Savitribai Phule Pune University (Formerly University of Pune)

Four Year Degree Program
B.Sc.(Computer Science)

With

Major: Computer Science

(Faculty of Science and Technology)

Syllabi for
T.Y.B.Sc. (Computer Science)

(For Colleges Affiliated to Savitribai Phule Pune University)

Choice Based Credit System (CBCS) Syllabus Under National Education Policy (NEP)

To be implemented from Academic Year 2026-2027

Title of the Course: B.Sc. (Computer Science)

Preamble:
The B. Sc. (Computer Science) and B. Sc. (Computer Science) (Honors) and (Research) course is a systematically designed program with Computer Science as a major subject under the faculty of Science and Technology. The objective of the course is to prepare students to undertake careers involving problem solving using computer science and technologies, or to pursue advanced studies and research in computer science. The syllabus which comprises of Computer Science (Major) subject along with that of the three allied subjects (Mathematics, Electronics and Statistics) (Minor) covers the foundational aspects of computing sciences and also develops the requisite professional skills and problem solving abilities using computing sciences.

Introduction:
At the first year of under-graduation, the basic foundations of two important skills required for software development are laid. A course in problem solving and programming along with a course in database fundamentals forms the preliminary skill set for solving computational problems. The practical courses are designed to supplement the theoretical training in the year. Along with Computer Science (Major), VSC and SEC courses help in building a strong technical foundation. Another aspect of this course is IKS which tells about the rich heritage and advancement of India in the field of computation.
In the second year of under-graduation, computational problem solving skills are further strengthened by a course in Data structures, C++ and python programming. Software engineering concepts that are required for project design are also introduced. Essential concepts of computer networking are also introduced this year. The practical course included in both semesters complements the theory courses. Field projects/ OJT are introduced so that students can implement the concept they have learnt in first year.
In Second Year, the "Subject 1: Computer Science" will be the Major Subject and the Minor subject will be chosen from "Subject 2 or Subject 3". Subject 2 and Subject 3 will not be available as Major Subjects in Second Year and Third Year

At the third year of under-graduation, all the subjects are designed to fulfill core Computer Science requirements as well as meet the needs of the software industry. Theory courses are adequately supplemented by hands-on practical courses. Major elective courses are taking care of recent advancement in the field of computer science. Minor and Skill Enhancement courses enable the students to acquire additional skills.
At the fourth year (honors) and (research) of under-graduation, all the subjects are designed to fulfill core Computer Science requirements as well as meet the needs of the software industry. Practical courses and field projects enable students to get hands-on training. Various learning tracks are open through Major elective courses. Research methodology course will create interest among the students to carry research in the field of computer science.

Objectives:

- To develop problem solving abilities using a computer.
- To build the necessary skill set and analytical abilities for developing computer based solutions for real life problems.
- To train students in professional skills related to the Software Industry.
- To prepare the necessary knowledge base for research and development in Computer Science.
- To help student's build-up a successful career in Computer Science and to produce entrepreneurs who can innovate and develop software products.

Eligibility

a) H.S.C. (10 + 2) Science stream with Mathematics.

b) Three years diploma course after S.S.C. (10^{th} std.) of Board of Technical Education conducted by Government of Maharashtra or its equivalent.

Programme Out comes:

PO No.	**Outcomes**
PO1	Develop creative skills, critical thinking, analytical skills and research to address the real world problems using computational skills
PO2	Understand and apply mathematical foundation, computing and domain knowledge and develop computing models for defined problems
PO3	Understand software project management and computing principles with computing knowledge to manage projects in multidisciplinary environments
PO4	Illustrate the concepts of systems fundamentals, including architectures and organization, operating systems, networking and communication
PO5	Understand and apply the concepts of Digital Electronics, Computer Architecture, IoT etc.
PO6	Recognize the need for and develop the ability to engage in continuous learning as a Computing professional
PO7	Apply modern computing tools, skills and techniques necessary for innovative software solutions
PO8	Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations
PO9	Gain Self Discipline and commit Professional Ethics in global economic environment
PO10	Individual & Team Work: Ability to work as a member or leader in diverse teams in multidisciplinary environment
PO11	Identify opportunities, entrepreneurship vision and use innovative ideas to create value and wealth for the betterment of the individual and society

Savitribai Phule Pune University

Structure of UG Program as per NEP-2020

Name of Program :- B.Sc. (Computer Science)

Major Course: - Computer Science

Level:- 4.5 (First Year) **Sem:-I**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Subject 1	CS-101-T	Problem Solving using 'C' Programming	2		2		20	30	50
	CS-102-P	Lab Course based on CS-101-T		2		4	20	30	50
Subject 2	MTC-101-T	Matrix Algebra	2		2		20	30	50
	MTC-102-P	Mathematics Practical I		2		4	20	30	50
Subject 3	ELC-101-T	Principles of Analog Electronics	2		2		20	30	50
	ELC-102-P	Electronics Practical Course I		2		4	20	30	50
IKS(2)	IKS-101-T	Generic IKS	2		2		20	30	50
GE/OE*(2)	OE-101-CS -T/ OE-102-CS -T/ OE-103-CS-T / OE-104-CS-T	Office Automation I / Introduction to Computers and Basics of Internet / Introduction to Google Apps I / Fundamentals of Computers I	2		2		20	30	50
SEC (2)	SEC-101-CS	Statistical Methods for Computer Science I		2		4	20	30	50
AEC(2)	AEC-101-ENG	English	2		2		20	30	50
VEC(2)	VEC-101-ENV	EVS-I	2		2		20	30	50
Total			14	08	14	16			550

** The subjects offered to other faculty students under OE vertical are OE-101-CS -P/ OE-102-CS -T/OE-103-CS-P / OE-104-CS-T. The students of B.Sc. (Computer Science) will opt the subjects offered by other faculty given in University Basket.*

Level:- 4.5 (First Year) **Sem:-II**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Subject 1	CS-151-T	Advanced C Programming	2		2		20	30	50
	CS-152-P	Lab Course Based on CS-151-T		2		4	20	30	50
Subject 2	MTC-151-T	Graph Theory	2		2		20	30	50
	MTC-152-P	Mathematics Practical II		2		4	20	30	50
Subject 3	ELC-151-T	Principles of Digital Electronics	2		2		20	30	50
	ELC-152-P	Electronics Practical Course II		2		4	20	30	50
GE/OE* (2)	OE-151-CS-T / OE-152-CS-T / OE-153-CS-T OE-154-CS-T	Office Automation II / Computer Fundamentals / Introduction to Google Apps II/ Fundamentals of Computers II	2		2		20	30	50
SEC(2)	SEC-151-CS-P	Statistical Methods for Computer Science II		2		4	20	30	50
AEC(2)	AEC-151-ENG	English	2		2		20	30	50
VEC(2)	VEC-151-ENV	EVS-II	2		2		20	30	50
CC(2)	CC-151-T	From University Basket	2		2		20	30	50
Total			14	08	14	16			550

** The subjects offered to other faculty students under OE vertical are OE-151-CS -P/ OE-152-CS -T/OE-153-CS-P / OE-154-CS-T. The students of B.Sc. (Computer Science) will opt the subjects offered by other faculty given in University Basket.*

***Exit option:** Award of UG Certificate in Major with 44 credits and an additional 4 credits core as per university guidelines OR Continue with Major and Minor*

Continue option: Student will select one subject among the (subject 2 and subject 3) as minor and subject 1 will be major subject

In Second Year, the "Subject 1: Computer Science" will be Major Subject and the Minor subject will be chosen from "Subject 2 or Subject 3". Subject 2 and Subject 3 will not be available as Major Subjects in Second Year and Third Year

Level:- 5.0 (Second Year) **Sem:-III**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (4+2)	CS-201-MJ-T	Data Structure -I	2		2		20	30	50
	CS-202-MJ-T	Database Management System I	2		2		20	30	50
	CS-203-MJ-P	Lab Course based on CS-201-MJ-T & CS-202-MJ-T		2		4	20	30	50
VSC(2)	CS-221-VSC-T	Software Engineering	2		2		20	30	50
IKS	CS-201-IKS-T	Indian Knowledge System in Computing	2		2		20	30	50
FP/OJT/ CEP(2)	CS-231-FP	Mini Project		2		4	20	30	50
Minor (2+2)	CS-241-MN-T	Mathematics or Electronics	2		2		20	30	50
	CS-242-MN-P	Mathematics or Electronics		2		4	20	30	50
GE/OE(2)	OE-201-CS-T OE -202-CS-T OE-203-CS-T OE-204-CS-T	E commerce I / Web Design I / Digital Marketing I/ AI for everyone I	2		2		20	30	50
AEC(2)	AEC-201-T	From University Basket	2		2		20	30	50
CC(2)	CC-201-T	From University Basket	2		2		20	30	50
Total			16	06	16	12			550

Level:- 5.0 (Second Year) **Sem:-IV**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (4+2)	CS-251-MJ-T	Data Structure - II	2		2		20	30	50
	CS-252-MJ-T	Database Management System II	2		2		20	30	50
	CS-253-MJ-P	Lab Course based on CS-251-MJ-T & CS-252-MJ-T		2		4	20	30	50
VSC(2)	CS-271-VSC-P	Advanced Python Programming		2		4	20	30	50
FP/OJT/ CEP(2)	CS-281-FP	Mini Project		2		4	20	30	50
Minor (2+2)	CS-291-MN-T	Mathematics or Electronics	2		2		20	30	50
	CS-292-MN-P	Mathematics or Electronics		2		4	20	30	50
GE/OE(2)	OE-251-CS-T / OE-252-CS-T / OE-253-CS-T / OE-254-CS-T	E commerce II/ Web Design II/ Digital Marketing II / AI for everyone - II	2		2		20	30	50
SEC(2)	SEC-251-CS-P / SEC-252-CS-P	Computer Networks / Statistical Analysis using R Software		2		4	20	30	50
AEC(2)	AEC251	From University Basket	2		2		20	30	50
CC(2)	CC-251-T	From University Basket	2		2		20	30	50
Total			12	10	12	20			550

Exit option: Award of UG Diploma in Major and Minor with 88 credits and an additional 4 credits core as per university guidelines OR Continue with Major and Minor

Level:- 5.5 (Third Year) **Sem:-V**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (8+4)	CS-301-MJ-T	Core Java	2		2		15	35	50
	CS-302-MJ-T	Operating Systems	2		2		15	35	50
	CS-303-MJ-T	Web Technology-I	2		2		15	35	50
	CS-304-MJ-T	Theory of Computer Science	2		2		15	35	50
	CS-305-MJ-P	Lab Course based on CS-302-MJ-T		2		4	15	35	50
	CS-306-MJ-P	Lab Course based on CS-301-MJ-T & CS-303-MJ-T		2		4	15	35	50
Major Elective (2+2)	CS-307-MJ-T	Data Science And Analytics	2		2		15	35	50
	CS-308-MJ-P	Lab Course based on CS-307-MJ-T		2		4	15	35	50
	OR								
	CS-309-MJ-T	Database Technologies	2		2		15	35	50
	CS-3010-MJ-P	Lab Course on CS-309-MJ-T		2		4	15	35	50
	OR								
	CS-3011-MJ-T	Embedded Systems	2		2		15	35	50
	CS-3012-MJ-P	Lab Course on CS-3011-MJ-T		2		4	15	35	50
VSC(2)	CS-321-VSC-P	Foundation of Artificial Intelligence and Machine Learning		2		4	15	35	50
FP/OJT/ CEP(2)	CS-331-FP	Project		2		4	15	35	50
Minor (2)	CS-341-MN-T	Mathematics or Electronics	2		2		15	35	50
Total			12	10	12	20			550

Level:- 5.5 (Third Year) **Sem:-VI**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (8+4)	CS-351-MJ-T	Advanced Java	2		2		15	35	50
	CS-352-MJ-T	Design Framework	2		2		15	35	50
	CS-353-MJ-T	Web Technology-II	2		2		15	35	50
	CS-354-MJ-T	Compiler Construction	2		2		15	35	50
	CS-355-MJ-P	Lab Course based on CS-352-MJ-T		2		4	15	35	50
	CS-356-MJ-P	Lab Course based on CS-351-MJ-T & CS-353-MJ-T		2		4	15	35	50
Major Elective (2+2)	CS-357-MJ-T	Android Programming	2		2		15	35	50
	CS-358-MJ-P	Lab Course based on CS-357-MJ-T		2		4	15	35	50
	OR								
	CS-359-MJ-T	Software Testing Tools	2		2		15	35	50
	CS-3510-MJ-P	Lab Course based on CS-359-MJ-T		2		4	15	35	50
	OR								
	CS-3511-MJ-T	Internet of Things	2						
	CS-3512-MJ-P	Lab Course based on CS-3511-MJ-T		2					
VSC(2)	CS-371-VSC-P	Agile Processes		2		4	15	35	50
FP/OJT/ CEP(4)	CS-381-OJT	OJT		4		8	30	70	100
Total			10	12	10	24			550

Level:- 6.0 (Fourth Year) **Sem:-VII (Research)**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (6+4)	CS-401-MJ-T	Advanced Operating System	2		2		15	35	50
	CS-402-MJ-T	Artificial Intelligence	2		2		15	35	50
	CS-403-MJ-T	Principles of Programming Language	2		2		15	35	50
	CS-404-MJ-P	Lab Course based on CS-401-MJ-T		2		4	15	35	50
	CS-405-MJ-P	Lab Course based on CS-402-MJ-T		2		4	15	35	50
Major Elective (2+2)	CS-406-MJ-T	Advance Databases and Web Technologies	2		2		15	35	50
	CS-407-MJ-P	Lab Course on CS-406-MJ-T		2		4	15	35	50
	OR								
	CS-408-MJ-T	Cloud Computing	2		2		15	35	50
	CS-409-MJ-P	Lab Course on CS-408-MJ-T		2		4	15	35	50
	OR								
	CS-410-MJ-T	C# .NET Programming	2		2		15	35	50
	CS-411-MJ-P	Lab Course on CS-410-MJ-T		2		4	15	35	50
FP/OJT/ CEP/RP (4)	CS-431-RP	Research Project		4		8	30	70	100
	CS-451-MN	Research Methodology	4		4		30	70	100
Total			12	10	12	20			550

Level:- 6.0 (Fourth Year) **Sem:-VIII (Research)**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (6+4)	CS-451-MJ-T	Design and Analysis of Algorithms	2		2		15	35	50
	CS-452-MJ-T	Mobile App Development Technologies	2		2		15	35	50
	CS-453-MJ-T	Software Project Management	2		2		15	35	50
	CS-454-MJ-P	Lab Course based on CS-451-MJ-T		2		4	15	35	50
	CS-455-MJ-P	Lab Course based on CS-452-MJ-T		2		4	15	35	50
Major Elective (2+2)	CS-456-MJ-T	Full Stack Development I	2		2		15	35	50
	CS-457-MJ-P	Lab Course based on CS-456-MJ-T		2		4	15	35	50
	OR								
	CS-458-MJ-T	Web Services	2		2		15	35	50
	CS-459MJ-P	Lab Course based on CS-458-MJ-T		2		4	15	35	50
	OR								
	CS-460-MJ-T	ASP DOT Net Programming	2		2		15	35	50
	CS-461-MJ-P	Lab Course based on CS-460-MJ-T		2		4	15	35	50
FP/OJT/ CEP(8)	CS-481-FP	Research Project		8		16	60	140	200
Total			08	14	08	28			550

