Anekant Education Society's

Tuljaram Chaturchand College, Baramati

(Autonomous)

Three Year B.Sc. Degree Program in Computer Science

(Faculty of Science & Technology)

CBCS Syllabus (2019 Pattern)

S.Y. B.Sc. (Computer Science) Sem- IV

For Department of Computer Science

Tuljaram Chaturchand College of Arts, Science & Commerce, Baramati

Class : F.Y.B.Sc. (Computer Science)			
Semester I		**Semester II**	
CSCO 1101	Basic Programming using C	CSCO 1201	Advanced Programming using C
CSCO1102	DBMS-I	CSCO 1202	DBMS-II
CSCO1103	Lab Course I : Basics of C	CSCO 1203	Lab Course I : Advanced C
CSCO1104	Lab Course II : DBMS I	CSCO1204	Lab Course II : DBMS II
Physical Education			

Class :S.Y.B.Sc. (Computer Science)			
Semester III		**Semester IV**	
CSCO 2301	Data Structures using C	CSCO2401	Object Oriented Concepts using Java
CSCO2302	Introduction to Web Technology	CSCO2402	Software Engineering
CSCO2303	Lab Course I : Based On CSCO2301	CSCO2403	Lab Course I: Based On 2401
CSCO2304	Lab Course II: based On CSCO2302	CSCO2404	Lab Course II : Based On CSCO2402 with Mini Project
Certificate Course I		Certificate Course II	
Environment Science (EVS) An Educational Trip conduct in IV semester			

Class: T.Y.B.Sc. (Computer Science)			
Semester V		**Semester VI**	
CSCO3501	System Programming &Operating System	CSCO3601	Advanced Operating System
CSCO 3502	Theoretical Computer Science	CSCO3602	Compiler Construction
CSCO3503	Computer Networks - I	CSCO3603	Computer Networks - II
CSCO3504	Web Development – I	CSCO3604	Web Development–II
CSCO3505	Advanced Programming in Java	CSCO3605	Advanced Java Technologies – Frameworks
CSCO3506	Object Oriented Software Engineering	CSCO3606	Software Metrics & Project Management
CSCO3507	Lab Course I: Based on CSCO3501	CSCO3607	Lab Course I: Based on CSCO3601
CSCO3508	Lab Course II: Based on CSCO3505	CSCO3608	Lab Course II: Based on CSCO3605 & Mini Project using JAVA
CSCO3509	Lab Course III: Based on CSCO3504	CSCO3609	Lab Course III: Based on CSCO3604 & Mini Project using PHP.
Certificate Course III		An Educational Trip conduct in this semester.	

S.Y.B.Sc.(Computer Science)

Semester-IV

Credit Structure & Syllabus

(Academic Year 2020-2021, Autonomous)

Course Structure for S. Y. B. Sc. (Computer Science) Sem-III & IV

Subject: Computer Science

Sem	Paper Code	Title of Paper	No. of Credits	Exam	Marks
III	CSCO2301	Data Structure Using C	3	I / E	60 + 40
	CSCO2302	Introduction to Web Technology	3	I / E	60 + 40
	CSCO2303	Lab Course – I based On Data structure	2	I / E	60 + 40
	CSCO2304	Lab Course – II Based on Web Technology	Grade	I/E	60 +40
		Certificate Course I	2	---	----
IV	CSCO 2401	Object Oriented Concepts using Java	3	I / E	60 + 40
	CSCO2402	Software Engineering	3	I / E	60 + 40
	CSCO2403	Lab Course – I on CSCO2401	2	I / E	60 + 40
	CSCO2404	Lab Course– II CSCO2402 with S.E. Project.	Grade	I/E	60 + 40
		Certificate Course II	2	----	----
		Environment Science	2	----	----

Class: S.Y. B. Sc.(Computer Science) (Semester- IV)
W.e.f. A.Y.- 2020-2021

Subject: Computer Science **Paper Code:** CSCO2401
Title of Paper: Object Oriented Concepts using Java **Paper:** I
Credit: 3(4 lectures /Week) **No. of Lectures:** 48

Prerequisites:

- Knowledge of C Programming Language.

Objective:

- To learn Object Oriented Programming language
- To handle abnormal termination of a program using exception handling
- To handle complex problems using object oriented concepts
- To design programs using multithreading

Learning Outcomes:

Course Outcomes: On completion of the course, student will be able to
CO1: Understand Object Oriented Concepts
CO2: Read input from different ways.
CO3: Define class with different access modifiers and create object.
CO4: Write implement real world problems using Java
CO5: Handle exceptions during programs
CO6: Handle String efficiently
CO7: Handle file with different modes and formats

1. An Introduction to Java [4]

1.1 A Short History of Java
1.2 Features or buzzwords of Java
1.3 Comparison of Java and C++
1.4 Java Environment
1.5 Simple java program
1.6 Java Tools – jdb, javap, javadoc
1.7 Java IDE – Eclipse/NetBeans (Note: Only for Lab Demonstration)

2. An Overview of Java [4]

2.1 Types of Comments
2.2 Data Types
2.3 Final Variable
2.4 Declaring 1D, 2D array
2.5 Accepting input using Command line argument
2.6 Accepting input from console (Using Buffered Reader class)

3. Objects and Classes [8]

3.1 Defining Your Own Classes
3.2 Access Specifiers (public, protected, private, default)
3.3 Array of Objects
3.4 Constructor, Overloading Constructors and use of 'this' Keyword
3.5 static block, static Fields and methods
3.6 Predefined class – Object class methods (equals(), toString(), hashcode(), getClass())
3.7 Inner class 3.8 Creating, Accessing and using Packages
3.9 Creating jar file and manifest file
3.10 Wrapper Classes
3.11 Garbage Collection (finalize() Method)
3.12 Date and time processing

