and Reliability Testing, Case studies on web based and GUI testing.	6
	Total Teaching Hours	**39**

List of Practical Assignments:

1) Design test cases for a Student Registration Form.
2) Write test cases in excel sheet for Social media application, website or web application. Develop a defect report after testing.
3) Perform White Box Testing.(Statement Coverage, Branch Coverage, Path Coverage, Flow Graph Drawing)
4) Perform automation testing using selenium-Install Selenium IDE / WebDriver. Write first Selenium Script. Record and Play Test Cases. Use Locators (ID, Name, XPath, CSS). Automate Login Form
5) Perform automation testing using TestNG & its Framework. Write TestNG test scripts. Using annotations generating test reports and perform Data-Driven testing.
6) Test any existing website application (IRCTC, Online Shopping Site, Airline Reservation Site) using any automated testing tool.

Text Books:

4. Srinivasan Desikan, Golalaswamy Ramesh, "Software Testing Principles and Practices" Pearson edition.
5. "Software Testing Techniques",Beizer B. Van Nostrand Reinhold (1990) 2nd edition, ISBN 0442-20672-0
6. "Managing the Testing Process ",Black, R. (2001) (3rd edition), John Wiley & Sons New York, ISBN: 978-0-470-40415-7.
7. "Software Engineering ",Ian Sommerville , 9th Edition, Pearson, Boston, ISBN-13: 978-0-13- 703515-1.

Reference Books:

1. Aditya P Mathur, "Foundations of Software Testing", Pearson.
2. Paul Ammann, Jeff Affutt, "Introduction to Software Testing", Cambridge University Press.
3. Renu Rajani, Pradeep Oak, "Software Testing-Effective Methods,Tool and Techniques", Tata McGraw Hills.
4. Stephan Kan, "Metrics and Models in Software Quality" Addison Wesley, 2nd Edition.
5. "Lessons Learned in Software Testing ",Kaner, C., Bach, J. and Petttico B. (2002), John Willey & Sons: New York, ISBN-13: 978-0471081128.
6. "The Art of Software Testing Myers", Glenford J. (1979), John Wiley & Sons: New York, ISBN13: 978-1118031964.
7. Daniel Galin, "Software Quality Assurance:From theory to implementation" , Pearson Addison Wesley.

Online Resources: 1. **https://onlinecourses.nptel.ac.in/noc26_cs59/preview**