Level:- 6.0 (Fourth Year) **Sem:-VII (Honors)**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (10+4)	CS-401-MJ-T	Advanced Operating System	2		2		15	35	50
	CS-402-MJ-T	Artificial Intelligence	2		2		15	35	50
	CS403MJ-T	Principles of Programming Language	2		2		15	35	50
	CS-404-MJ-P	Lab Course based on CS401MJ		2		4	15	35	50
	CS-405-MJ-P	Lab Course based on CS402MJ		2		4	15	35	50
	CS-406-MJ-T	Advanced Networking	2		2		15	35	50
	CS-407-MJ-T	Digital Marketing	2		2		15	35	50
Major Elective (2+2)	CS-408-MJ-T	Advance Databases and Web Technologies	2		2		15	35	50
	CS-409-MJ-P	Lab Course on CS-408-MJ-T		2		4	15	35	50
	OR								
	CS-410-MJ-T	Cloud Computing	2		2		15	35	50
	CS-411-MJP-T	Lab Course on CS-410-MJ-T		2		4	15	35	50
	OR								
	CS-412-MJ-T	C# .NET Programming	2		2		15	35	50
	CS-413-MJ-P	Lab Course on CS-412-MJ-T		2		4	15	35	50
	CS-441-MN-T	Research Methodology	4		4		30	70	100
Total			16	06	16	12			

Level:- 6.0 (Fourth Year) **Sem:-VIII (Honors)**

Course Type	Course Code	Course Title	Credits		Teaching Scheme Hr/Week		Evaluation Scheme and Max Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core (10+4)	CS-451-MJ-T	Design and Analysis of Algorithms	2		2		15	35	50
	CS-452-MJ-T	Mobile App Development Technologies	2		2		15	35	50
	CS-453-MJ-T	Software Project Management	2		2		15	35	50
	CS-454-MJ-P	Lab Course based on CS-451-MJ-T		2		4	15	35	50
	CS-455-MJ-P	Lab Course based on CS-452-MJ-T		2		4	15	35	50
	CS-456-MJ-T	Crypto Currency Technologies	2		2		15	35	50
	CS-457-MJ-T	Cyber Security	2		2		15	35	50
Major Elective (2+2)	CS-458-MJ-T	Full Stack Development I	2		2		15	35	50
	CS-459-MJ-P	Lab Course based on CS-458-MJ-T		2		4	15	35	50
	OR								
	CS-460-MJ-T	Web Services	2		2		15	35	50
	CS-461-MJ-P	Lab Course based on CS-460-MJ-T		2		4	15	35	50
	OR								
	CS-462-MJ-T	ASP DOT Net Programming	2		2		15	35	50
	CS-463-MJ-P	Lab Course based on CS-462-MJ-T		2		4	15	35	50
FP/OJT/ CEP(4)	CS-481-OJT	OJT		4		8	30	70	100
Total			12	10	12	20			

SEM-V

Savitribai Phule Pune University T.Y.B.Sc.
(Computer Science) - SEM– V

Course Type: Major **Course Code: CS-301-MJ-T**

Course Title : Core Java

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Knowledge of C Programming language

Course Objectives

- To learn Object Oriented Programming language
- To study various java programming concepts like Interface, File and Exception Handling etc.
- To design User Interface using Swing

Course Outcomes

On completion of the course, student will be able to–

CO1: Learn the data types, variables, operators, control statements and loops.

CO2: Understand the concept of classes, objects, packages.

CO3: Learn and implement the concept of Inheritance and Interface.

CO4: Understand File and Exception Handling.

CO5: Implement application using the abstract classes and adapter classes.

CO6: Develop GUI based applications.

Course Contents

Chapter 1	**An Introduction to Java**	**5 Hours**

1.1 Object Oriented Programming Concepts

1.2 Introduction to Java

1.3 Features of Java

1.4 Java Environment:

1.4.1 Structure of Java Program

1.5 Java Development Tools

1.6 Basic concepts in Java-

1.6.1 Comments, Data Types

1.6.2 Tokens

1.6.3 Variable

1.7 Control Flow Statements

1.8 Arrays in Java

1.9 Accepting Input (Command Line Arguments, BufferedReader, DataInputStream, Scanner)

Chapter 2	**Objects and Classes**	**6 Hours**

2.1 Defining your own classes
2.2 Access Specifiers: public, protected, private, default
2.3 Array of Objects
2.4 Constructors, Overloading Constructors
2.5 Use of 'this' keyword static block, static fields and methods
2.6 Predefined Classes
  2.6.1 Object Class, Methods (equals(), toString(),hashcode(), getClass())
  2.6.2 String Class And StringBuffer Class,Formatting String data using format() method
2.7 Creating , Accessing And Using Packages
2.8 Wrapper Classes

Chapter 3	**Inheritance and Interface**	**6 Hours**

3.1 Inheritance Basics (extends Keyword) and Types of Inheritance
3.2 Superclass, Subclass and use of Super Keyword
3.3 Method Overriding and runtime polymorphism
3.4 Use of final keyword related to method and class
3.5 Use of abstract class and abstract methods
3.6 Defining and Implementing Interfaces
3.7 Runtime polymorphism using interface
3.8 Functional Interface and Marker Interface
3.8 Adapters and Anonymous Inner Class

Chapter 4	**Exception and File Handling**	**5 Hours**

4.1 Dealing with errors
4.2 Exception class
  4.2.1 Checked and Unchecked Exception
  4.2.2 Logger Class
  4.2.3 Catching Exceptions,
  4.2.4 Multiple Catch Block
  4.2.5 Nested try block
  4.2.6 Creating User Defined Exception
4.3 Introduction to Files and Streams
 4.3.1 Input-OutputStream : FileInput/OutputStream, BufferedInput/OutputStream,
 4.3.2 DataInput/OutputStream
   4.3.2.1 Reader-Writer : FileReader/Writer, BufferedReader/Writer,

4.3.2.2 InputStreamReader, OutputStreamWriter

Chapter 5	**User Interface with JavaFX**	**8 Hours**

5.1 What is JavaFX?

5.2 Architecture of JavaFX

5.3 LifeCycle of a JavaFX Application

5.4 Common JavaFX Layouts

5.4.1 HBox,VBox, BorderPane, GridPane, FlowPane, StackPane

5.5 Containers and Components

5.5.1 Button, Label, Text, TextArea, CheckBox And RadioButton, List, ComboBox, Menu And related Classes

5.6 JavaFX Charts

5.6.1 Pie Chart, Line Chart, Area Chart, Bar Chart

5.7 Event Handling: Adding and Removing Event Handlers

Reference Books:

R1: Complete reference Java by Herbert Schildt(5th edition)

R2: Java 2 programming black books, Steven Horlzner

R3: Programming with Java , A primer ,Forth edition , By E. Balagurusamy

R4: Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Pres

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - SEM– V

Course Type: Major **Course Code: CS-302-MJ-T**

Course Title : Operating Systems

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

Data structures like stack, queue, linked list, tree, graph, hashing, file structures, any structured programming language

Course Objectives

- The course provides concepts that underlie the operating systems
- Understanding Modern Operating System and their principles.
- Introduction to the process scheduling, CPU scheduling algorithms, memory management, Deadlock handling , file system and Disk Management as well as disk scheduling algorithms

Course Outcomes

Upon the successful completion of the course students will be able to

CO1: Understand concepts of Operating System, its structure and services provided by OS.
CO2: Understand the mechanisms of context switching, Scheduling etc.
CO2: Analyze the efficiency and performance of algorithms related to CPU scheduling.
CO3: Analyze the efficiency of paging, segmentation, virtual memory for efficient memory Utilization.
CO4: Understand the techniques of deadlock handling.
CO5: Implement Scheduling algorithms, memory management techniques and disk scheduling techniques.
CO6: Understand and evaluate disk organization and management

Course Contents

Chapter 1	**Introduction to Operating System**	**3Hours**

1.1 Basic concepts-Definition, operating system functions and services, need or purpose of operating system.
1.2 Operating system Structure.
1.3. Types of Operating System.
1.4 Operating system operation
1.5 Booting process of Operating system
1.6 Advantage and disadvantages of operating System.

Chapter 2	**Process and Process Scheduling**	**7 Hours**

2.1 Basic Concepts: Process, Process states, Process Control Block, Operations on Process
2.2 Process Scheduling- Scheduling, Scheduling queues, CPU Schedulers and its type, context switching, CPU and I/O Burst Cycle, Scheduling Criteria, types of scheduling.
2.3 Scheduling Algorithms: First Come First Serve(FCFS),Shortest Job First(SJFs), Priority Scheduling , Round Robin (RR)

Chapter 3	**Memory Management**	**8 Hours**

3.1 Introduction to memory management, address binding, Logical versus physical address space, Role and functions of Memory Management Unit (MMU)
3.2 Paging concepts-paging model, page table structure and working of paging.
3.3 Segmentation concepts–Segmentation, Hardware, Paging VS Segmentation

3.4 Virtual Memory Management-Concept, Demand paging, Performance of demand paging

3.5 Page replacement algorithms – FIFO, Optimal, LRU, MFU.

Chapter 4	**Deadlock**	**7 Hours**

4.1 Basic Concepts-Deadlock definition, process synchronization, resource allocation graph and Necessary condition.

4.2 Deadlock handling Methods- Deadlock Prevention, Deadlock Avoidance (Safe state, Banker's Algorithm), Deadlock Detection

4.3 Recovery from Deadlock – Process termination, Resource preemption.

Chapter 5	**File System Management and Disk Scheduling**	**5 Hours**

5.1 Basic Concepts: File, File attributes, File operations.

5.2 Allocation Methods – Contiguous allocation, Linked allocation, Indexed allocation

5.3 Free Space Management and Disk Structure–Free space management techniques: Bit vector, linked list, Grouping, Counting.

5.4 Disk structure

5.5 Disk Scheduling algorithms- FCFS Scheduling, SSTF Scheduling, Scan scheduling, and look scheduling.

5.6

Reference Books:

R1: Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, Student Edition, Wiley Asia Students Edition

R2: Operating Systems: Internals and Design Principles, Prentice Hall of India. William Stallings,

R3: Advanced Concepts in Operating Systems, Tata McGraw Hill Inc, 2001 (Text Book) M Singhal and NG Shivaratri

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: Major **Course Code: CS-303-MJ-T**

Course Title: Web Technology-I

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Knowledge of Programming Fundamentals
- Basic knowledge of Computer Fundamentals
- Familiarity with basic web browser and web Applications

Course Objectives

- Introduction to scripting languages.
- Development of interactive and dynamic client-side applications.
- To analyze asynchronous programming concepts and execution model for backend development.
- To design and implement basic server-side applications and file handling systems.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand the structure and functionality of HTML5 elements, CSS3 styling techniques and core JavaScript concepts used in frontend development.

CO2: Develop responsive and interactive web pages using HTML5, CSS3and modern JavaScript (ES6+) features.

CO3: Analyze asynchronous programming concepts and Node.js execution model for handling server-side operations efficiently.

CO4: Design and implement basic backend applications including HTTP server creation and file handling using Node.js modules.

Course Contents

Chapter 1	**Frontend Fundamentals (HTML, CSS, Core JavaScript)**	**9 Hours**

1.1 HTML5 Fundamentals

1.1.1 Structure of HTML5 document (<!DOCTYPE>, <html>, <head>, <body>)

1.1.2 Semantic elements (header, nav, section, article, footer, aside)- Forms and input controls-Input types, labels, validation attributes, Form submission methods (GET/POST)

1.1.3 Multimedia elements – Audio, video, iframe embedding, Accessibility considerations (alt text, captions)

1.2 CSS3 Styling & Layout- Selectors, specificity, pseudo-classes, pseudo-elements, Box model, margin, padding, border, overflow, Layout techniques -Flexbox alignment and distribution, CSS Grid rows, columns, areas, Responsive design, Media queries, mobile-first approach, Fluid images and typography

1.3 Core JavaScript-Variables, data types, operators, Conditional statements and loops, Functions and scope, Arrays and objects manipulation, DOM selection, traversal, and modification, Event handling (click, submit, keyboard, load)

Chapter 2	**Advanced JavaScript (ES6+)**	**5 Hours**

2.1 ES6 Syntax Features-Arrow functions and lexical this, Template literals and string interpolation, Object and array destructuring, Default parameters

2.2 Advanced Operators and Modules-Spread and rest operators, ES6 modules (import, export, module structure)

2.3 Asynchronous JavaScript-Callback concept and limitations, Promises lifecycle (pending, fulfilled, rejected), Chaining promises and error propagation, async / await syntax and try-catch handling

2.4 Logic-Building Programs-Mathematical logic (prime, factorial, Fibonacci), Pattern generation using loops, Searching and sorting fundamentals

Chapter 3	**Node.js Fundamentals**	**10 Hours**

3.1 Node.js Environment-Node runtime architecture, npm installation, package management, Understanding package.json

3.2 Execution Model-Event loop phases, Non-blocking I/O and callbacks

3.3 Core Modules-File system (fs), Path utilities (path), HTTP server creation (http)

3.4 Basic Server Development-Creating and running HTTP server, Handling requests and responses, Executing JavaScript outside browser

Chapter 4	**File Handling with Node.js**	**6 Hours**

4.1 File Operations- Reading, writing, appending, deleting files, Synchronous vs asynchronous methods

4.2 JSON Handling-Parsing JSON data, Writing structured JSON files

4.3 Directory Management-Creating, listing, renaming, deleting folders

4.4 File Upload Basics-Introduction to upload middleware, Handling uploaded file storage

Reference Books:

R1: Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley Publications.

R2: Jon Duckett, *JavaScript and JQuery: Interactive Front-End Development*, Wiley Publications.

R3: Marijn Haverbeke, *Eloquent JavaScript (3rd Edition)*, No Starch Press.

R4: Kyle Simpson, *You Don't Know JS: ES6 & Beyond*, O'Reilly Media.

R5: Eric A. Meyer and Estelle Weyl, *CSS: The Definitive Guide*, O'Reilly Media.

R6: Ethan Brown, *Web Development with Node and Express*, O'Reilly Media.

R7: Azat Mardan, *Practical Node.js*, Apress.

R8: Andrew Mead, *Learning Node.js Development*, Packt Publishing.

Savitribai Phule Pune University T.Y.B.Sc.
(Computer Science)
Sem – V

Course Type: Major **Course Code: CS-304-MJ-T**

Course Title : Theory of Computer Science

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Knowledge of Sets, relations, functions, and logic in discrete mathematics
- Knowledge of Fundamental Algorithms and Data Structures.
- Basic knowledge of Programming Concepts (any language)

Course Objectives

- To introduce the mathematical foundations of computation.
- To examine the computational complexity of algorithms and problems.
- To learn about grammars and formal languages and how they are used in computing.
- To gain knowledge of formal computation models like Turing machines and automata.
- To develop strong problem-solving and proof-building abilities.

Course Outcomes

On completion of the course, student will be able to–

CO1: Utilize grammars, automata, and formal language principles to simulate computational issues.
CO2: Sort or classify issues according to their decidability and computability.
CO3: Understand theoretical ideas to real-world fields like cryptography, algorithms, AI and compilers.
CO4: Build formal proofs for decidability, complexity, and language recognition.
CO5: Understand Turing machines design and finite automata.
CO6: Use formal languages and automata to model real world problem

Course Contents

Chapter 1	**Finite Automaton**	**8 Hours**

1.1 Introduction: Symbol, Alphabet, String, Prefix and Suffix of Strings,
1.2 Deterministic finite Automaton – Definition, and Examples.
1.3 Nondeterministic finite automaton – Definition and Examples.
1.4 NFA To DFA (NFA without Epsilon)
1.5 Finite automaton with output – Moore machine and Mealy machine, Definition and Examples.
1.6 Difference between Moore machine and Mealy machine
1.7 Minimization of DFA, Algorithm and Problem using Table Method.

Chapter 2	Regular Expressions and Languages	6 Hours

2.1 Regular Expressions (RE): Definition and Example
2.2 Regular language -Definition and Examples.
2.3 Conversion of RE to FA-Examples.
2.4 Pumping lemma for regular languages and applications.