4. Inheritance and Interface [7]

4.1 Inheritance Basics (extends Keyword) and Types of Inheritance
4.2 Superclass, Subclass and use of Super Keyword
4.3 Method Overriding and runtime polymorphism
4.4 Use of final keyword related to method and class
4.5 Use of abstract class and abstract methods
4.6 Defining and Implementing Interfaces

4.7 Runtime polymorphism using interface
4.7 Object Cloning

5. Exception Handling [4]

5.1 Dealing Errors
5.2 Exception class, Checked and Unchecked exception
5.3 Catching exception and exception handling
5.4 Creating user defined exception
5.5 Assertions

6. Strings, Streams and Files [7]

6.1 String class and String Buffer Class
6.2 Formatting string data using format() method
6.2 Using the File class
6.3 Stream classes Byte Stream classes Character Stream Classes
6.4 Creation of files
6.5 Reading/Writing characters and bytes
6.6 Handling primitive data types
6.7 Random Access files

7. Collection [6]

7.1 Introduction to the Collection framework
7.2 List – Array List, Linked List and Vector, Stack, Queue
7.3 Set – HashSet, TreeSet, and Linked HashSet
7.4 Map – HashMap, Linked Hash Map, Hash table and Tree Map
7.5 Interfaces such as Comparator, Iterator, ListIterator, Enumeration

8. Multithreading [8]

8.1 What are threads?
8.2 Life cycle of thread
8.3 Running and starting thread using Thread class
8.4 Thread priorities
8.5 Running multiple threads
8.6 The Run able interface
8.7 Synchronization and interthread communication

References:

1) Complete reference Java by Herbert Schildt (5th edition)

2) Java 2 programming black books, Steven Horlzner

3) Programming with Java , A primer ,Forth edition , By E. Balagurusamy

4) Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press

Mapping of this course with Programme Outcomes

Course Outcomes	Program Outcomes						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	3	3	3	1	1	2	2
CO2	3	3	3	1	1	2	2
CO3	3	3	3	1	1	2	2
CO4	3	3	3	1	1	2	2
CO5	3	3	3	1	1	2	2
CO6	3	3	3	1	2	2	2
CO7	3	3	3	1	2	3	3

1. PO1 with All COs

CO1: PO1:3 (Strongly related): as computer system simulation involves applying mathematical and computational fundamentals to understand and model IT systems.

CO2: PO1: 3 (Strongly related): as implementing assembly programs requires a deep understanding of computer fundamentals and mathematics.

CO3: PO1: 3 (Strongly related): as working with shell commands and system calls requires knowledge of computer fundamentals and IT applications.

CO4:PO1: 2(Strongly related): as CPU scheduling algorithms involve applying mathematical and computational principles to optimize system performance.

CO5: PO1: 2 (Strongly related): as shell scripting and command-line interfaces are integral parts of IT applications, requiring computer knowledge.

CO6: PO1: 3 (Strongly related): as implementing the Banker's algorithm involves applying mathematical and computational concepts to address system deadlock issues.

CO7: PO1: 3 (Strongly related): as troubleshooting in these areas requires applying computer knowledge and problem-solving skills.

2. PO2 with All COs

CO1: PO2: 2 (Strongly related): as understanding computer system simulation is foundational to designing and developing solutions in various IT contexts.

CO2: PO2: 3 (Strongly related): as the ability to implement assembly programs is a key aspect of designing and developing solutions at a low level.

CO3: PO2: 3 (Strongly related): as familiarity with shell commands and system calls is essential for designing and developing solutions involving command-line interfaces.

CO4: PO2: 2 (Strongly related): as knowledge of CPU scheduling algorithms is crucial for designing efficient and optimized system solutions.

CO5: PO2: 2(Strongly related): as proficiency in shell scripting and command-line interfaces is valuable in designing and developing practical solutions.

CO6: PO2: 2 (Strongly related): as implementing the Banker's algorithm is a specific design solution for deadlock avoidance in system development.

CO7: PO2: 3 (Strongly related): as troubleshooting skills are essential for the ongoing development and maintenance of solutions.

3. PO3 with All COs

CO1: PO3: 3 (Strongly related): as simulation tools are modern tools extensively used for understanding and modeling complex computer systems.

CO2: PO3: 3(Strongly related) as modern tools for assembly programming are essential for efficient and error-free implementation.

CO3: PO3: 3 (Strongly related): as proficiency in using modern command-line tools is crucial for effective utilization of shell commands and system calls.

CO4: PO3: 3(Strongly related): as modern tools are employed to simulate and analyze the performance of various CPU scheduling algorithms.

CO5: PO3: 3 (Strongly related): as modern tools and editors are commonly used for efficient development and execution of shell scripts.

CO6: PO3: 3 (Strongly related): as the implementation and simulation of algorithms often involve the use of modern programming tools and environments.

CO7: PO3: 3 (Strongly related): as modern debugging and profiling tools are essential for effective troubleshooting in various areas of system development.

4. PO4 with All COs

CO1: PO4: 1(Partially related): as computer system simulation, is more aligned with technical aspects than direct environmental and sustainability considerations.

CO2: PO4: 1(Partially related): as assembly programming focuses on technical skills rather than direct implications for environmental and sustainability concerns.

CO3: PO4: 1(Partially related): as shell commands and system calls are more technical in nature and have limited direct connection to environmental and sustainability aspects.

CO4: PO4: 1 (Partially related): as CPU scheduling algorithms are primarily technical and do not have a strong direct link to environmental and sustainability considerations.

CO5: PO4: 1 (Partially related): as shell scripting and command-line interfaces are technical skills with limited direct impact on environmental and sustainability aspects.

CO6:PO4: 1(Partially related): as the Banker's algorithm focuses on technical aspects of deadlock avoidance rather than environmental or sustainability implications.

CO7: PO4: 1 (Partially related): as troubleshooting skills are more aligned with technical problem-solving and have limited direct connection to environmental and sustainability concerns.

5. PO5 with All COs

CO1: PO5: 1 (Partially related): as computer system simulation is more aligned with technical aspects, and its connection to ethical principles is indirect.

CO2PO5: 1(Partially related): as assembly programming primarily focuses on technical skills rather than direct ethical considerations.

CO3: PO5: 1 (Partially related): as shell commands and system calls are technical in nature and have limited direct connection to ethical principles.

CO4: PO5: 1 (Partially related): as CPU scheduling algorithms are primarily technical and do not have a strong direct link to ethical considerations.

CO5: PO5: 1(Partially related): as shell scripting and command-line interfaces are technical skills with limited direct impact on ethical principles.

CO6: PO5: 2 (Moderately related): as the implementation of the Banker's algorithm may involve considerations related to ethical and responsible programming practices.

CO7: PO5: 2 (Moderately related) as troubleshooting involves ethical considerations, such as maintaining the integrity and security of systems.

6. PO6 with All COs

CO1: PO6: 2(Moderately related): as collaborative efforts may be involved in designing and interpreting simulations.

CO2: PO6: 2 (Moderately related): as teamwork may be required for collaborative coding, code reviews, or troubleshooting.

CO3: PO6: 2 (Moderately related): as working on command-line interfaces and scripting may involve collaboration and knowledge sharing within a team.

CO4: PO6: 2 (Moderately related): as understanding and implementing scheduling algorithms may require teamwork for analysis and optimization.

CO5: PO6: 2 (Moderately related): as collaboration and sharing of scripts within a team may be necessary for effective system management.

CO6: PO6: 2(Moderately related): as collaborative efforts may be needed to implement and test the Banker's algorithm in a simulated environment.

CO7: PO6: 3(Strongly related): as troubleshooting often involves collaboration and collective problem-solving within a team.

7. PO7 with All COs

CO1: PO7: 2 (Moderately related): as simulation skills may contribute to innovative problem-solving and employability in technical roles.

CO2: PO7: 2 (Moderately related): as assembly programming skills may enhance employability in technical fields and contribute to innovative solutions.

CO3: PO7: 2 (Moderately related): as proficiency in shell commands and system calls is valuable for employability and innovation in system administration and development.

CO4: PO7: 2 (Moderately related):as knowledge of CPU scheduling algorithms can contribute to innovative solutions and employability in system optimization roles.

CO5:PO7: 2 (Moderately related): as scripting skills are often sought after in IT roles, contributing to employability and potential innovation in automation.

CO6: PO7: 2 (Moderately related): as implementation of algorithms demonstrates technical competence relevant to employability and innovation.

CO7: PO7: 3 (Strongly related): as troubleshooting skills are critical for employability.

Class: S.Y. B. Sc.(Computer Science) (Semester- IV)

Subject: Computer Science **Paper Code:**CSCO2402

Title of Paper: Software Engineering **Paper:** II

Credit: 3 (4 Lectures/Week) **No. of lectures:** 48

Prerequisites:

- Basic knowledge of DBMS & RDBMS.
- Knowledge of C HTML5, CSS & JAVASCRIPT.

Objective:

- To teach basics of System Analysis and Design.
- To teach principles of Software Engineering.
- To teach various process models used in practice.
- To know about the system engineering and requirement engineering.
- To build analyze is model.

Learning Outcomes:

CO1: Identify the type of system and its solution from the real-life problems or case studies.

CO2: Implement the complete and thorough knowledge of software engineering principles and practices.

CO3: Master over the complete process of Software Development Life Cycle (SDLC).

CO4: Explore all the Process Models of software engineering in detail.

CO5: Identify and Apply requirement engineering concepts for solving the real-life problems or case studies.

CO6: Identify and Apply structured analysis and data flow analysis techniques.

CO7: Explore agility concept and XP process model in detail.

1. SystemConcepts [5] (R1: Chapter 1 & R3: Chapter 1)

- **1.1** System Definition
- **1.2** Characteristics of a System: Organization, Subsystem, Interaction, Interdependence, Integration, Central objective, Standards, Black box.
- **1.3** Elements of a system: Outputs, Inputs, Processor(s), Control, Feedback, Environment, Boundaries, Interface.
- **1.4** Types of Systems: Physical & Abstract Systems, Open & Closed Systems, Computer-based Systems (MIS : Management Information System & DSS : Decision Support System)

2. Software and Software Engineering [5] (R2 : Chapter 1)

- **2.1** The Nature of Software
  - **2.1.1** Defining Software
  - **2.1.2** Software Application Domains
  - **2.1.3** Legacy Software
- **2.2** Software Engineering
- **2.3** The Software Process
- **2.4** Software Engineering Practice
  - **2.4.1** The Essence of Practice
  - **2.4.2** General Principles
- **2.5** Software Myths

3. System Development Life Cycle (SDLC) [8] (R3 : Chapter 1)

- **3.1** Introduction
- **3.2** Activities of SDLC
  - **3.2.1** Preliminary Investigation (Request Clarification, Feasibility Study,