Chapter 3	Context-Free Grammars and Languages	5 Hours

3.1 Definition of CFG
3.1.1 Components: Non-terminals, Terminals, Production Rules, Start Symbol
3.2 Simplification of CFG:
3.2.1 Removing Useless Symbols
3.2.2 Unit Production
3.2.3 ϵ-production
3.2.4 Nullable Symbol.
3.3 Normal Forms: Chomsky Normal Form (CNF), Examples
3.4 Greibach Normal Form (GNF), Examples

Chapter 4	Push Down Automata	5 Hours

4.1 Definition of PDA and examples
4.2 Construction of PDA using empty stack and final State method: Examples using stack method.

Chapter 5	Turing Machine	6 Hours

5.1 The Turing Machine Model, Definition and Design of TM
5.2 Problems on language recognizers.
5.3 Language accepted by TM.

Reference Books:

R1: Introduction to the Theory of Computation - Michael Sipser

R2: Introduction to Automata Theory, Languages and Computation - John E. Hopcraft, Rajeev Motwani, Jeffrey D. Ullman, Third Edition, Pearson Education Publication

R3: Theory of Computer Science Automata, Languages and Computation - K.L.P. Mishra, N. Chandrasekaran, Publication- Prentice Hall of India

R4: Introduction to Languages and The Theory of Computation - John C. Martin The McGrawHill, Fourth Edition

R5: An Introduction to the Theory of Computer Science: Languages and Machines - Thomas A. Sudkamp

Savitribai Phule Pune University Pune

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: Major **Course Code: CS-305-MJ-P**

Course Title: Practical Course based on CS – 302-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Data structures like stack, queue, and linked list.
- File, file attributes and file operations.

Course Objectives

- To understand the concept of process scheduling with the help of simulation.
- To study the concept of demand paging in operating system.
- To implement Banker's algorithm for Deadlocks in Process management.
- To simulate File system management
- To study and implement various algorithms of disk scheduling

Course Outcomes

After completion of this course students will be able to,

CO1: Understand and execute the Operations on processes.

CO2: Understand and implement the concept of process scheduling.

CO3: Implement and analyse FIFO, LRU, Optimal, and MFU page replacement algorithms using C Programming.

CO4: Execute deadlock detection algorithms.

CO5: Implement the File allocation methods

CO6: Execute and analyse working of disk scheduling algorithms.

Operating Environment/Software's –

1. Operating system platform – Linux
2. Programming language - C

Suggested List of Assignments

Assignment No.	Title	No of Slots (12)
Assignment 1	Operations on processes :(Create a child process using fork() and commands like exec(), execv() and execvp())	**01 slot**
Assignment 2	Simulation of CPU Scheduling Algorithms – FCFS, SJF, Priority and Round Robin	**03 slots**
Assignment 3	Simulation of demand paging using memory page replacement algorithms – FIFO, LRU, OPT, MFU	**02 slots**
Assignment 4	Simulation of Banker's algorithm of deadlock avoidance in processes of operating system	**02 slots**
Assignment 5	Simulation of File Allocation methods Contiguous allocation, Linked allocation, Indexed allocation	**03 slots**

Assignment 6	Simulation of Disk Scheduling algorithms – FCFS, SSTF, Scan, Look	**01 slot**

Reference Books:

R1: Operating System Concepts ,Avi Silberschatz,Peter Galvin,Greg Gagne, Student Edition,Wiley Asia

R2: Operating Systems – A Practical Approach, R. S. Salaria

R3: Operating Systems: Design and Implementation, Andrew S. Tanenbaum, Albert S. Woodhull

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: DSEC - III Course Code: 306-MJ-P

Course Title : Practical Course based on CS-301-MJ-T & CS-303-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
4 Hours / week	2	IE : 15 marks UE: 35 marks

Prerequisites

- Basic knowledge of HTML tags and CSS properties.
- Fundamental understanding of programming logic (loops, conditions).
- Familiarity with web browser developer tools.

Course Objectives:

- Enable students to write, compile, and execute Java programs using basic syntax and structure.
- Practice using decision-making statements and loops to solve real-world problems.
- To design and style structured, responsive web interfaces using semantic HTML and CSS layouts
- To implement client-side logic and manipulate the Document Object Model (DOM) dynamically
- To utilize modern JavaScript features (ES6+) for cleaner and more efficient coding practices.
- To handle asynchronous operations and simulate real-world data fetching scenarios.
- To set up and run a basic web server and manage local data storage using Node.js.

Course Outcomes:

CO1 : Understand the concept of classes and objects

CO2 : Access and manipulate object data using arrays.

CO3 : Develop modular programs using packages and implement interfaces for abstraction.

CO4: Create responsive, multi-column web layouts using Flexbox and Media Queries.

CO5: Perform dynamic DOM manipulation and client-side form validation.

CO6: Apply ES6+ features.

CO7: Implement asynchronous logics and build basic server-side applications and REST APIs.

Operating Environment/Software's –

- Operating system platform – Linux , JDK 17 or 21
- Editor –Eclipse IDE , Visual Studio Code, Sublime Text, or Atom.
- Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge.
- Runtime: Node.js (v14.x or higher) and npm.

Suggested List of Assignments

Assignment No.	Title	No of Slots (12)
Assignment 1	**Java Tools and IDE, Simple java programs** (Introduction to the java environment Use of java tools like java, javac, jdb and javadoc Defining simple classes and creating objects.)	**01 slot**
Assignment 2	**Objects and Class** (Creating a class and array of objects. Creating a package.)	**01 slot**
Assignment 3	**Inheritance and Interfaces** (To implement inheritance in java. To define abstract classes. To define and use interfaces and Functional Interface.)	**01 slot**
Assignment 4	**Exception And File Handling** (User defined Exception classes. Creation of files and demonstration of I-O operations)	**01 slot**
Assignment 5	**GUI Designing, Event Handling** (To demonstrate GUI creation using JavaFX, Event classes, JavaFX Charts)	**02 slots**
Assignment 6	**Frontend Foundations (HTML5 & CSS3)**	**01 slot**
Assignment 7	**Interactive Web Logic (Core JavaScript & DOM)**	**01 slot**
Assignment 8	**Modern Scripting with ES6+**	**01 slot**
Assignment 9	**Asynchronous Programming in JavaScript**	**01 slot**
Assignment 10	**Server-Side Basics with Node.js**	**01 slot**
Assignment 11	**File System & API Fundamentals**	**01 slot**

Reference Books:

R1: Head First Java – by Kathy Sierra & Bert Bates

R2 : Java: The Complete Reference – by Herbert Schildt

R3 : Core Java Volume I – Fundamentals – by Cay S. Horstmann

R4: Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke.

R5: Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics by Jennifer Robbins.

R6: Node.js Web Development by David Herron.

R7: You Don't Know JS (Book Series) by Kyle Simpson. R5: JavaScript: The Definitive Guide by David Flanagan.

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: Major Elective **Course Code: CS-307-MJ-T**

Course Title: Data Science And Analytics

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic of mathematics and statistics
- Advance Python programming knowledge
- Knowledge of databases

Course Objectives

- To equip students with knowledge and skills for data-driven problem solving using real-world datasets.
- To develop a strong understanding of Data Science concepts, lifecycle, and applications across domains.
- To enable students to perform data cleaning, pre-processing, transformation, and dimensionality reduction.
- To train students in effective data visualization techniques and tools for meaningful interpretation of data.
- To introduce foundational machine learning techniques for prediction, classification, and clustering.
- To encourage analytical thinking and ethical practices in data handling and decision-making.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand fundamental concepts, components, and real-world applications of Data Science.

CO2: Classify and interpret different types of data, identify outliers, and assess data quality using standard data quality measures.

CO3: Apply data pre-processing techniques to prepare datasets for analysis.

CO4: Analyse datasets using Exploratory Data Analysis (EDA) and evaluate appropriate data visualization techniques and tools to effectively communicate insights.

CO5: Apply data analytics techniques and evaluate analytical models.

CO6: Build and analyse basic machine learning models, select suitable learning approaches, and validate models for real-world data science problems.

Course Contents

Chapter 1	**Introduction to Data Science**	**4 Hours**

1.1 Introduction to data science

1.2 Concept of big data –

1.2.1 Definition

1.2.2 Data types with example (Structured, Semi-Structured, Unstructured Data)

1.2.3 Challenges with unstructured data

1.2.4 Data Sources (Open Data, Social Media Data, Multimodal Data, standard datasets)

1.2.5 Data Formats (Integers, Floats, Text Data, Text Files, Dense Numerical Arrays,

Compressed or Archived Data, CSV Files, JSON Files, XML Files, HTML Files, Tar Files, GZip Files, Zip Files, Image Files

1.2.6 3 V's of bigdata (Volume, Velocity, Variety)give

1.3 Application of Data Science

1.4 Data Science Lifecycle

1.5 Role of Data Scientist

1.6 Data Scientist's Toolbox(Programming Languages, Libraries, Database, Visualization tools)

Chapter 2	**Data Understanding and Preprocessing**	**3 Hours**

2.1 Concepts of data –

- 2.1.1 Definition
- 2.1.2 Types of Data (Qualitative (categorical) – Nominal, Ordinal and Quantitative (numerical) – Discrete, Continuous)
- 2.1.3 Outlier – Types of outlier with example

2.2 Data Pre-processing –

- 2.2.1 Definition
- 2.2.2 Need to pre-process the data
- 2.2.3 Data quality
- 2.2.4 Measures of Data Quality **(**Accuracy, Completeness, Consistency, Timeliness, Validity, and Uniqueness)

Chapter 3	**Data Pre-processing Techniques**	**7 Hours**

3.1 Data Pre-processing techniques-

- 3.1.1 Data Cleaning – Missing Value, Noisy Data, Inconsistent Data
- 3.1.2 Data Integration - Problems in data integration
- 3.1.3 Data Transformation – Strategies of data transformation (Rescaling, Normalizing, Binarizing, Standardizing, Label and One Hot Encoding)

3.2 Data Reduction -

- 3.2.1 Dimensionality Reduction
- 3.2.2 Feature Selection
- 3.2.3 Feature Extraction
- 3.2.4 Data Cube Aggregation
- 3.2.5 Numerosity Reduction

3.3 Data Discretization –

- 3.3.1 Approaches of data Discretization (Top-down, Bottom-up)
- 3.3.2 Types of Data Discretization techniques (Equal Width Binning, Equal Frequency Binning)

3.5 Introduction to Exploratory Data Analysis(EDA)

Chapter 4	**Data Visualization**	**3 Hours**

4.1 Introduction

4.2 Advantages of data Visualization

4.3 Advantages and visual encoding

4.4 Data visualization libraries

4.5 Data visualization tools - Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts, Donut charts, Boxplots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots

4.5 Advanced data visualization tools- Wordclouds, Visualization of geospatial data

Chapter 5	**Introduction to Data Analytics**	**4 Hours**

5.1 Introduction

5.2 Life Cycle of Data Analytics
5.3 Types of data analytics - Descriptive Analytics, Diagnostic Analytics, Predictive Analytics, Prescriptive Analytics
5.4 Mathematical models - Introduction, Need of mathematical models in Data Analytics
5.5 Model evaluation –
Metrics for evaluating classifiers – Accuracy, Confusion matrix, Precision, Recall and F1 Score, ROC(Receiver-Operator Characteristic), AUC (Area Under ROC Curve), Mean Squared Error (MSE), Cross Validation, Log-loss

Chapter 6	**Introduction to Machine Learning**	**9 Hours**

6.1 Introduction to Machine Learning, Deep Learning, Artificial intelligence
6.2 Applications for machine learning in data science
6.3 Modelling process, Engineering features and selecting a model
6.4 Training the model
6.5 Validating the model
6.6 Predicting new observations
6.7 Types of machine learning- Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement Learning
6.8 Machine learning models –
6.8.1 Supervised Learning Model – Regression (Linear Regression, Polynomial Regression, Logistic Regression), Classification (KNN, Random Forest)
6.8.2 Unsupervised Learning Models - Clustering (K-means clustering), Association (Apriori Algorithm)

Reference Books:

R1: Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, Third Edition, 2012.
R2: Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020
R**3:** A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press
R4: The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017
R5: Introduction to Machine Learning - Alpaydin Ethem, fourth edition.
R6: M1-Machine-Learning-Tom-Mitchell, McGraw-Hill

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: Major Elective **Course Code: CS-308-MJ-P**

Course Title: Lab Course based on CS-307-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic of mathematics and statistics
- Basic programming knowledge of python
- Knowledge of databases

Course Objectives

- To understand fundamental concepts of data processing, data visualization, and machine learning.
- To apply data cleaning, integration, transformation, reduction, and discretization techniques on real datasets.
- To use basic and advanced data visualization tools to analyze and present data insights effectively.
- To build supervised and unsupervised machine learning models
- To evaluate machine learning model performance using standard metrics and validation techniques.
- To encourage analytical thinking and ethical practices in data handling and decision-making.

Course Outcomes

On completion of the course, student will be able to–

CO1: Apply data cleaning, integration, transformation, and reduction techniques to prepare datasets for analysis.

CO2: Create and interpret basic and advanced data visualizations to identify patterns, trends, and insights.

CO3: Implement supervised machine learning models for regression and classification to solve real-world prediction problems.

CO4: Analyse and evaluate the performance of machine learning models using appropriate metrics and validation techniques.

CO5: Apply unsupervised learning techniques such as clustering and association rule mining to discover hidden patterns in data.

Operating Environment –

Operating System : Linux

Python 3.8+

Code Editor(VS Code/Jupyter)

Suggested List Of Assignments

Assignment No.	Title	Number of slots 12
Assignment 1	Techniques for Data Processing (Cleaning , Integration, Transformation , Reduction and Discretization)	**1 slot**
Assignment 2	Fundamentals Tools Data Visualization (Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts, Donut charts, Bubble plots)	**2 slots**
Assignment 3	Advanced Tools for Data Visualization (Boxplots, Heat map, Dendrogram, Venn diagram, Treemap, 3D scatter plots ,Word clouds, Geospatial data)	**2 slots**
Assignment 4	Supervised Machine Learning Models for Regression ((Linear Regression, Polynomial Regression, Logistic Regression)	**3 slots**
Assignment 5	Supervised Machine Learning Models for Classification (KNN, Random Forest)	**2 slots**
Assignment 6	Unsupervised Machine Learning Models for Clustering (K-means clustering) and Association Rule Mining(Apriori Algorithm)	**2 slots**

Reference Books:

R1: Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann,
Third Edition, 2012.

R2: Fundamentals of Data Visualization, Wilke, C. O., O'Reilly Media, 2019.

R3: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Géron, A., 2nd Edition,
O'Reilly Media, 2019.

R4: Python Data Science Handbook, VanderPlas, J., O'Reilly Media, 2016.

R5: Introduction to Machine Learning - Alpaydin Ethem, fourth edition

R6: M1-Machine-Learning-Tom-Mitchell, McGraw-Hill

Savitribai Phule Pune University
T.Y.B.Sc.(Computer Science) - Sem – V

Course Type: Major Elective **Course Code: CS-309-MJ-T**

Course Title: Database Technologies

Teaching Scheme	**No. of Credits**	**Examination Scheme**
02 Hours / Week	2	IE: 15 Marks UE: 35 Marks

Prerequisites-

- Basic Database Knowledge: Understanding of RDBMS concepts, SQL queries, tables, and relationships.
- Programming language (C, C++, Java, or Python) for database interaction.
- Concepts of network architecture and client-server communication

Course Objectives-

After completing this course, students will be able to:

- Understand the evolution of databases and modern data management technologies.
- Learn NoSQL databases and their types (Key–Value, Document, Column-Family, Graph).
- Gain practical skills in MongoDB for data modelling, CRUD operations, and aggregation.
- Understand distributed, cloud, and Big Data database concepts, including scalability and fault tolerance.
- Apply database concepts to real-world applications and explore emerging trends and career opportunities.