Request Approval)

3.2.2 Determination of System Requirements

3.2.3 Design of System

3.2.4 Development of Software

3.2.5 System Testing (Unit Testing, Integration testing, System Testing)

3.2.6 System Implementation &Evaluation

3.2.7 System Maintenance

4. Process Models [6] (R2 : Chapter 2)

4.1 A Generic Process Model

4.2 Prescriptive Process Models

4.2.1 The Waterfall Model

4.2.2 Incremental Process Models

4.2.3 Evolutionary Process Models

4.2.3.1 Prototyping

4.2.3.2 Spiral Model

4.2.4 Concurrent Models

5. Requirements Engineering [8] (R2 : Chapter 5)

5.1 Introduction

5.2 Requirements Engineering Tasks

5.2.1 Inception ,Elicitation , Elaboration , Negotiation ,Specification, Validation, Requirements Management

5.3 Initiating the Requirements Engineering Process

5.3.1 Identifying the Stakeholders

5.3.2 Recognizing Multiple Viewpoints

5.3.3 Working toward Collaboration

5.4 Fact Finding Techniques**(R3 : Chapter 3)**

5.4.1 Interview ,Questionnaire , Record Review , Observation

6. Structured Analysis Development Strategy [10] (R3 : Chapter 4)

6.1 Structured Analysis

6.1.1 Structured Analysis?

6.1.2 Components of Structured Analysis

6.1.3 Data Flow Analysis?

6.1.4 Features & Tools of Data Flow Analysis

6.1.5 Logical Data Flow Diagram (Logical DFD)

6.1.6 Physical Data Flow Diagram

6.1.6.1 Notations

6.1.6.2 Drawing a Context Diagram

6.1.6.3 Exploding a Context diagram into Greater detail (1^{st} level, 2^{nd} Level Defect…)

6.1.6.4 Evaluating Data Flow Diagram for Correctness

6.1.7 A Data Dictionary

6.1.7.1 Concepts of Data Dictionary

6.1.7.2 Importance of Dictionary

6.1.7.3 Function of Data Dictionary

7. An Agile View of Process [6] (R2 : Chapter3)

7.1 Introduction Agility?

7.2 Introduction to Agile Process?

7.2.1 The Politics of Agile Development

7.2.2 Human Factors

7.2.3 Agile Process Models

7.2.4 Extreme Programming(XP)

7.2.5 Adaptive Software Development(ASD)

7.2.6 Dynamic Systems Development Method(DSDM)

Reference Books:

R1: System Analysis and Design (Second Edition) by Elias M. Awad, Galgotia Publications Pvt. Ltd.

R2: Software Engineering: A Practitioner's Approach (Seventh Edition) by Roger S.Pressman, McGraw Hill International Edition.

R3: Analysis and Design of Information Systems (Second Edition) by James A. Senn, McGraw Hill International Editions.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)						
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**
CO1	3	3	3	2	3	3	3
CO2	3	3	2	2	3	3	3
CO3	3	3	3	2	3	3	3
CO4	2	2	2	2	3	3	3
CO5	3	3	3	3	3	3	3
CO6	3	3	3	2	3	3	3
CO7	2	2	2	2	3	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

CO1:PO1:(Strongly Relates 3)This CO aligns with the application of computer fundamentals and problem-solving skills in real-life situations, which is a key aspect of PO1.

CO2:PO1:(Strongly Relates 3)This CO is directly related to applying knowledge of software engineering principles, which includes mathematics and computer fundamentals as outlined in PO1.

CO3:PO1:(Strongly Relates 3)The Software Development Life Cycle involves various stages where mathematical, statistical, and computer fundamentals are applied, aligning with the skills outlined in PO1.

CO4:PO1:(Moderately Related 2)While understanding process models involves a knowledge of software engineering, it may not directly require the application of mathematics, statistics, or computer fundamentals in every aspect.

CO5:PO1:(Strongly Relates 3)Requirement engineering involves understanding and solving real-life problems, connecting it strongly with the application of mathematics, statistics, and computer fundamentals.

CO6:PO1:(Strongly Relates 3)This CO aligns with the application of structured analysis techniques, which includes a solid understanding of mathematics and computer fundamentals as per PO1.

CO7:PO1:(Moderately Related 2)While agility concepts and XP process models are part of software engineering, they may not be directly tied to the application of mathematics, statistics, or computer fundamentals in all aspects.

Mapping of PO2 with All CO'S

CO1:PO2:(Strongly Relates 3)Designing solutions for IT applications involves the identification of system types and solutions, aligning with the objective of designing solutions using the latest technologies as stated in PO2.

CO2:PO2:(Strongly Relates 3)The implementation of solutions using various languages is a key aspect of software engineering principles, connecting strongly with the objective of implementing knowledge of software engineering in PO2.

CO3:PO2:(Strongly Relates 3)Designing solutions involves a thorough understanding and mastery over the Software Development Life Cycle, linking closely with the objective of mastering the SDLC process in PO2.
CO4:PO2:(Moderately Related 2)While exploring process models is part of designing solutions, it may not be directly tied to the application of the latest technologies in all aspects.
CO5:PO2:(Strongly Relates 3)Designing solutions involves identifying and applying requirement engineering concepts, aligning strongly with the objective of designing solutions for IT applications in PO2.
CO6: PO2: Weightage 3 (Strongly Relates 3)Designing solutions requires the application of structured analysis and data flow analysis techniques, connecting closely with the objective of designing solutions in PO2.
CO7: PO2: Weightage 2 (Moderately Related)While exploring agility concepts and XP process models may be part of designing solutions, it may not be directly tied to the use of the latest technologies in all aspects.

Mapping of PO3 with All CO'S

CO1: PO3: (Strongly Relates 3)The use of modern tools often involves identifying system types and solutions from real-life problems, connecting strongly with the objective of modern tool usage in PO3.
CO2: PO3: (Moderately Related 2)Modern tool usage involves implementing knowledge, but it may not be directly tied to the application of software engineering principles in all aspects.
CO3: PO :(Strongly Relates 3)The use of modern tools is closely tied to mastering the SDLC process, aligning with the objective of modern tool usage in PO3.
CO4: PO3 : (Moderately Related 2)While exploring process models may be part of modern tool usage, it may not be the primary focus of selecting and applying modern engineering and IT tools.
CO5: PO3 : (Strongly Relates 3)Modern tool usage often involves applying requirement engineering concepts, aligning closely with the objective of modern tool usage in PO3.
CO6: PO3: (Strongly Relates 3)The use of modern tools often includes applying structured analysis and data flow analysis techniques, connecting strongly with the objective of modern tool usage in PO3.
CO7: PO3 : (Moderately Related 2)Exploring agility concepts and XP process models may be part of modern tool usage, but it may not be the primary focus of selecting and applying modern engineering and IT tools.

Mapping of PO4 with All CO'S

CO1: PO4 : (Moderately Related 2)While understanding the impact of IT solutions on societal and environmental contexts is important, directly linking this to the identification of system types and solutions may be a moderate relationship.
CO2: PO4 : (Moderately Related 2)The implementation of software engineering principles may have indirect connections to understanding the impact of solutions on societal and environmental contexts.
CO3: PO4 : (Moderately Related 2)While the mastery of SDLC is crucial for effective software development, the direct connection to environmental and societal impact may be moderate.
CO4: PO4 : (Moderately Related 2)Exploring process models in software engineering may not directly align with understanding the impact on societal and environmental contexts, but there could be indirect connections.
CO5: PO4: (Strongly Relates 3)Requirement engineering often involves understanding real-life problems, and connecting this with the impact on societal and environmental contexts is a strong relationship.
CO6: PO4 : (Moderately Related 2)While the application of analysis techniques is vital in software development, the direct link to environmental and societal impact may be moderate.
CO7: PO4 : (Moderately Related 2)Exploring agility concepts and XP process models may not have a direct link to environmental and societal impact, but there could be some indirect connections.

Mapping of PO5 with All CO'S

CO1: PO5: (Strongly Relates 3)The application of ethical principles and commitment to professional ethics is closely related to identifying system types and solutions in real-life situations.

CO2: PO5 : (Strongly Relates 3)Applying ethical principles is crucial in the implementation of software engineering principles and practices, forming a strong relationship.

CO3: PO5: (Strongly Relates 3)Mastering SDLC involves ethical considerations at various stages, making the relationship strong.

CO4: PO5: Weightage 3 (Strongly Relates)Ethical considerations are important when exploring and selecting process models in software engineering, resulting in a strong relationship.

CO5: PO5 : Weightage 3 (Strongly Relates)Requirement engineering often involves ethical considerations, forming a strong connection with the application of ethical principles.

CO6: PO5: Weightage 3 (Strongly Relates)Ethical principles apply throughout the software development process, including the application of analysis techniques, creating a strong relationship.

CO7: PO5: Weightage 3 (Strongly Relates)Understanding and exploring agility concepts and XP process models require ethical considerations, establishing a strong connection.

Mapping of PO6 with All CO'S

CO1: PO6: Weightage 3 (Strongly Relates)Functioning effectively in individual and team settings is closely tied to the ability to identify system types and solutions in real-life scenarios.

CO2: PO6: Weightage 3 (Strongly Relates)Effective individual and team work is essential in implementing software engineering principles and practices.

CO3: PO6: Weightage 3 (Strongly Relates)Mastering SDLC requires both individual effectiveness and the ability to work in teams and multidisciplinary settings.

CO4: PO6: Weightage 3 (Strongly Relates)Exploring process models involves collaboration within a team, establishing a strong relationship with the proficiency objective.

CO5: PO6: Weightage 3 (Strongly Relates)Applying requirement engineering concepts involves both individual and team efforts, forming a strong connection with the proficiency objective.

CO6: PO6 : Weightage 3 (Strongly Relates)Applying analysis techniques often requires collaboration within a team, aligning well with the proficiency objective.

CO7: PO6 : Weightage 3 (Strongly Relates)Exploring agility concepts and XP process models involves teamwork and effective collaboration, establishing a strong relationship with the proficiency objective.

Mapping of PO7 with All CO'S

CO1:PO7: Weightage 3 (Strongly Relates)Identifying opportunities and pursuing them to create value aligns with the skill of identifying system types and solutions in real-life scenarios.

CO2:PO7: Weightage 3 (Strongly Relates)Implementing software engineering principles and practices is crucial for pursuing opportunities and creating value, linking strongly with the proficiency objective.

CO3:PO7: Weightage 3 (Strongly Relates)Mastering SDLC is essential for developing the capacity to study and research independently, contributing to the skills needed for entrepreneurial endeavors.

CO4:PO7: Weightage 3 (Strongly Relates)Exploring process models is relevant to understanding and pursuing opportunities, contributing to innovation and entrepreneurial skills.

CO5:PO7: Weightage 3 (Strongly Relates)Applying requirement engineering concepts is crucial for identifying and solving real-life problems, aligning strongly with the innovation and entrepreneurial skills objective.