Course Outcomes-

After successfully completing this course, students will be able to:

CO1: Understand the differences between RDBMS, NoSQL, and Big Data platforms.

CO2: Identify and use appropriate NoSQL database types for different applications.

CO3: Design and implement MongoDB databases with CRUD, indexing, and aggregation.

CO4: Understand distributed/cloud database architectures and basic security and recovery strategies.

CO5: Apply database knowledge to industrial use cases and stay informed about emerging trends and career paths.

Course Contents

Chapter 1	**Evolution of Databases and Modern Data Management**	**5 Hours**

1.1 Evolution of data storage systems
1.2 Limitations of traditional RDBMS in Big Data and cloud environments
1.3 Structured vs semi-structured vs unstructured data
1.4 Introduction to Modern Database Technologies
1.4.1 SQL databases (relational, traditional)
1.4.2 NoSQL databases: Document, Key–Value, Column-Family, Graph
1.4.3 Distributed and cloud databases
1.4.4 Big Data platforms (HDFS, Hadoop ecosystem)
1.4.5 Comparison: Features, scalability, consistency, availability
1.5 Industrial relevance and emerging trends

Chapter 2	**NoSQL Databases – Concepts & Architecture**	**6 Hours**

2.1 Introduction to NoSQL databases
2.1.1. Definition, need, and evolution of NoSQL
2.1.2. Limitations of traditional RDBMS and why NoSQL is needed
2.2 CAP Theorem & ACID vs BASE
2.2.1. Consistency, Availability, Partition Tolerance (CAP)
2.2.2. ACID properties in RDBMS vs BASE properties in NoSQL
2.3 Distributed data storage concepts
2.3.1. Data partitioning (sharding), replication, clustering
2.3.2. Nodes, clusters, and data distribution in distributed systems
2.4 Scalability, replication, and fault tolerance
2.4.1 Vertical vs Horizontal scaling
2.4.2 Replication strategies
2.4.3 Handling failures and ensuring data availability

Chapter 3	**NoSQL Data Models**	**3 Hours**

3.1 Types of NoSQL Databases
3.1.1 Key–Value: Redis, DynamoDB - Overview and applications
3.1.2 Column-Family: Cassandra, HBase - Concept and applications
3.1.3 Graph: Neo4j - Nodes, relationships, Cypher query overview
3.1.4 Document-oriented model – MongoDB
3.1.4.1 MongoDB architecture & installation
3.1.4.2 Collections & documents
3.1.4.3 Data types in MongoDB
3.1.4.4 CRUD operations
3.1.5 Indexing and aggregation framework
3.1.6 Mini project design (Student Management / Blog System)

Chapter 4	**Big Data Technologies and Hadoop Ecosystem**	**4 Hours**

4.1 Introduction to Big Data – characteristics, sources, challenges

4.2 Hadoop ecosystem overview – HDFS, MapReduce basics

4.3 Comparison: RDBMS vs NoSQL vs Big Data platforms

4.4 Real-world case studies – IoT, E-commerce, Social Media

Chapter 5	**Distributed & Cloud NoSQL Databases, Security, and Emerging Trends**	**5 Hours**

5.1 Overview of distributed and cloud NoSQL databases – Cassandra, Redis, Neo4j (conceptual)

5.2 Security challenges in NoSQL and cloud databases

5.3 Authentication and authorization concepts

5.4 Backup and recovery in distributed databases

5.5 Industrial applications of NoSQL databases

5.5.1 Case studies (E-commerce, Social Media, IoT)

5.5.2 Emerging trends and career opportunities

Reference Books:

R1 : Database System Concepts – Abraham Silberschatz, Henry F. Korth, S. Sudarshan (Fundamentals of RDBMS, transactions, and SQL concepts)

R2 : NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence – Pramod J. Sadalage, Martin Fowler (Types of NoSQL databases, data models, and use cases)

R3 : MongoDB: The Definitive Guide – Kristina Chodorow (Practical guide for MongoDB CRUD, indexing, aggregation, replication, and sharding)

R4 : Hadoop: The Definitive Guide – Tom White (Big Data concepts, HDFS, MapReduce, and Hadoop ecosystem overview)

R5 : Designing Data-Intensive Applications – Martin Kleppmann (Distributed databases, replication, scalability, fault tolerance, and cloud database principles)

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem –V

Course Type: Major Elective **Course Code: CS -3010-MJ-P**

Course Title: Lab Course based on CS-309-MJ-T

Teaching Scheme	No. of Credits 2	Examination Scheme
04 Hours / Week		IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic understanding of Database Management Systems (DBMS)
- Knowledge of SQL and data organization concepts
- Familiarity with basic programming concepts and JSON format

Course Objectives

- To understand modern data models and data representation techniques
- To install, configure, and work with MongoDB database systems
- To perform data manipulation and querying using MongoDB
- To explore advanced data processing techniques like aggregation and indexing
- To understand graph databases and perform querying using Neo4j

Course Outcomes

On completion of the course, student will be able to–

CO1: Design and represent data using structured and semi-structured formats

CO2: Install and configure MongoDB and create databases and collections

CO3: Perform CRUD operations and manipulate data in MongoDB

CO4: Apply aggregation and indexing techniques for efficient data processing

CO5: Model and query graph data using Neo4j and Cypher language

Operating Environment/Softwares –

MongoDB Community Server
MongoDB Compass
Neo4j Desktop / Neo4j Browser
Windows / Linux Operating System
Command Line Interface / VS Code

Suggested List of Assignments

Assignment No.	Title	No of Slots (12)
Assignment 1	Design and Representation of Data using Structured, Semi-Structured, and JSON Formats	**2 Slots**

Assignment 2	Installation and Configuration of MongoDB with Database and Collection Creation	**2 Slots**
Assignment 3	Execution of CRUD Operations and Data Manipulation in MongoDB	**2 Slots**
Assignment 4	Application of Aggregation Framework and Indexing Techniques in MongoDB	**2 Slots**
Assignment 5	Modeling of Graph Data using Nodes and Relationships in Neo4j	**2 Slots**
Assignment 6	Implementation of Graph Queries and Relationship Analysis using Cypher in Neo4j	**2 Slots**

Reference Books:

R1 : Big Data Black Book by DT Editorial Services

R2 : Data Modeling with NoSQL Database by Pramod Sadalage

R3 : NoSQL Distilled by Pramod J. Sadalage, Martin Fowler

R4 : SQL & NoSQL Databases: Models, Languages, Security and Architectures by Andreas Meier, Michael Kaufmann

Savitribai Phule Pune University, Pune
T. Y. B. Sc. (Computer Science) - Sem – V
Course Type: Major Elective **Course Code: CS-3011-MJ-T**
Course Title: Embedded System

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

Students are expected to have prior knowledge of:

- Basic Digital Electronics (logic gates, flip-flops, counters)
- Fundamentals of Microprocessors and Microcontrollers
- Basic C programming concepts (data types, control structures, functions)
- Elementary understanding of electronics components (resistors, sensors, ICs)

Course Objectives

The objectives of this course are to:

- Introduce the fundamental concepts and architecture of embedded systems
- Enable understanding of embedded hardware building blocks and interfaces
- Familiarize students with Single Board Computers with emphasis on Arduino
- Develop skills in programming and interfacing peripherals using Arduino
- Expose students to real-world embedded system applications and case studies.

Course Outcomes

On successful completion of the course, the student will be able to:

CO1: Explain the concept, evolution, classification, and applications of embedded systems
CO2: Identify and describe the core components of a typical embedded system
CO3: Understand the architecture and ecosystem of Arduino-based systems
CO4: Develop basic programs using Arduino IDE and C/C++ based sketches
CO5: Interface sensors and actuators with Arduino for data acquisition and control
CO6: Design simple embedded solutions for real-life applications.

Course Contents

Chapter 1	**Introduction to Embedded system**	**8 Hours**

1.1 Definition and Characteristics of Embedded Systems
1.2 Evolution of Embedded Systems
1.3 Embedded Systems vs General Purpose Computing Systems
1.4 Classification of Embedded Systems:
  1.4.1 Based on generation
  1.4.2 Based on complexity and performance
  1.4.3 Based on deterministic behavior (hard/soft real-time)
  1.4.4 Based on triggering mechanisms
1.5 Purpose and Functional Aspects of Embedded Systems:

1.5.1 Data acquisition, storage, and representation
1.5.2 Data communication
1.5.3 Signal processing
1.5.4 Monitoring and control
1.5.5 Application-specific user interface
1.6 Major Application Areas: Consumer electronics, automotive, medical, industrial, communication, and IoT systems.

Chapter 2	**Embedded System Architecture**	**8 Hours**

2.1 Typical Embedded System Block Diagram and Functional Explanation
2.2 Processing Core of Embedded Systems:
2.2.1 General purpose processors
2.2.2 Domain-specific processors
2.2.3 ASICs, PLDs, and COTS components
2.3 Memory Organization:
2.3.1 RAM, ROM, PROM, Flash memory
2.3.2 Criteria for memory selection in embedded systems
2.4 Sensors and Actuators:
2.4.1 Introduction to sensors and classification
2.4.2 Actuators: types and applications
2.5 Communication Interfaces:
2.5.1 On-board interfaces: UART, I2C, SPI, Parallel Interface
2.5.2 External interfaces: RS-232, USB, IrDA

Chapter 3	**Arduino Fundamentals**	**7 Hours**

3.1 Introduction to Single Board Computers (SBCs)
3.2 Arduino Ecosystem and Open-Source Philosophy
3.3 Arduino UNO Architecture: ATmega328P Block Diagram
3.4 Power Supply, Clock System, and Reset Circuitry
3.5 Arduino Development Board Variants and Shields
3.6 Arduino IDE: Installation, Sketch Structure, and Compilation Process

Chapter 4	**Programming & Interfacing with Arduino**	**7 Hours**

4.1 Digital Input/Output Programming
4.2 Analog Input, ADC, and PWM Concepts
4.3 Timers, Interrupts, and Serial Communication (UART, I2C, SPI)
4.4 Interfacing Sensors: Temperature, Light, Motion Sensors
4.5 Interfacing Actuators: LED, Relay, DC Motor, Servo Motor
4.6 Case Studies: Home Automation, Security Systems, Environmental Monitoring, Industrial Automation

Reference Books:

R1 : Introduction to Embedded Systems by Shibu K, McGraw -Hill Publication
R2 : Embedded System By Rajkamal
R3 : Arduino made Simple with Interactive Projects By Ashwin Pajankar, BPB Publication.
R4 : Exploring Arduino: Tools and Techniques for Engineering Wizardry By Jeremy Blum, John Wiley & Sons, Inc.

Savitribai Phule Pune University, Pune

T. Y. B. Sc. (Computer Science) - Sem – V

Course Type: Major Elective Course Code: CS-3012-MJ-P

Course Title: Lab Course on CS-3011

Teaching Scheme	No. of	Examination Scheme
04 Hours / Week	Credits 2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic knowledge of electronics and digital logic
- Knowledge of Microcontroller basics (Arduino/8051) and Raspberry Pi

Course Objectives

The objectives of this laboratory course are to:

- Reinforce theoretical concepts of embedded systems through hands-on experiments
- Develop skills in Arduino programming and debugging
- Enable students to interface sensors and actuators with embedded platforms
- Familiarize students with serial communication and real-time control
- Encourage application-oriented thinking through mini case studies

Course Outcomes

On completion of the laboratory course, students will be able to:

CO1: Use Arduino IDE for writing, compiling, and uploading programs

CO2: Implement digital and analog I/O based embedded applications

CO3: Interface common sensors and actuators with Arduino

CO4: Use timers, interrupts, and communication protocols effectively

CO5: Design and demonstrate simple embedded system applications

Operating Environment/Softwares –

Laboratory Requirements

- Laptop/ Desktop System
- Arduino UNO boards with USB cable
- Breadboard, jumper wires
- Sensors: LM35/DHT11, LDR, PIR
- Actuators: LEDs, relay module, DC motor, servo motor
- Power supply and basic electronic components

Course Contents

SECTION - I (12 Experiments)

Group A:	**Basics of Embedded Systems & Arduino (Any 3)**	**3 Slots**

Experiment-1 is compulsory and Select any two form Experiment-2 to 4

1. Study of Arduino UNO board: pin configuration, power supply, onboard components, Installation and familiarization of Arduino IDE; writing and executing first sketch (on board led Blink program) (Compulsory)
2. Digital output interfacing: LED blinking using delay and loop constructs.
3. Interfacing of LEDs with Arduino for pattern generation.
4. Digital input interfacing: Multiple Switch/button interfacing with Multiple LEDs control

Group B:	**Analog I/O and Timing (Any 3)**	**3 Slots**

5. Analog input interfacing using potentiometer and displaying values on Serial Monitor
6. PWM-based LED brightness control using analogWrite()
7. Display of real time on serial monitor with Interfacing to an External Real-Time Clock (RTC)
8. Interrupt-based input handling using external interrupts.

Group C:	**Sensor Interfacing (Any 3)**	**3 Slots**

9. Interfacing temperature sensor (LM35 / DHT11) and displaying values on Display.
10. Interfacing light sensor (LDR) for automatic light control system
11. Interfacing ultrasonic / IR sensor for Displacement measurement.
12. Interfacing of speakers / buzzers to generate different tones.

Group D:	**Actuators and Communication (Any 3)**	**3 Slots**

13. Interfacing relay module for AC/DC load control (demonstration-level)
14. Interfacing DC motor using motor driver (L293D / L298) to control speed using Potentiometer.
15. Controlling an RGB LED Using the BlinkM Module (I2C communication)
16. Serial communication using USB: data transmission between Arduino and PC

SECTION - II : ACTIVITY (Any ONE Equivalent to 3 Experiments)

1. Electronics Hobby Project – Design and develop a small electronics-based project and submit a report.
2. Field Visit Report – Visit an electronics industry and submit a detailed report.

Reference Books

R1 : Shibu K., Introduction to Embedded Systems, McGraw-Hill
R2 : Rajkamal, Embedded Systems, Tata McGraw-Hill
R3 : Ashwin Pajankar, Arduino Made Simple with Interactive Projects, BPB Publications
R4 : Jeremy Blum, Exploring Arduino, Wiley

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: VSC **Course Code: CS-321-VSC-P**

Course Title: Foundation of Artificial Intelligence and Machine Learning

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours/Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic knowledge of Linear Algebra and Understanding of Probability and Statistics
- Basic programming knowledge in Python

Course Objectives

- To understand fundamental concepts of Machine Learning and AI
- To implement basic ML algorithms using Python
- To develop hands-on programming skills for implementing AI and ML algorithms to solve practical problems.
- To identify and analyze applications of AI and ML in various domains such as healthcare, finance, education, and robotics.