CO6:PO7: Weightage 3 (Strongly Relates) Applying analysis techniques is relevant to innovation and entrepreneurial skills, as it contributes to problem-solving and value creation.

CO7: PO7: Weightage 3 (Strongly Relates) Exploring agility concepts and XP process models is relevant to innovation and adaptability, supporting entrepreneurial skills.

Class: S.Y.B.Sc.(Computer Science) (Semester- IV)

Subject: Computer Science **Paper Code:** CSCO2403
Title of Paper: Lab course-I On Programming in Java **Paper:** III (Lab Course-I)
Credit: 2 (3 Hour Practical /Batch/Week) **No. of Practicals:** 12

Prerequisites:

- ✓ Basic knowledge of Object-Oriented Programming.
- ✓ Knowledge of Java Programming Language.

Learning Outcomes: On completion of the course, student will be able to
CO1.On completion of the course, student will be able to Understand Object Oriented Concepts
CO2. Understand the difference between Java & CPP
CO3. Read input from different ways.
CO4. Define class with different access modifiers and create object.
CO5.Write real world problems using Java
CO6. Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
CO7. Demonstrate the use of various OOPs concepts with the help of programs.

Sr. No	Assignment Title
1	**Java Tools**
2	**Classes , Methods , Objects**
3	**Array of Objects, Access Modifiers**
4	**Packages**
5	**Inheritance**
6	**Interfaces**
7	**Exception Handling and Assertions**
8	**I/O and String Handling**
9	**File Handling**
10	**Collection**
11	**Multithreading**
12	**Multithreading**

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)						
	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	2	2	3	3	2	3	3
CO2	2	1	2	2	2	3	3
CO3	3	1	2	2	1	2	2
CO4	2	2	3	3	2	3	2
CO5	2	2	3	3	2	3	3
CO6	3	3	3	3	2	3	3
CO7	3	3	3	3	2	3	3

Mapping of PO1 With All CO's with Justification:

CO1: With PO1: Object-oriented concepts are foundational to programming languages and software development, contributing to the broader knowledge areas mentioned in PO1.

CO2: With PO1: Understanding the difference between Java and C++ is language-specific and may not directly contribute to the comprehensive knowledge enrichment across diverse IT domains listed in PO1.

CO3: With PO1: Reading input is a basic programming skill, but it may not directly align with the specified knowledge areas in PO1.

CO4: With PO1: Creating classes with access modifiers and instantiating objects are fundamental OOP concepts that contribute to understanding programming languages, though not as directly aligned with the diverse IT domains in PO1.

CO5: With PO1: Writing real-world problems in Java is specific to programming and contributes to knowledge in programming languages, but it may not directly cover the wide range of IT domains listed in PO1.

CO6: WithPO1: Understanding inheritance and related concepts contributes to programming knowledge, though it may not directly encompass the broad range of IT domains in PO1.

CO7: With PO1 : Demonstrating OOPs concepts through programs is specific to programming and contributes to knowledge in programming languages, though it may not directly cover all IT domains in PO1.

Mapping of PO2 With All CO's with Justification:

CO1: WithPO2 :A strong understanding of Object Oriented Concepts is fundamental to comprehending various dimensions of software application and project development.

CO2: WithPO2 : Understanding the differences between Java and C++ is language-specific and may not be directly tied to grasping all dimensions of software application and project concepts.

CO3: With PO2 : Reading input is a basic programming skill but may not directly contribute to a comprehensive understanding of all dimensions of software application and project concepts.

CO4: With PO2 : Defining classes with access modifiers and creating objects contributes to a foundational understanding of software development but may not cover all dimensions of software applications and projects.

CO5: WithPO2 : Writing real-world problems in Java is specific to programming and contributes to understanding software development, but it may not directly cover all dimensions of software applications and projects.

CO6: With PO2 : Understanding inheritance and related concepts is crucial for a comprehensive grasp of various dimensions within software application and project development.

CO7: With PO2 : Demonstrating OOPs concepts through programs is directly aligned with practical applications and contributes to a deep understanding of multiple dimensions within software application and project development.

Mapping of PO3 With All CO's with Justification:

CO1: With PO3 : Demonstrating Object Oriented Concepts with the use of ICT aligns with the goal of understanding computer subjects through practical applications.

CO2: WithPO3 : Understanding the difference between Java and CPP is relevant to computer subjects, and demonstrating this knowledge through ICT usage contributes to a practical understanding.

CO3: WithPO3 : Reading input using various methods is a programming skill that, when demonstrated through ICT, contributes to the practical understanding of computer subjects.

CO4: With PO3 : Defining classes with access modifiers and creating objects, when demonstrated using ICT, enhances the practical understanding of computer subjects, especially in programming.

CO5: WithPO3 : Writing real-world problems in Java and demonstrating these programs with ICT aligns with the goal of understanding computer subjects through practical application.

CO6: WithPO3 : Classifying inheritance and related concepts with ICT-based demonstrations enhances the practical understanding of computer subjects, combining theoretical knowledge with hands-on application.

CO7: WithPO3 : Demonstrating the use of OOPs concepts through programs with the help of ICT aligns with the overall goal of understanding computer subjects through practical applications.

Mapping of PO4 With All CO's with Justification:

CO1: With PO4 : A strong understanding of Object Oriented Concepts is essential for developing in-house applications, forming the foundation for effective software design and implementation.

CO2: WithPO4 : Understanding the difference between Java and CPP can be relevant to choosing the appropriate programming language for in-house applications, contributing to informed decision-making in project development.

CO3: WithPO4 : Reading input from different sources is a fundamental skill in programming, contributing to the ability to handle diverse input scenarios when developing in-house applications.