Course Outcomes

On completion of the course, student will be able to–

CO1: Analyze and solve real-world constraint satisfaction problems
CO2: Optimize search algorithms using techniques
CO3: Understand and execute the working of Artificial Neural Networks and CNN.
CO4: Understand, Evaluate and Implement the model based learning
CO5: Implement the fundamental concepts Naïve Bayes
CO6: Utilize AI and ML libraries such as NumPy, Pandas, Scikit-learn

Operating Environment / Tools (Suggested)

- Operating System: Linux
  Python (3.x)
  Jupyter Notebook / Google Colab
  Libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn

Course Contents

Assignment 1	**Problem Solving Using Search**	**1 Slot**

- Space State Representation
- Blind / Uninformed Searches: Breadth First Search (BFS) – Monkey Banana
- Water Jug Problem

- Depth Limited Search

Assignment 2	Informed Search Techniques	2 Slots

- Best First Search
- A* Algorithm
- AO* Algorithm
- Heuristic functions
- Hill Climbing
- Means-End Analysis

Assignment 3	Constraint Satisfaction Problems (CSP)	1 Slot

- Backtracking algorithm
- Domain Reduction Works
- GRAPH/Map Coloring Problem
- Sudoku

Assignment 4	Game Playing Algorithms for Decision Making	2 Slots

- Minimax Algorithm
- Alpha-Beta Pruning
- Game AI (Chess, Tic-Tac-Toe)
- Monte Carlo Tree Search (MCTS)

Assignment 5	Knowledge Representation and Logical Reasoning	1 Slot

- Propositional Logic and Predicate Logic
- Rule-based systems
- Forward Chaining
- Backward Chaining
- Ontological Engineering and Knowledge Graphs

Assignment 6	Basics of Machine Learning	1 Slot

- Machine Learning Types and Learning Algorithms
- Supervised
  - Bayes' Theorem, Gaussian and Bernoulli Naïve Bayes,
  - AdaBoost (Boosting)
  - Multiple Models (Voting)
  - Decision Trees
  - Random Forest (Bagging)
- Unsupervised
  - Distance Metrics
  - Objective Functions
  - K-Means

Assignment 7	Support Vector Machine (SVM)	1 Slot

- Hyperplane
- Classification

- Linear SVM
- Non-linear SVM

Assignment 8	Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN)	3 Slots

- Basic Neural Network
- ANN model for classification
- ANN for Regression
- CNN for Image recognition
- CNN for Object detection
- CNN for Computer vision tasks

Assignment 9	Case Studies	1 Slot

- Robot Navigation
- Timetable Scheduling
- Fraud Detection Systems
- Autonomous Systems Such as Self-Driving Cars
- Handwritten Digit Recognition and Speech Processing.

Reference Books :

R1: Artificial Intelligence: A Modern Approach by Stuart Russell & Peter Norvig

R2: Artificial Intelligence by Elaine Rich, Kevin Knight, Shivashankar Nair

R3: Artificial Intelligence: Structures and Strategies for Complex Problem Solving by George F. Luger

R4: Artificial Intelligence with Python by Prateek Joshi

R5: Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow by Aurélien Géron

R6: Introduction to Machine Learning with Python by Andreas Müller & Sarah Guido

R7: An Introduction to Statistical Learning (ISLR) by James, Witten, Hastie, Tibshirani

R8 : Support Vector Machines and Perceptrons by M.N. Murty, Rashmi Raghava

Savitribai Phule Pune University, Pune

T. Y. B. Sc. (Computer Science) - Sem – V

Course Type: FP/OJT/CEP Course Code: CS-331-FP

Course Title: Project

Teaching Scheme	No. of Credits 2	Examination Scheme
4 Hours/Week		CE:15 marks EE:35 marks

Prerequisites

- Knowledge of Software Development Life Cycle (SDLC)
- Understanding of system analysis and design concepts
- Familiarity with database concepts and data organization
- Basic knowledge of computer applications and tools
- Understanding of problem-solving and logical thinking techniques
- Ability to work in a team and communicate effectively

Course Objectives: -

- To enable students to apply theoretical knowledge to real-world problem solving.
- To develop skills in requirement analysis, system design, and implementation.
- To provide hands-on experience in software development using appropriate tools and technologies.
- To enhance teamwork, collaboration, and project management skills.
- To train students in preparing professional project documentation and presentations.

Course Outcomes: -

After completing the course, students will be able to:

CO1: Identify real-world problems and define appropriate project objectives and scope.

CO2: Analyze system requirements and design solutions using suitable methodologies and tools.

CO3: Develop and implement a software application using appropriate programming languages and technologies.

CO4: Design and manage databases effectively for the project system.

CO5: Demonstrate teamwork, communication, and project presentation skills.

CO6: Prepare structured project documentation following academic standards.

Project Guidelines

- Students can choose a project topic and implement the same using any language/technology covered in the curriculum so far. The operating environment must be windows.
- Students should work in a team of maximum 3 students.
- Each student within the group must work actively and contribute to the project work.
- The student group will work independently throughout the project work including: Preliminary investigation, Requirement specification, database design and system design.
- Project guide must conduct project presentations (minimum 2) to monitor the progress of the project groups.
- At the end of the project, the group should prepare a report which should conform to academic standards.
- The final project presentation with demonstration (UE) will be evaluated by the project guide (appointed by the college) and one external examiner (appointed by the University).

- All points in the project index, is a compulsory part for the project evaluation and should be carried out by each project group.
- The report should follow index as –

INDEX

Sr. No.	Title	Page No.	Signature of the Instructor
1	**Preliminary Investigation** 1.1 Problem Identification 1.2 Problem Statement / Definition 1.3 Purpose/objective and goals 1.4 Feasibility study 1.5 Project Scope and Limitations		
2	**Requirement Specification** 2.1 System Requirement - Software/hardware specifications - Platform, RAM, Memory storage (if required - specification of server / device) 2.2 Technical Requirement - Programming language, Tools if any 2.3 Functional Requirements 2.4 Data Requirements (As per the project requirement, student can include - Performance requirement & security requirements etc.)		
3	**Database Design** 3.1 Identify end users of the system 3.2 Identify the entities and attributes through ER diagrams 3.3 Identify all tables, fields, relationships between table		
4	**System Design** 4.1 Class diagram 4.2 Object diagram 4.3 Component diagram 4.4 Deployment diagram 4.5 Use case diagram 4.6 Activity diagram 4.7 Sequence diagram		
5	**Input and Output screen Layout**		
6	**Bibliography / References**		

Evaluation Guidelines:

CE (15 Marks)		EE (35 Marks)		
First Presentation	**Second Presentation**	**Project Logic / Presentation**	**Project Documentation**	**Viva**
7	**8**	**15**	**10**	**10**

SEM-VI

Savitribai Phule Pune University
T.Y.B.Sc. (Computer Science) - Sem – VI
Course Type: Major Core Course Code: CS-351-MJ-T
Course Title: Advanced Java

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Knowledge of Core Java
- Knowledge of Database.

Course Objectives

- To strengthen object oriented programming concepts.
- To develop database applications, web applications and enterprise application
- To develop real-world applications.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand the structure of the Collection Framework

CO2: Implement multithreading concepts

CO3: Develop database applications using JDBC

CO4: Build complete dynamic web systems using Servlet and JSP.

CO5: Understand the architecture of Spring Boot applications

CO6: Develop RESTful web services using Spring Boot.

Course Contents

Chapter 1	**Collections**	**4 Hours**

1.1 Introduction to the Collection framework
1.1.1 List - ArrayList, LinkedList
1.1.2 Set - HashSet, TreeSet,
1.1.3 Map – HashMap ,Hashtable and TreeMap
1.2 Interfaces
1.2.1Comparator, Iterator, ListIterator,
1.2.2 Enumeration

Chapter 2	**Multithreading**	**4 Hours**

2.1 Introduction of Thread
2.2 Life cycle of thread
2.3 Creating threads
2.3.1 Thread class
2.3.2 Runnable interface
2.4 Thread priorities
2.5 Multiple Thread
2.5.1 Synchronization
2.5.2 Interthread communication

Chapter 3	**Database Programming**	**6 Hours**

3.1 Introduction to Java Database Connectivity
3.2 The design of JDBC
3.3 Types of drivers
3.4 Executing SQL statements, query execution
3.5 Scrollable and updatable ResultSet

Chapter 4	**Servlets and JSP**	**6 Hours**

4.1 Introduction to Servlet and Hierarchy of Servlet
4.2 Life cycle of servlet
4.3 Handing GET and POST request (HTTP)
4.4 Handling data from HTML to servlet
4.5 Retrieving data from database to servlet
4.6 Session tracking Techniques
4.6.1 User Authorization,
4.6.2 URL rewriting,
4.6.3 Hidden form fields,
4.6.4 Cookies
4.6.5 HttpSession
4.7 Introduction to JSP, Life cycle of JSP
4.8 Implicit Objects
4.9 Scripting elements
4.9.1 Declarations
4.9.2 Expressions,
4.9.3 Scriplets
4.9.4 Comments
4.10 JMixing Scriplets and HTML

Chapter	Spring Boot and REST API	10 Hours

5.1 Introduction of Spring framework
5.2 Spring Framework Modules
5.3 Inversion of Control, Types Ioc Containers
5.4 Spring Boot Basics
5.5 Spring Boot MVC Fundamentals
5.6 Spring Boot with Database and Data JPA
5.7 Spring Boot - REST API

Reference Books:

R1: Programming with Java , A primer ,Forth edition , By E. Balagurusamy
R2: Complete reference Java by Herbert Schildt(5th edition)
R3: Java 2 programming black books, Steven Horlzner
R4: Core Java Volume-II-Advanced Features, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press
R5: Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press

Savitribai Phule Pune University T.Y.B.Sc.
(Computer Science) - Sem – VI

Course Type: Major **Course Code: CS-352-MJ-T**

Course Title : Design Framework

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic knowledge of HTML5 and CSS3
- Familiarity with Object-Oriented Programming (OOP) principles such as classes, objects, and inheritance.
- Basic command-line operations and Experience with any code editor or IDE
- Basic knowledge of REST concepts and client–server communication (optional but beneficial).

Course Objectives

- To introduce students to Angular framework and Single Page Application (SPA) concepts.
- To develop programming skills using TypeScript for Angular application development.
- To enable students to design component-based web applications.
- To provide hands-on experience with routing, forms, and validation techniques.
- To familiarize students with services and REST API integration.
- To guide students in building and deploying a complete Angular SPA mini project.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand the fundamentals of modern Angular and explain core concepts required to build single-page applications.

CO2: Develop Angular applications from the ground up using standalone components and signals for efficient and scalable design.

CO3: Implement Angular routing for navigation between views.

CO4: Develop forms with validation and integrate REST APIs using services.

CO5: Build and deploy a complete Angular Single Page Application mini-project.

CO6: Develop modern, complex, responsive and scalable web applications with Angular

Course Contents

Chapter 1	**Introduction to Angular**	**4 Hours**

1.1 Introduction to Angular Framework

1.1.1 Angular Features and Architecture

1.1.2 SPA Concepts

1.1.3 Angular vs AngularJS

1.2 TypeScript Fundamentals

1.2.1 Data Types, Classes, Interfaces

1.2.2 Functions and Arrow function
1.2.2 Modules and Decorators

1.3 Angular Design Philosophy

1.3.1 Component reusability

1.3.2 Maintainable application structure

1.3.3 Separation of concerns

Chapter 2	**Understanding Components & Data Bindings**	**7 Hours**

2.1 Angular CLI(Angular Command line Interface)

2.1.1 ng new, ng generate, ng build, ng serve

2.1.2 Project Structure

2.2 Components and Templates

2.2.1 Creating Components and Components Lifecycle Hooks(ngOnlnit, ngOnChanges, ngOnDestroy

2.2.2 Template Syntax

2.3 Data Binding

2.3.1 Interpolation

2.3.2 Property Binding

2.3.3 Event Binding

2.3.4 Two-way Binding

2.4 Directives and Pipes

2.4.1 Structural Directives

2.4.2 Attribute Directives

2.4.3 Built-in Pipes

Chapter 3	**Angular Routing**	**7 Hours**

3.1 Introduction to Angular Router

3.1.1 Concept of Client-side Routing

3.1.2 Router Module and router-outlet

3.1.3 Navigation using router Link

3.2 Route Configuration

3.2.1 Defining Routes and Path Mapping

3.2.2 Default and Wildcard Routes

3.2.3 Passing Parameters in Routes

3.3 Advanced Routing Concepts

3.3.1 Child Routes

3.3.2 Route Guards (CanActivate, CanDeactivate)

3.3.3 Lazy Loading of Modules

3.4 Preloading Strategies

3.4.1 Preload all modules

3.4.2 Custom preloading strategy

3.4.3 Error handling during navigation, Handling 404 and Error Routes

Chapter 4	**Services and Dependency Injection**	**6 Hours**

4.1 Introduction to Services

4.1.1. Creating a Service

4.1.2 Built in Services

(HTTP Services, Routing Services, From Services)

4.2 Dependency Injection

4.2.1 Core concepts in Dependency Injection

4.2.2 Need for DI in modern applications

4.2.3 Benefits of DI

4.2.4 Angular DI Framework

4.2.5 Providing Dependencies

4.2.6 Injection Tokens

4.2.7 DI Decorators

4.3 Providers

4.3.1 Types of Providers

4.3.2 Using RESTful services with DI

Chapter 5	**Angular Forms and Validation**	**6 Hours**

5.1 Introduction to Angular Forms

5.1.1 Data flow in forms

5.2 Template driven forms

5.2.1 Form Controls

5.2.2 Built-in Validators

5.2.3 Custom Validation

5.3 Reactive forms

5.3.1 FormGroup and FormControl

5.3.2 Dynamic Forms

5.4 Handling input fields with forms

5.5 Use of ngModel & ngGroup

5.6 Angular Validating

5.6.1 Validating forms with ngModel & ngGroup

5.6.2 Validating specific validation error

5.6.3 Output error messages

Reference Books:

R1: Angular Development with TypeScript by Yakov Fain & Anton Moiseev

R2: Getting Started with Angular by Victor Hugo Garcia.

R3: Angular: Up and Running by Shyam Seshadri

R4: Angular in Action by Jeremy Wilken

R5: Learning Angular (4th Edition) by Aristeidis Bampakos, Packt Publishing.

R6: Pro Angular by Adam Freeman

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem –VI

Course Type: Major **Course Code: CS-353-MJ-T**

Course Title : Web Technology-II

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic knowledge of HTML, CSS, JavaScript, and ES6+ features.
- Ability to create a basic server using Node.js.
- Understanding of relational databases and basic SQL (CRUD operations).
- Fundamental knowledge of HTTP, client–server architecture, and REST concepts.

Course Objectives

- To understand and implement database connectivity using PostgreSQL and Node.js with performance and security considerations.
- To design and develop RESTful APIs using Express framework following MVC architecture principles.
- To develop interactive frontend applications using React components, hooks, and API integration.
- To implement secure authentication and authorization mechanisms using sessions, cookies, and JWT.

Course Outcomes

On completion of the course, student will be able to–

CO1: Implement database connectivity using PostgreSQL with secure and optimized query execution techniques.

CO2: Design and develop RESTful APIs using Express framework following MVC architecture and proper documentation standards.

CO3: Develop interactive frontend applications using React components, hooks, and backend API integration.

CO4: Analyze and implement secure authentication and authorization mechanisms using sessions, cookies, and JWT.

Course Contents

Chapter 1	**Database Connectivity**	**4 Hours**

1.1 PostgreSQL Integration- Installing PostgreSQL and pg module, Establishing database connection

1.2 Performance and Security-Connection pooling, Parameterized queries, Environment variables configuration, SQL injection prevention techniques

Chapter 2	**CRUD Operations and REST API**	**6 Hours**

2.1 REST Architecture- REST principles and resource design, HTTP methods and status codes
2.2. Express Framework Structure- Routing and middleware, MVC architectural overview
2.3. CRUD Implementation-Create, Read, Update, Delete APIs, Pagination and filtering techniques
2.4. API Documentation-Postman collections, Swagger/ OpenAPI basics

Chapter 3	**Introduction to React**	**10 Hours**

3.1 React Fundamentals- Components and JSX syntax, Rendering flow and virtual DOM concept
3.2. State Management- Props vs state, Updating UI dynamically
3.3. React Hooks- useState for state handling, useEffect for lifecycle and API calls
3.4. Integration and Routing- Fetching backend APIs, Basic client-side routing

Chapter 4	**Forms, Sessions and Cookies**	**7 Hours**

4.1 Session Management-Express-session configuration, Session storage and lifecycle
4.2. Cookies Handling-Creating, reading, deleting cookies, Secure and HTTP-only flags
4.3. Form Processing-Server-side validation, Login and logout workflow
4.4. Authentication Models- Session-based vs token-based comparison

Chapter 5	**Authentication and Authorization**	**3 Hours**

5.1 Credential Security-Password hashing using bcrypt, Salting and verification
5.2. Token-Based Authentication-JWT structure and lifecycle, Token generation and validation
5.3. Access Control-Role-based authorization, Middleware for protected routes

Reference Books:

R1: Brown, E. (2019). Web Development with Node and Express (2nd ed.). O'Reilly Media.
R2: Cantelon, M., Harter, M., & Holowaychuk, T. J. (2017). Node.js in Action (2nd ed.). Manning Publications.
R3: Banks, A., & Porcello, E. (2020). Learning React (2nd ed.). O'Reilly Media.
R4: Simpson, K. (2015). You Don't Know JS: ES6 & Beyond. O'Reilly Media.
R5: Wilkens, A. (2015). Mastering PostgreSQL. Packt Publishing.

Savitribai Phule Pune University
T.Y.B.Sc. (Computer Science)
Sem – VI

Course Type: Major **Course Code: CS- 354-MJ-T**

Course Title : Compiler Construction

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic knowledge of familiarity with Finite Automata (FA), Regular Expressions (RE), and Context-Free Grammars (CFG).
- Knowledge of stacks, trees, graphs, symbol tables and Algorithms and basic complexity concepts
- Basic understanding of machine coding and instruction sets.

Course Objectives

- To understand lexical analysis, syntax analysis, and semantic analysis.
- To describe the basic ideas and stages involved in compiler design.
- To examine the runtime environment and intermediary code creation.
- To recognize how to manage symbol tables and handle errors.
- To learn how to build and implement compiler components in a practical way.

Course Outcomes

On completion of the course, student will be able to–

CO1: Describe compiler phases and structure of a compiler.
CO2: Build or create lexical analyzers by utilizing finite automata and regular expressions.
CO3: Build parsers with context-free grammars and parsing methods (LL, LR).
CO4: Evaluate or design straightforward target code generators and comprehend runtime environments
CO5: Utilize code optimization strategies
CO6: Use CFG's to parse programming language

Course Contents

Chapter 1	**Context-Free Grammars and Languages**	**6 Hours**

1.1 Context-Free Grammar (CFG)
1.1.1 Basics of CFG
1.1.2 Parse Tree
1.1.3 Left-most Derivation (LMD) – Definition and Example
1.1.4 Right-most Derivation (RMD) – Definition and Example
1.1.5 Examples of CFG for simple expressions or statements

1.2 Ambiguous Grammar
- 1.2.1 Definition / Concept of Ambiguity
- 1.2.2 Examples of Ambiguous Grammar
- 1.2.3 Effect of Ambiguity on Parsing
- 1.2.4 Resolving Ambiguity (Briefly: precedence & associativity)

Chapter 2	**Introduction to Compiler**	**4 Hours**

2.1 Definition of Compiler

2.2 Structure of Compiler
- 2.2.1 Front End
- 2.2.2 Back End

2.3 Phases of Compiler
- 2.3.1 Lexical Analysis
- 2.3.2 Syntax Analysis
- 2.3.3 Semantic Analysis (idea only)
- 2.3.4 Intermediate Code Generation (idea)
- 2.3.5 Code Optimization (idea)
- 2.3.6 Code Generation (idea)

2.4 Types of Compiler (One-pass, Multi-pass, Cross Compiler)
- 2.4.1 Definition
- 2.4.2 Advantages / Disadvantages / Features
- 2.4.3 Examples

Chapter 3	**Lexical Analysis(Scanner)**	**5 Hours**

3.1 Role of Lexical Analyzer

3.2 Tokens, Lexemes & Patterns
- 3.2.1 Definitions + Example

3.3 Finite Automata as Lexical Analyzer
- 3.3.1 Role of FA in token recognition
- 3.3.2 FA Structure and Working
- 3.3.3 Identifier and Number Recognition Using FA
- 3.3.4 Lexical Error Detection
- 3.3.5 Advantages of FA in Lexical Analysis

Chapter 4	**Syntax Analysis (Parser)**	**12 Hours**

4.1 Introduction to Parser
- 4.1.1 Definition of Parser
- 4.1.2 Types of Parsers
  - 4.1.2.1 Top-Down Parsers
  - 4.1.2.2 Bottom-Up Parsers

4.2 Top-Down Parsers
- 4.2.1 Backtracking: Method & Problems
- 4.2.2 Drawbacks of Top-Down Parsing with Backtracking
- 4.2.3 Left Recursion

- 4.2.3.1 Direct Left Recursion
- 4.2.3.2 Indirect Left Recursion
- 4.2.3.3 Examples
- 4.2.4 Left Factoring
- 4.2.4.1 Need for Left Factoring
- 4.2.4.2 Examples

4.3 Recursive Descent Parsing
- 4.3.1 Definition
- 4.3.2 Implementation of Recursive Descent Parser Using Recursive Procedures
- 4.3.3 Examples

4.4 FIRST & FOLLOW
- 4.4.1 Definition of FIRST Set
- 4.4.2 Definition of FOLLOW Set
- 4.4.3 Rules to Find FIRST & FOLLOW
- 4.4.4 Examples

4.5 Predictive Parser (LL(1) Parser)
- 4.5.1 Definition
- 4.5.2 Model
- 4.5.3 Implementation
- 4.5.4 Examples

4.6 Bottom-Up Parsers
- 4.6.1 Definition
- 4.6.2 Examples of Bottom-Up Parser Implementation
- 4.6.3 Types of Bottom-Up Parsers
- 4.6.3.1 Shift-Reduce Parsing
- 4.6.3.2 LR Parsers (SLR, CLR, LALR)
- 4.6.3.3 Operator Precedence Parsing

4.7 Shift-Reduce Parsing
- 4.7.1 Stack Implementation
- 4.7.2 Operations
- 4.7.2.1 Shift
- 4.7.2.2 Reduce
- 4.7.2.3 Accept
- 4.7.2.4 Error
- 4.7.3 Example(s)

4.8 LR Parsers
- 4.8.1 Components of LR Parser
  - Stack
  - Input Buffer
  - ACTION Table
  - GOTO Table
- 4.8.2 Types of LR Parsers
- 4.8.2.1 SLR(1)

4.8.2.2 Canonical LR (CLR)

4.8.2.3 LALR

4.8.3 Differentiation Between LR Parsers, Examples

4.9 Operator Precedence Parser

4.9.1 Definition

4.9.2 Operator precedence & associativity and Example

Chapter 5	**Code Generation and Optimization**	**3 Hours**

5.1 Compilation of expression

5.2 Concepts of operand descriptors and register descriptors with example.

5.3 Intermediate code for expressions

5.3.1 Postfix notations

5.3.2 Triples

5.3.3 Quadruples

5.3.4 Expression trees.

5.4 Code Optimization

5.4.1 Optimizing transformations

5.4.2 Compile time evaluation

5.4.3 Elimination of common sub expressions

5.4.4 Dead code elimination

5.4.5 Frequency reduction

5.4.6 Strength reduction

Reference Books :

R1: Compilers: Principles, Techniques, and Tools - Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman

R2: Modern Compiler Implementation - Andrew Appel and Jens Palsberg

R3: Advanced Compiler Design and Implementation — Steven S. Muchnick

R4: Principles of Compiler Design - Alfred V. Aho, Jeffrey D. Ullman, Narosa Publication House.

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – VI

Course Type: Major **Course Code: CS-355-MJ-P**

Course Title: Lab Course based on CS-352-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Basic knowledge of HTML, CSS, and JavaScript.
- Fundamentals of TypeScript.
- Understanding of object oriented programming concepts
- Familiarity with web browsers and basic command-line usage.

Course Objectives

- To Develop Angular applications using components, routing, and modules.
- To Design forms with validations and manage user input effectively.
- To Create and use services with dependency injection and API integration.
- To Build basic single-page applications with error handling and debugging.

Course Outcomes

On completion of the course, student will be able to–

CO1: Analyze and design a simple Angular application by integrating components, directives, data binding, and built-in pipes to solve a given problem.

CO2: Implement dynamic routing using route parameters and query parameters.

CO3: Develop mechanisms to handle invalid routes and navigation errors using 404 pages and error handling techniques.

CO4: Implement template-driven forms with built-in validations, validation messages, and form submission for basic data collection.

CO5: Create and use Angular services with dependency injection to share data and implement basic business logic across components.

Operating Environment/Softwares –

Operating System: Linux
Framework: Angular (latest stable version)
Runtime Environment: Node.js
Package Manager: npm
Code Editor / IDE: Visual Studio Code
Web Browser: Google Chrome / Mozilla Firefox
Optional Tools: Angular CLI, Postman (for API testing)

Suggested List of Assignments

Assignment No.	Title	No of Slots (12)
Assignment 1	• Angular Setup & TypeScript Fundamentals • TypeScript Interfaces & Angular Concepts	**2 Slots**
Assignment 2	• Angular Project Setup, Components & Lifecycle • Data Binding, Directives & Pipes	**2 Slots**
Assignment 3	• Angular Basic Routing & Navigation • Handling 404 Page, Error Handling During Navigation,	**2 Slots**

	Registration form using Reactive Forms + Signals.	
Assignment 4	• Angular Forms & Validation • Design a registration form using Reactive Forms + Signals.	**2 Slots**
Assignment 5	• Angular Services & Dependency Injection • Fetching Data Using Http Client	**2 Slots**
Assignment 6	• REST API Integration with Search Filter • Build a complete SPA	**2 Slots**

Reference Books:

R1: Angular: Up and Running – Shyam Seshadri
(Covers Angular fundamentals, routing, components, and services)

R2: Learning Angular – Aristeidis Bampakos and Pablo Deeleman
(Comprehensive guide including routing, guards, and lazy loading)

R3: Pro Angular – Adam Freeman
(Advanced concepts including routing strategies and application structure)

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – VI

Course Type: DSEC - VI **Course Code: CS-356-MJ-P**

Course Title : Practical Course based on CS-351-MJ-T & CS-353-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
4 Hours / week	2	IE : 15 marks UE: 35 marks

Prerequisites

- Understanding of object oriented programming concepts
- Basic knowledge of HTML, CSS, and JavaScript.

Course Objectives:

- To implement database connectivity and perform secure data operations using PostgreSQL.
- To understand and implement concepts such as JDBC, Servlets, and JSP for building dynamic applications.
- To understand the fundamentals of Spring Boot and its role in simplifying the Spring Framework.
- To understand the fundamentals of web development using Node.js, Express.js, and React.
- To develop RESTful APIs and implement CRUD operations using proper routing and middleware.
- To apply frontend concepts like React components, JSX, and hooks for building dynamic interfaces.

Course Outcomes:

On completion of the course, student will be able to–

CO1: Develop database applications using JDBC.

CO2: Learn fundamental concepts of web application processing using Servlets.

CO3: Understand modern Java frameworks for enterprise application development.

CO4: Develop real-world applications using Spring Boot and ORM technologies.

CO5: Understand basic concepts of web development and REST APIs.

CO6: Create and use APIs using Node.js and Express.js.

CO7: Build user interfaces using React components and hooks.

Guidelines:

Operating Environment :

- Operating system: Linux
- Editor: Any linux based editor like vi, gedit and Use of IDE – Eclipse / nano / VS Code (Linux version) etc.
- Compiler :javac

- Database :postgresql
- Runtime Environment: Node.js, API Testing Tool: Postman /Thunder
- Frontend Library: React.js
- Backend Framework: Express.js
- API Documentation Tool: Swagger / OpenAPI

Assignment No.	Title	No of Slots (12)
Assignment 1	**Collections** • To Implement various Interfaces and classes through algorithms. • To Demonstrate Cursor Objects (Enumeration, Iterator, List Iterator, Comparator)	**01 slot**
Assignment 2	**Multithreading** • To demonstrate multithreading using Thread Synchronization, • Inter-thread Communication, Thread Priorities.	**01 slot**
Assignment 3	**Database Programming** • To communicate with a database using java. • To execute queries on tables.	**01 slot**
Assignment 4	**Servlets** • To understand server-side programming. • Simple steps to create and execute servlets. • JSP Actions, and Handling data from HTML	**02 slot**
Assignment 5	**Spring Boot** • To create and understand the steps to develop Spring, Spring Boot	**01 slot**
Assignment 6	**Secure Database Connectivity using Node.js and PostgreSQL** • Connect Node.js with PostgreSQL securely using the pg module. • Perform CRUD operations using parameterized queries for SQL injection prevention. • Use environment variables and connection pooling for security and performance.	**01 slot**
Assignment 7	**CRUD API Development using Express.js with Postman and Swagger Documentation** • Develop RESTful APIs using Express.js and perform CRUD operations. • Test APIs using Postman and document them using Swagger.	**01 slot**
Assignment 8	**React components, JSX implementation and state management** • Understanding React Components and JSX Syntax. • Implement JSX , Managing Dynamic UI with React State	**01 slot**
Assignment 9	**Implementation of React Hooks and Client-Side Routing with API Integration**	**01 slot**

	• Implement React Hooks Using useState and useEffect • Integrate API with Client-Side routing • Building Multi-Page Applications Using React Router	
Assignment 10	**Forms, Cookies, Sessions and Authentication in Express.js** • Create and understand forms, cookies , session • Understand authentication process using express.js	**01 slot**
Assignment 11	**Authentication and Authorization using JWT and Bcrypt** • Understand authentication and Authorization using JWT • Understand authentication and Authorization using Bcrypt	**01 slot**

Reference Books:

R1 : Head First Servlets and JSP – by Bert Bates & Kathy Sierra

R2 : Murach's Java Servlets and JSP – by Joel Murach

R3 : Spring Boot in Action – by Craig Walls

R4 : Web Development with Node and Express – by Ethan Brown

R5 : REST API Design Rulebook by Mark Masse

Savitribai Phule Pune UniversityT.Y.B.Sc.
(Computer Science) - Sem – VI
Course Type: Major/Minor **Course Code: CS-357-MJ-T**
Course Title : Android Programming

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Concepts of object oriented programming
- Basic knowledge of the java programming.

Course Objectives

- Students should learn the Android Fundamentals and Android architecture framework.
- Students should understand GUI Design concepts and design Android GUI Layout.
- Students should be able to design visually appealing and intuitive user interfaces for Android apps, using appropriate layouts, widgets, and styles.
- Students should Develop and design event-driven programming with UI Controls.
- Students should understand how to manage data in Android applications, using SQLite databases, shared preferences, and data storage.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand the fundamentals of mobile application development, the Android operating system, its architecture, core components, and the Android Software Development Kit (SDK).

CO2: Understand and use activities, intents and layouts to build functional android applications.

CO3: Apply Android UI components, layouts, and design principles to create visually Appealing, responsive, and user-friendly mobile interfaces.

CO4: Implement appropriate data storage techniques in Android applications using SQLite databases, Shared Preferences.

CO5: Design, develop, test, and deploy complete Android applications independently using Android development tools.

Course Contents

Chapter 1	**Fundamentals of Android Programming**	**3 Hours**

1.1 Introduction to Android.
1.2 Overview and History of Android
1.3 Need and features of Android.
1.4 Android Architecture.