CO4: With PO4: Defining classes with various access modifiers and creating objects is crucial for structuring and implementing in-house applications in an organized and efficient manner.

CO5: With PO4: Writing real-world problems using Java is directly applicable to developing in-house applications, demonstrating the practical implementation of programming concepts in a project setting.

CO6: WithPO4 : Classifying inheritance and related concepts is vital for designing the architecture of in-house applications, ensuring efficient code organization and maintenance.

CO7: WithPO4 : Demonstrating OOPs concepts through programs is directly aligned with developing in-house applications, showcasing the practical application of theoretical knowledge in project scenarios.

Mapping of PO5 With All CO's with Justification:

CO1: WithPO5 : Interacting with IT experts during visits may provide insights into real-world applications of Object Oriented Concepts, enhancing understanding beyond theoretical knowledge.

CO2: WithPO5 : Interacting with IT experts during visits can offer opportunities to discuss and understand the practical implications of choosing between Java and CPP in real-world scenarios.

CO3: With PO5 : Interacting with IT experts during visits may not be directly tied to the skill of reading input from different ways, as this is more focused on programming skills.

CO4: With PO5 : Interacting with IT experts during visits may provide insights into the practical aspects of defining classes with various access modifiers and creating objects in real-world projects.

CO5: WithPO5 : Interacting with IT experts during visits may involve discussions on real-world problems and their solutions using Java, contributing to practical insights into project development.

CO6: With PO5 : Interacting with IT experts during visits may offer opportunities to discuss and understand the practical aspects of inheritance, binding, exception handling, and generic programming in real-world projects.

CO7: WithPO5: Interacting with IT experts during visits may involve demonstrating OOPs concepts in practical scenarios, providing insights into their application in real-world projects.

Mapping of PO6 With All CO's with Justification:

CO1: With PO6 : The industrial internship provides practical exposure, allowing students to apply and deepen their understanding of Object Oriented Concepts in real-world IT projects.

CO2: WithPO6 :The industrial internship offers an opportunity to gain firsthand experience in the IT industry, allowing students to understand and potentially work with both Java and CPP, contributing to a practical understanding of their differences.

CO3: With PO6 : While the industrial internship may involve various tasks, including reading input from different sources, it may not be the primary focus, making it moderately related to the skill of reading input.

CO4: With PO6 : The industrial internship allows students to actively participate in defining classes with access modifiers and creating objects, contributing to their practical skills in software development.

CO5: With PO6: The industrial internship involves working on real-world problems, providing an excellent opportunity for students to apply their Java programming skills in practical scenarios.

CO6: With PO6 : The industrial internship allows students to classify inheritance and understand its applications, as well as gain practical experience in early and late binding, exception handling, and generic programming.

CO7: WithPO6 : The industrial internship provides a platform for students to actively demonstrate the application of various OOPs concepts through real-world programming projects in an industry setting.

Mapping of PO7 With All CO's with Justification:

CO1: With PO7: A strong understanding of Object-Oriented Concepts is a key employability factor in the IT industry, aligning with the goal of making students employable.

CO2: WithPO7: Understanding the differences between Java and CPP is relevant to the employability of students, as it enables them to adapt to the technology demands of the IT industry.

CO3: WithPO7 : Reading input from different sources is a fundamental skill, contributing to employability by enhancing adaptability to diverse data sources in the IT industry.

CO4: WithPO7: The ability to define classes with various access modifiers and create objects is a practical skill highly valued in the IT industry, enhancing employability.

CO5: WithPO7 : Writing real-world problems using Java reflects practical application skills, making students more employable in the IT industry where problem-solving is crucial.

CO6: With PO7 : Understanding inheritance, early and late binding, exception handling, and generic programming contributes to students' employability by equipping them with advanced programming skills.

CO7: With PO7: Actively demonstrating the use of OOPs concepts through programs showcases practical proficiency, making students more employable in the IT industry

Class: S.Y.B.Sc.(Computer Science) (Semester- IV)

Subject: Computer Science **Paper Code:** CSCO2404

Title of Paper: Lab Course – IIon Soft. Engineering using Mini Project

Paper:IV(Grade) (Lab Course-II)

Credit: Grade (3 Hour Practical/Week/batch) **No. of Practical:** 12

Prerequisites:

- Basic knowledge of DBMS& RDBMS.
- Knowledge of programming languages and scripting.

OBJECTIVES:

1. To understand the process of designing and implementing Software.
2. To understand the Software Engineering concept for building different application.

Learning Outcome:

CO1. Identify problem definition and problem scope of real-life case studies.

CO2. Master over the various feasibility studies like technical feasibility, economical feasibility and operational feasibility.

CO3. Collect and Analyze data requirements and functional requirements.

CO4. Implementation of database normalization.

CO5. Implementation of various validation techniques with the help of JavaScript and Bootstrap.

CO6. Master over the front-end screen designing with the help of HTML5, CSS, JavaScript and Bootstrap technology.

CO7. Build various real-life websites using HTML5, CSS, JavaScript and Bootstrap technology.

Sr. No.	Assignment Name
1.	Problem definition, Scope
2.	Feasibility Study
3.	Gathering Data Requirements and Functional Requirement
4.	ER Diagrams
5.	Designing the Normalization Database
6.	Designing the Queries related to Functional requirements
7.	User interface analysis
8.	Screen designed (by using HTML5)
9	Build a Software engineering project through all the above conceptual Ideas.