Chapter 2	Introduction to Android Environment	3 Hours

2.1 Android Environment
  2.1.1 Android IDE - Android Studio, Eclipse, Visual Studio with Xamarin
  2.1.2 Android SDK and JDK
  2.1.3 Android Tools – Android Development Tools(ADT), Android Virtual Devices(AVD), and Emulator
  2.1.4 Dalvik Virtual Machine Vs Java Virtual Machine
2.2 Android App/ Project folder structure
2.3 Resources and Manifest file
2.4 Android Application Hello World

Chapter 3	Activities, Intents and Layouts	7 Hours

3.1 Introduction to Activity
  3.1.1 Activity Life Cycle
3.2 Introduction to Intent
  3.2.1 Types of Intents: Implicit, Explicit
3.3 Introduction to Layout and Types
  3.3.1 View and View Group
  3.3.2 Linear Layout
  3.3.3 Relative Layout
  3.3.4 Constraint Layout
  3.3.5 Frame layout
  3.3.6 Table Layout
  3.3.7 Grid Layout
  3.3.8 Scroll Layout

Chapter 4	Android User Interface Design: GUI Components, menus and events	9 Hours

4.1 GUI Components
  4.1.1 Display Components – TextView, ImageView, AutoComplete TextView
  4.1.2 Input Components – EditText, CheckBox, RadioButton, RadioGroup, Spinner
  4.1.3 Action Components – Button, ImageButton, ToggleButton
  4.3.3 Notification Component - Toast
4.2 Data Display Controls
  4.2.1 List View
  4.2.2 Grid View
4.3 Dialog Box
  4.3.1 Alert Dialog
  4.3.2 Date Picker Dialog
  4.3.3 Time Picker Dialog
  4.3.4 Custom Dialog
4.4 Menus

4.4.1 Option Menu
4.4.2 Context Menu
4.4.3 Popup Menu
4.5 Event driven programming in Android
4.5.1 Event Source
4.5.2 Event Listener
4.5.3 Event Handler

Chapter 5	**Android Components and SQLite Database**	**8 Hours**

5.1 Components in Android: Activity, Services, Broadcast Receiver, Content Provider
5.2 Life Cycle of Android Components: Services, Broadcast Receiver
5.3 SQLite Database
5.3.1 Introduction to SQLite Database
5.3.2 SQLite, OpenHelper, and SQlite Database
5.3.3 Creating, Opening and Closing databases
5.3.4 Insert, Update, Delete operations in database
5.3.5 Fetching data from database.

Reference Books:

R1: Android Programming: The Big Nerd Ranch Guide -Bill Phillips, Chris Stewart

R2: Android Cookbook by Ian F. Darwin O'Reilly Media, Inc.

R3: Beginning Android by Mark L. Murphy, Wiley India Pvt Ltd publication.

R4: Android Programming for Beginners by Maclean David, Komatineni Satya, Allen Grant

R5: Building Android Apps by in easy Steps, McGraw-Hill Education publication.

Pro Android 5 by Hortan,John

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – V

Course Type: Major Elective **Course Code: CS-358-MJ-P**

Course Title: Lab Course based on CS-357-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Concepts of object oriented and XML layouts
- Knowledge of the java programming

Course Objectives

- Students should learn the Android Fundamentals and Android architecture framework.
- Students should understand GUI Design concepts and design Android GUI Layout.
- Students should be able to design visually appealing and intuitive user interfaces for Android apps, using appropriate layouts, widgets, and styles.
- Students should Develop and design event-driven programming with UI Controls.
- Students should understand how to manage data in Android applications, using SQLite databases, shared preferences, and data storage.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand the fundamentals of mobile application development, the Android operating system, its architecture, core components, and the Android Software Development Kit (SDK).

CO2: Develop Android applications from scratch by implementing user interface design, handling user interactions, and integrating essential application features.

CO3: Apply Android UI components, layouts, and design principles to create visually appealing, responsive, and user-friendly mobile interfaces.

CO4: Implement appropriate data storage techniques in Android applications using SQLite databases, Shared Preferences.

CO5: Design, develop, test, and deploy complete Android applications independently using Android development tools.

Operating Environment/Softwares –

Operating system: Linux

Programming Environment (Programming Language: Java, UI Design: XML Layouts, SDK: API Level 26 (Android 8.0))

Any compatible open source Android IDE (like - Android Studio, Eclipse, Visual Studio with SQLite database compatibility)

Computer System (Computer system with i3 and above processors which is available in the laboratory with minimum 8GB RAM)

Suggested List of Assignments

Assignment No.	Title	No of Slots (12)
Assignment 1	Introduction to Android and required installations	**2 Slots**
Assignment 2	Activity, Layout and Intent	**2 Slots**
Assignment 3	Android GUI Design Applications	**2 Slots**
Assignment 4	Android Event Handling	**2 Slots**
Assignment 5	Android Adapter and Menu	**2 Slots**
Assignment 6	Android Database Programming with SQLite	**2 Slots**

Reference Books:

R1: Android Programming: The Big Nerd Ranch Guide -Bill Phillips, Chris Stewart

R2: Android Cookbook by Ian F. Darwin O'Reilly Media, Inc.

R3: Beginning Android by Mark L. Murphy, Wiley India Pvt Ltd publication.

R4: Android Programming for Beginners by Maclean David, Komatineni Satya, Allen Grant

R5: Building Android Apps by in easy Steps, McGraw-Hill Education publication.

Pro Android 5 by Hortan, John

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – VI

Course Type: Major Elective **Course Code: CS-359-MJ-T**

Course Title : Software Testing Tools

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Knowledge of basic principles of software engineering
- Knowledge of software engineering models

Course Objectives

- To understand fundamental concepts of software testing and quality assurance.
- To learn various testing types and strategies.
- To design test cases, test plans, and defect reports.
- To gain practical knowledge of designing test cases.
- To apply software testing tools for real-world applications.

Course Outcomes

On completion of the course, student will be able to–

CO1: Learn software testing principles and identify errors and bugs.

CO2: Understand Software Testing Life Cycle and types of software testing techniques and tools.

CO3: Design test cases, Understand Entry and Exit Criteria for Test Case Design.

CO4: Understand how to write test plan and Prepare Test Report

CO5: Develop and evaluate a complete testing solution for a given software application using different Testing Tools

Course Contents

Chapter 1	**Introduction to Software Testing**	**4 Hours**

1.1 Introduction to Software Testing
1.2 Applications of Software Testing
1.3 Tester and Role of tester
1.4 Testing lifecycle
1.5 Verification and Validation
1.6 Software Testing – Myths
1.7 Understanding Errors, Bugs and Defects
1.7.1 Definitions: Error, Bug, Defect, Failure
1.7.2 Types of errors: syntax, logical, runtime
1.7.3 Examples of common application bugs
1.7.4 Impact of bugs on software quality
1.8 Techniques for Identification of Errors & Bugs
1.8.1 Requirement analysis and ambiguity detection
1.8.2 Code walkthrough and reviews
1.8.3 Static vs Dynamic testing for defect detection
1.8.4 Common bug types in web and desktop applications

Chapter 2	**Software Testing Life Cycle & Types of Software Testing**	**7 Hours**

2.1 Software Testing Life Cycle (STLC)
2.2 Software Testing Environment
2.3 Types of Software Testing
2.3.1 Manual Testing
2.3.1.1 White Box Testing
i. Path testing
ii. Conditional testing
Iii. Loop testing
iv. Unit testing
v. Testing the code from memory point of view
vi. Testing the code from performance point of view
2.3.1.2 Black Box Testing
a) Functional Testing
i. Unit Testing
ii. Integration Testing - Incremental & Non-incremental
iii. System Testing
iv. Acceptance Testing
b) Non-functional
i. Performance Testing - Load Testing, Volume Testing
ii. Usability Testing
2.3.1.3 Grey Box Testing
2.3.2 Automated Testing
2.4 Introduction to tools (Manual and Automated testing)

Chapter 3	**Introduction to Test case design**	**7 Hours**

3.1 Introduction to Defects
3.1.1 Defects, Defect Life Cycle, Defect Report
3.1.2 Defect Management Process - Stages, Advantages, Disadvantages
3.1.3 Introduction of Defect Tracking Tool -
- Open-source Tool - Bugzilla, MantisBT, Redmine
- Non-open source tool - Jira, HP ALM
3.2 Basics of Test Case Design
3.2.1 Definition of test case and test scenario
3.2.2 Importance of test case design in STLC
3.2.3 Difference between test case, test suite and test script
3.2.4 Characteristics of a good test case
3.3 Entry Criteria for Test Case Design
3.4 Exit Criteria for Test Case Execution
3.5 Designing Test Cases Using Excel
3.5.1 Structure of a test case template in Excel
3.5.2 Creating reusable templates
3.5.3 Version control and documentation standards
3.6 Features of Effective Testing Methods
3.6.1 Characteristics of good test cases
3.6.2 Risk-based testing approach
3.6.3 Importance of repeatability and traceability
3.6.4 Advantages of structured test case design
3.7 Designing and documenting test cases for
3.7.1 simple programs
3.7.2 programs using loops and control structures

Chapter 4	**Writing Test Cases and Test Plan**	**6 Hours**

4.1 Write Test Plan for given Application with Resources required

4.1.1 Test Plan Template
4.1.2 Create a Test Plan
4.2 Write Test Case for given Application
4.2.1 Test Case Template
4.2.2 Writing Test Case
4.3 Prepare Test Report for Test Cases Executed
4.3.1 Test Incident Report
4.3.2 Test Summary Report
4.4 Case Study

Chapter 5	Software Testing Tools	6 Hours

5.1 Software Testing Tools - Definition, Need, Advantages and limitations
5.2 Acceptance Testing Tools
5.2.1 FitNesse
5.2.2 Cucumber
5.3 Functional Testing Tools
5.3.1 TestComplete
5.3.2 Selenium
5.4 Performance Testing Tools
5.4.1 Apache JMeter
5.4.2 LoadRunner

Reference Books:

R1: Software testing Principle and Practices By Ramesh Desikan, Pearson Education, ISBN 81-7758-121-X 5.
R2: Software Testing Principles and Tools By M.G. Limaye TMG Hill Publication, ISBN 13:978-0-07-013990-9 3.
R3: Software Testing Principles and Practices By Naresh Chauhan, Oxford University Press, ISBN 0-19-806184-6 4.
R4: Software Testing Concepts and Tools By Nageshwar Rao , Dreamtech ,ISBN 81-7722 712-2
R6: The Art of Software Testing, 3rd Edition, by Glenford J. Myers, Tom Badgett, Corey Sandler.
R7: Software Testing, 2nd Edition, 2005 by Ron Patton
R8: A Practitioner's Guide to Software Test Design – Lee Copeland
R9: Ultimate Selenium WebDriver for Test Automation by Robin Gupta
R10: Apache JMeter Official Documents

Savitribai Phule Pune University
T.Y.B.Sc. (Computer Science) - Sem – VI
Course Type: Major Elective **Course Code: CS-3510-MJ-P**
Course Title: Lab Course based on CS-359-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Knowledge of software testing types.
- Knowledge of different test cases & test plans.

Course Objectives

- To understand how to design test cases.
- To know how to perform defect tracking
- To understand acceptance testing
- To understand functional testing.
- To be familiar with performance testing tool.

Course Outcomes

On completion of the course, student will be able to–

CO1: Design test cases using Excel

CO2: Design test case Design for web pages

CO3: Perform Defect tracking using Bugzilla

CO4: Perform acceptance test using FitNesse / Cucumber

CO5: Perform functional testing using Selenium

CO6: Perform performance testing using JMeter

Operating Environment/Softwares –

Operating System - Linux

Softwares -

- Excel
- Bugzilla
- Cucumber / FitNeesse
- Selenium
- JMeter

Assignment No.	Title	No of Slots (12)
Assignment 1	Test Case Design using Excel Tool for simple program	**2 Slots**
Assignment 2	Test Case Design for web pages	**2 Slots**
Assignment 3	Defect tracking using Bugzilla	**2 Slots**
Assignment 4	Acceptance Testing using FitNesse / Cucumber	**2 Slots**
Assignment 5	Functional Testing using Selenium	**2 Slots**
Assignment 6	Performance Testing using JMeter	**2 Slots**

Reference Books:

R1: Foundations of Software Testing – Dorothy Graham, Erik van Veenendaal, Isabel Evans & Rex Black

R2: Software Testing with Bugzilla

R3: Bugzilla Guide (Official Manual)

R4: FitNesse: For Better Collaboration on Acceptance Tests – Robert C. Martin (Uncle Bob) & Tim Ottinger

R5: The Cucumber Book: Behaviour-Driven Development for Testers and Developers – Matt Wynne & Aslak Hellesøy

R6: Selenium Testing Tools Cookbook – Unmesh Gundecha

R7: Selenium WebDriver Recipes in Python/Java – O'Reilly

R8: Performance Testing with JMeter 4 – Bayo Erinle

Savitribai Phule Pune University

T.Y. B.Sc. (Computer Science) - Sem – VI

Course Type: Major Elective **Course Code: CS-3511-MJ-T**

Course Title: Internet of Things

Teaching Scheme	No. of Credits	Examination Scheme
02 Hours / Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Students Should have basic knowledge of Networking, Internet, and Electronics
- Programming Skills: Proficiency in programming languages, particularly C and Python.
- Students are expected to have successfully completed courses in: Embedded Systems and Microcontrollers (Arduino)

Course Objectives

- Understand the concepts, architecture of Internet of Things.
- Explain IoT hardware platforms, sensors, actuators, and embedded controllers.
- Understand communication models and protocols used in IoT systems.
- Design IoT systems using single board computers.
- Understand IoT data processing, cloud platforms, and security issues.
- Apply IoT concepts to real-world applications and industry use cases.

Course Outcomes

On successful completion of the course, students will be able to:

CO1: Explain IoT concepts, architecture, and enabling technologies.

CO2: Identify and interface IoT hardware components using embedded platforms.

CO3: Select suitable communication protocols for IoT applications.

CO4: Develop basic IoT applications using Raspberry Pi.

CO5: Explain data handling, cloud integration, and security concerns in IoT.

CO6: Analyze and apply IoT solutions in smart and industrial domains

Course Contents

Chapter 1	**IoT Fundamentals and System Architecture:**	**8 Hours**

1.1 What is IoT? Definition, Characteristics, Application areas; Things in IoT and smart objects

1.2 Overview of Industrial Revolutions: Industry 1.0 to Industry 4.0

1.3 IoT Reference Model: 7-layer IoT reference model Published by the IoTWF

1.4 IoT enabling Technology (Concept only): WSN, Cloud Computing, Big Data, Embedded systems and boards, communication protocols, User Interfaces

1.5 WSN: Architecture, types of nodes

1.6 IoT Levels: Level-1 to level-5

1.7 IoT Distinction: IoT Versus Embedded Systems, IoT Versus WSN

Chapter 2	**Introduction to Sensor, actuation, Hardware Platforms & Cloud Integration:**	**8 Hours**

2.1 Sensors: Classification and examples

2.2 Actuators: Concept and Types of Actuators

2.3 Hardware Platform in IoT:

2.3.1 Introduction Raspberry Pi, Architecture Overview

2.3.2 Operating System for Raspberry Pi
2.3.3 Installation & configuration
2.3.4 GPIO architecture
2.3.5 Interfacing of LED, Switch and its Programming using Python
2.4 Cloud Computing for IoT:
2.4.1 Need and advantages of Cloud integration in IoT
2.4.2 Cloud Deployment Models: Public, Private, and Hybrid
2.4.3 Cloud Services: SaaS, PaaS, and Iaas
2.4.4 IoT cloud platforms (Overview)

Chapter 3	**IoT Communication Protocols and Networking technologies:**	**7 Hours**

3.1 IoT Protocols: MQTT, CoAP, XMPP, 6LoWPAN
3.2 IoT communication technologies: Zigbee, Z-Wave, NFC, Wi-Fi
3.3 Networking technologies:
3.3.1 Low power local area networking (LPLAN)
3.3.2 Low power wide area networking (LPWAN)
3.3.3 Comparative study of LoRa, Sigfox NB-IoT, LTE Cat –M

Chapter 4	**IoT Challenges, Security and Real-World Application:**	**7 Hours**

4.1 Challenges in IoT
4.2 Privacy and Data protection issues;
4.3 Case Study:
4.3.1 Smart Home
4.3.2 Smart Agricultural
4.3.3 Industrial IoT (IIoT)
4.3.4 Integration of IoT with AI

Reference Books:

R1: Vijay Madisetti and Arshdeep Bahga, — "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014.
R2: David Hanes, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, ISBN-13: 978-1-58714-456-1, ISBN-10: 1-58714-456-5, 2017
R3: RMD Sundaram Shriram K Vasudevan, Abhishek S Nagarajan, "Internet of Things", John Wiley and Sons.
R4: Enabling the Internet of Things: fundamentals, design, and applications / Muhammad Azhar Iqbal, Sajjad Hussain, Huanlai Xing, Muhammad Ali Imran., John Wiley & Sons Ltd
R5: Simon Monk, "Raspberry Pi Cookbook, Software and Hardware Problems and Solutions," O'Reilly Media, Inc
R6: Wireless Sensor Networks Technology: Protocols and Applications -Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley and Sons.

Savitribai Phule Pune University, Pune
T. Y. B. Sc. (Computer Science) - Sem – VI
Course Type: Major Elective Course Code: CS-3512-MJ-P
Course Title: Lab Course based on CS-3511-MJ-T

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours / Week	2	IE: 15 Marks

		UE: 35 Marks

Prerequisites

- Basic knowledge of Communication Protocols and Sensors
- Knowledge of Microcontroller basics (Arduino/8051)
- Knowledge of programming (C/Python) and wireless communication

Course Objectives

The objectives of this laboratory course are to:

- Understand the architecture and working of Raspberry Pi 4, including GPIO interfacing and OS setup.
- Develop practical skills in interfacing sensors and actuators such as LED, push button, temperature sensor, ultrasonic sensor, relay, and servo motor.
- Implement wireless communication protocols like Wi-Fi, Bluetooth, and MQTT for IoT applications.
- Integrate IoT devices with cloud platforms for real-time data monitoring, visualization, and storage.
- Design and develop real-world IoT applications such as smart home, agriculture, health monitoring, and industrial automation systems.
- Enhance problem-solving, system integration, and project development skills through hands-on experiments and activity-based learning.

Course Outcomes (Practical)

On completion of the laboratory course, students will be able to:

CO1: Understand the architecture of Raspberry Pi 4, configure Raspberry Pi OS.

CO2: Interface digital and analog sensors (DHT11/DHT22, LDR, Ultrasonic, PIR) and actuators (LED, Relay, Servo Motor) using GPIO pins and develop basic control programs.

CO3: Configure Wi-Fi and Bluetooth communication and implement HTTP and MQTT protocols using tools like Mosquitto for IoT data exchange.

CO4: Upload, store, and visualize sensor data on cloud platforms such as Thing Speak and develop dashboards using Node-RED.

CO5: Design and implement complete IoT-based application systems such as smart home automation, smart agriculture, and health monitoring solutions.

CO6: Prepare technical documentation and project reports based on practical implementation or industrial field visits.

Course Contents

SECTION - I (12 Experiments)

Group A	**Sensor & Actuator Interfacing using Raspberry Pi 4 (Any 3)**	**3 Slots**

1. Interfacing LED with Raspberry Pi (Digital Output).
2. Interfacing Push Button with Raspberry Pi (Digital Input).
3. Interfacing Temperature & Humidity Sensor (DHT11/DHT22).
4. Interfacing PIR Motion Sensor and buzzer alert system.

Group B	**IoT Communication (Wi-Fi / Bluetooth) (Any 3)**	**3 Slots**

5. Wi-Fi configuration on Raspberry Pi and IP address identification.
6. Implement MQTT Publish–Subscribe using Mosquitto.
7. Sensor data uploaded to the cloud using MQTT protocol.

8. Bluetooth communication between Raspberry Pi and Mobile.		
Group C	**Cloud Integration for IoT (Any 3)**	**3 Slots**
9. Upload temperature sensor data to ThingSpeak Cloud. 10. Real-time data visualization on cloud dashboard. 11. Store and retrieve IoT data using cloud services (IaaS / PaaS concept). 12. Node-RED installation and dashboard creation on Raspberry Pi.		
Group D	**IoT Application Based Experiments (Any 3)**	**3 Slots**
13. Smart Home Automation using Wi-Fi and Relay. 14. Smart Agriculture System (Soil moisture + Motor control). 15. Smart Health Monitoring (Temperature + Pulse sensor). 16. Industrial Parameter Monitoring using Cloud Dashboard.		

SECTION-II: ACTIVITY (Any ONE Equivalent to 3 Experiments)

1. Electronics Hobby Project – Design and develop a small electronics-based project and submit a report.
2. Field Visit Report – Visit an electronics industry and submit a detailed report.

Laboratory Requirements

- Monitor, Keyboard, Mouse, Active Internet Connection
- Raspberry Pi boards with HDMI cable
- Breadboard, jumper wires
- Sensors, Actuators
- Power supply and basic electronic components

Reference Books

1. Vijay Madisetti and Arshdeep Bahga, — "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014.
2. David Hanes, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, ISBN-13: 978-1-58714-456-1, ISBN-10: 1-58714-456-5, 2017
3. RMD Sundaram Shriram K Vasudevan, Abhishek S Nagarajan, "Internet of Things", John Wiley and Sons.
4. Enabling the Internet of Things: fundamentals, design, and applications / Muhammad Azhar Iqbal, Sajjad Hussain, Huanlai Xing, Muhammad Ali Imran., John Wiley & Sons Ltd
5. Simon Monk, "Raspberry Pi Cookbook, Software and Hardware Problems and Solutions," O'Reilly Media, Inc
6. Wireless Sensor Networks Technology: Protocols and Applications -Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley and Sons.

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – VI

Course Type: VSC **Course Code: CS-371-VSC-P**

Course Title: Agile Process

Teaching Scheme	No. of Credits	Examination Scheme
04 Hours/Week	2	IE : 15 Marks UE: 35 Marks

Prerequisites

- Fundamental principles of software engineering
- Programming basic principles, basic Git ideas and core database knowledge.

Course Objectives

- To learn the core values, principles, and popular frameworks followed in the software industry.
- To implement Agile requirements engineering through the use of user stories, acceptability criteria, and setting priorities.
- To incorporate industry-standard tools for Agile planning, collaboration, and version control.
- To carryout Agile estimation, sprint planning, review, and retrospective activities effectively.
- To develop a functional product incrementally while following to quality standards such as testing and code review.

Course Outcomes

On completion of the course, student will be able to–

CO1: Understand the Agile Manifesto, its fundamental principles, and the key differences between plan driven and Agile development methodologies.

CO2: Understand and create a product backlog utilizing user stories, acceptability criteria, and prioritization methods.

CO3: Utilize Scrum practices and roles to effectively plan and execute iterations for a small team project.

CO4: Utilize Kanban boards, work-in-progress limits, and performance metrics to enhance workflow efficiency and delivery.

CO5: Perform estimation, monitoring, and reporting utilizing burndown/burnup charts and velocity metrics.

CO6: Perform quality assurance practices including the development of test cases, foundational automated testing, continuous integration, and peer review.

Operating Environment / Tools (Suggested)

- Operating System: Windows/Linux
    - Version Control: Git and GitHub/GitLab
    - Agile Tool: Jira / Azure DevOps Boards (any one)
    - Documentation: Google Docs / Markdown
    - Communication: MS Teams / Slack (optional)

- CI (basic): GitHub Actions (optional)
- Testing: JUnit / PyTest / Selenium (as applicable)

Course Contents

Assignment 1	**Agile Fundamentals and Team Setup**	**2 Slots**

Topic Name: To learn the concepts of Agile and establish an effective team collaboration workflow.

Tools: GitHub and Markdown documentation.

Tasks:

a. Analyze and provide a summary of the Agile Manifesto and its 12 principles, including relevant real-world examples.
b. Assemble a team of three to five students and choose a mini-project domain (web, app, or API).
c. Setup a Git repository, define a branching strategy (main and feature branches), and prepare a README file.
d. Set up an Agile board featuring columns: Backlog, To Do, In Progress, Review, and Done.

Expected Outputs:

- Summary document on Agile methodology (1–2 pages).
- Link to the project repository accompanied by a README file.
- Screenshot or export of the board.

Assignment 2	**User Stories and Acceptance Criteria**	**1 Slot**

Topic Name: Create user stories and develop specific acceptance criteria for requirements.
Tools and Artefacts: Agile board tool, Documentation/Markdown

Tasks:

a. Identify stakeholders and create 6–10 user stories using the format: As a , I want , so that .
b. Create acceptance criteria utilizing the Given-When-Then framework for a minimum of six stories.
c. Design the Definition of Ready (DoR) and Definition of Done (DoD) for the team.

Expected Outputs:

• User story backlog of 6 to 10 stories.
• Document outlining acceptance requirements.
• Definition of Ready (DoR) and Definition of Done (DoD) checklist.

Assignment 3	**Backlog Prioritization and Release Planning**	**1 Slot**

Topic name: Priorities backlog and prepare a brief release utilizing Agile methodologies.
Tools/Artifacts: Agile board application, Spreadsheet/Documents

Tasks:

a. Priorities stories utilizing either MoSCoW or WSJF methodology.
b. Develop a clear release strategy for 2–3 sprints, each last 1–2 weeks.
c. Establish the scope of the MVP and develop a roadmap defining key milestones.

Expected Outputs:

• Prioritized backlog accompanied by justification.
• Document outlining the release plan and MVP scope.
• Overview of the roadmap

Assignment 4	**Agile Estimation and Sprint Planning**	**1 Slot**

Topic Name: Evaluate work and execute sprint planning for Sprint
Tools/Artifacts: Planning poker (cards/application), Agile board tool
Tasks:

a. Evaluate user stories utilizing story points (Planning Poker).
b. Split chosen stories into tasks and designate responsible parties.
c. Establish the Sprint Goal and Sprint Backlog for Sprint 1.
d. Establish sprint capacity and confirm commitments.

Expected Outputs:

• Table of story point estimates.
• Sprint 1 strategy (goal, backlog, assignments).
• Document of capacity and commitment.

Assignment 5	**Sprint Execution with Daily Scrum and Tracking**	**2 Slots**

Evaluate work and execute sprint planning for Sprint 1.
Tools/Artifacts: Planning poker (cards/application), Agile board tool
Tasks:

a. Evaluate user stories utilizing story points (Planning Poker).
b. Split chosen stories into tasks and designate responsible parties.
c. Establish the Sprint Goal and Sprint Backlog for Sprint 1.
d. Establish sprint capacity and confirm commitments.

Expected Outputs:

• Table of story point estimates.
• Sprint 1 strategy (goal, backlog, assignments).
• Document of capacity and commitment.

Assignment 6	**Testing and Quality Practices in Agile**	**2 Slots**

Topic name: Integrate testing into sprint workflow and ensure product quality.
Tools: Unit testing framework, Continuous Integration (optional)
Tasks:

a. Develop a minimum of six-unit tests or API tests in accordance with the acceptance criteria.
b. Develop test cases and align them with user stories.
c. Optional: Set up a fundamental CI pipeline to execute tests upon submit.
d. Conduct a basic code review checklist focused on maintainability.

Expected Outputs:

• Test case documentation and corresponding mappings.
• Verify code within the repository.
• Continuous Integration workflow file (if applicable).

Assignment 7	**Sprint Review and Demo**	**2 Slots**

Topic name: Demonstrate working increment and collect feedback.

Tools: Slides (optional), Demonstration video (optional), Agile board

Tasks:

a. Prepare a sprint demonstration of the completed stories (working increment).

b. Gather feedback from peers and the instructor, and document the corresponding action items.
c. Revise the backlog in accordance with feedback and adjust priorities accordingly.

Expected Outputs:
• Documentation of sprint review notes and a comprehensive feedback record.
• Revised backlog screenshot.
• Demonstration evidence (screenshots/video link).

Assignment 8	**Sprint Retrospective and Continuous Improvement**	**1 Slot**

Topic name: Reflect on process and define improvements for next sprint.

Tools: Slides (optional), Demonstration video (optional), Agile board

Tasks:

a. Execute a retrospective utilising the Start-Stop-Continue framework or the 4Ls (Liked, Learnt, Lacked, Longed for).
b. Recognize a minimum of three enhancements and translate them into executable tasks.
c. Identify a singular process metric for enhancement (cycle time, faults, throughput).

Expected Outputs:
• Retrospective analysis.
• Action items for improvement are on the agenda.
• Definition of the metric and its baseline value.

Reference Books :

R1: Agile Estimating and Planning, Mike Cohn

R2: Scrum: The Art of Doing Twice the Work in Half the Time**,** Jeff Sutherland

R3: Succeeding with Agile: Software Development Using Scrum, Mike Cohn

R4: Agile Testing: A Practical Guide for Testers and Agile Teams, Lisa Crispin, Janet Gregory

Savitribai Phule Pune University

T.Y.B.Sc. (Computer Science) - Sem – VI

Course Type: FP/OJT/CEP **Course Code: CS-381-OJT**

Course Title:On Job Training (OJT)

Teaching Scheme	No. of Credits	Examination Scheme
8 Hours/Week	4	CE:30 marks EE:70 marks

Prerequisites

- Basic knowledge of Programming concepts (C / Java / Python),Database fundamentals, Web technologies (HTML, CSS, basic scripting)
- Understanding of Software Development Life Cycle (SDLC), Basic computer networks and operating systems
- Willingness to follow industry ethics and professional conduct.

Course Objectives: -

The objectives of the On-the-Job Training course are to:

1. Provide students with hands-on industry exposure relevant to Information Technology.
2. Enable learners to apply theoretical knowledge to real-world IT problems.
3. Develop professional skills such as teamwork, communication, and time management.
4. Familiarize students with industry tools, technologies, and workflows.
5. Enhance employability and career readiness through practical experience.

Course Outcomes: -

After successful completion of OJT, the student will be able to:

CO1: Apply IT theories and concepts to solve real-time organizational problems.

CO2: Use appropriate software tools, technologies, and platforms used in the IT industry.

CO3: Analyze requirements, design solutions, and implement applications under guidance.

CO4: Demonstrate professional behavior, communication skills, and teamwork.

CO5: Prepare technical documentation and present work effectively.

CO6: Evaluate organizational practices, ethics, and real-world IT workflows.

Guidelines

1. Duration: 4–6 Weeks (Minimum 120–160 Hours)
2. Nature: Mandatory / Skill-based / Industry-oriented
3. IT Companies / Software Firms
   - Startups / MSMEs
   - Government or Private Organizations (IT departments)
   - Research Institutes / Incubation Centers
   - Recognized Training Centers with industry projects
4. Training must be relevant to Information Technology.
5. AREAS OF OJT

- Software Development
- Web Application Development
- Mobile App Development
- Database Management
- Networking & System Administration
- Cloud Computing
- Cyber Security

- Data Analytics / AI basics
- IT Support & Maintenance

INTERNAL ASSESSMENT (30 MARKS)

Internal assessment shall be conducted by the internal faculty coordinator.

Component	Marks
Attendance & Discipline	05
Weekly Progress Report / Logbook	10
Guide / Industry Supervisor Feedback	10
Presentation / Review	05
Total Internal Marks	**30**

EXTERNAL ASSESSMENT (70 MARKS)

External evaluation shall be conducted by an external examiner appointed by the university.

Component	Marks
OJT Final Report (Documentation & Format)	20
Project / Task Implementation	25
Viva-Voce (External Examiner)	15
Industry Exposure & Learning Outcome	10
Total External Marks	**70**

OJT REPORT CONTENTS

1. Certificate from Organization
2. Student Declaration
3. Organization Profile
4. Objectives of OJT
5. System / Project Overview
6. Weekly Work Description
7. Tools & Technologies Used
8. Learning Outcomes
9. Screenshots / Code Samples (if applicable)
10. Challenges & Solutions
11. Conclusion
12. References