Mapping of this course with Programme Outcomes

Course Outcomes	Programme Outcomes (POs)						
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**
CO1	3	3	3	3	3	3	3
CO2	2	3	3	3	3	3	3
CO3	3	3	3	2	3	3	3
CO4	3	3	3	1	2	2	2
CO5	3	3	3	2	3	3	3
CO6	3	3	3	2	2	3	3
CO7	3	3	3	2	3	3	3

Weight: 1 - Partially related 2 - Moderately Related 3 - Strongly related

Justification of Mapping of PO1 with All CO'S

CO1: PO1: Weightage 3 (Strongly Relates) Identifying problem definitions and scopes in real-life case studies requires the application of mathematics, statistics, and computer fundamentals, aligning strongly with the proficiency objective.

CO2: PO1: Weightage 2 (Moderately Related) While feasibility studies involve some application of computer knowledge, the direct link to mathematics, statistics, and computer fundamentals may be moderate.

CO3: PO1: Weightage 3 (Strongly Relates) Collecting and analyzing data requirements involve the application of statistics and computer fundamentals, strongly relating to the proficiency objective.

CO4: PO1: Weightage 3 (Strongly Relates) Database normalization involves applying mathematical and computational concepts, forming a strong relationship with the proficiency objective.

CO5: PO1: Weightage 3 (Strongly Relates) Implementing validation techniques with JavaScript and Bootstrap requires computer knowledge, aligning strongly with the proficiency objective.

CO6: PO1: Weightage 3 (Strongly Relates) Front-end screen designing involves applying computer fundamentals and technologies, strongly relating to the proficiency objective.

CO7:PO1: Weightage 3 (Strongly Relates)Building websites with HTML5, CSS, JavaScript, and Bootstrap technology directly applies computer knowledge, forming a strong relationship with the proficiency objective.

Justification of Mapping of PO2 with All CO'S

CO1:PO2: Weightage 3 (Strongly Relates) Designing solutions for IT applications involves understanding problem definitions and scopes, aligning strongly with the proficiency objective.

CO2:PO2: Weightage 3 (Strongly Relates) Mastering feasibility studies is essential in the design and development of IT solutions, forming a strong relationship with the proficiency objective.

CO3:PO2: Weightage 3 (Strongly Relates) Designing solutions requires a thorough understanding of data and functional requirements, strongly relating to the proficiency objective.

CO4:PO2: Weightage 3 (Strongly Relates) Implementing database normalization is a key aspect of developing IT solutions, aligning strongly with the proficiency objective.

CO5:PO2: Weightage 3 (Strongly Relates) Implementing validation techniques with JavaScript and Bootstrap is directly related to developing IT solutions using the latest technologies, forming a strong relationship.

CO6:PO2 : Weightage 3 (Strongly Relates) Mastering front-end screen designing involves using the latest technologies, strongly relating to the proficiency objective.

CO7:PO2: Weightage 3 (Strongly Relates) Building websites with the specified technologies directly applies to developing solutions for IT applications using the latest technologies, forming a strong relationship.

Justification of Mapping of PO3 with All CO'S

CO1:PO3: Weightage 3 (Strongly Relates) Creating, selecting, and applying modern tools to complex IT applications involves first identifying the problem definition and scope, aligning strongly with the proficiency objective.

CO2:PO3: Weightage 3 (Strongly Relates) The use of modern tools often requires a mastery of feasibility studies, forming a strong relationship with the proficiency objective.

CO3:PO3: Weightage 3 (Strongly Relates) Modern tools are frequently employed in the collection and analysis of data and functional requirements, strongly relating to the proficiency objective.

CO4:PO3: Weightage 3 (Strongly Relates) The use of modern tools in database normalization directly aligns with the proficiency objective.

CO5:PO3: Weightage 3 (Strongly Relates) Applying validation techniques with JavaScript and Bootstrap, as modern tools, strongly relates to the proficiency objective.

CO6:PO3: Weightage 3 (Strongly Relates) Mastering front-end screen designing with modern technologies is directly in line with the proficiency objective.

CO7:PO3: Weightage 3 (Strongly Relates) Building websites using modern technologies is an application of modern tools, forming a strong relationship with the proficiency objective.

Justification of Mapping of PO4 with All CO'S

CO1:PO4: Weightage 3 (Strongly Relates) Understanding the impact of IT analyst solutions in societal and environmental contexts requires first identifying the problem definition and scope, aligning strongly with the proficiency objective.

CO2:PO4: Weightage 3 (Strongly Relates) Assessing the feasibility of IT solutions includes considering their impact on society and the environment, forming a strong relationship with the proficiency objective.

CO3:PO4: Weightage 2 (Moderately Related) While the collection and analysis of requirements may not directly address societal and environmental impact, there may be some indirect connections.

CO4:PO4: Weightage 1 (Partially Related) The direct implementation of database normalization may not inherently relate to societal and environmental impact.

CO5:PO4: Weightage 2 (Moderately Related) The application of validation techniques may have some indirect implications for societal and environmental considerations.

CO6:PO4: Weightage 2 (Moderately Related) Front-end screen designing may indirectly influence the user experience and environmental considerations, forming a moderate relationship.

CO7:PO4: Weightage 2 (Moderately Related)

Building websites may indirectly impact societal and environmental aspects, particularly if the websites serve a specific purpose related to sustainability or societal well-being.

Justification of Mapping of PO1 with All CO'S

CO1:PO5: Weightage 3 (Strongly Relates) Applying ethical principles and professional responsibilities often starts with identifying and defining problems in real-life case studies, forming a strong relationship with the proficiency objective.

CO2: PO5: Weightage 3 (Strongly Relates) Committing to professional ethics and responsibilities is integral to mastering feasibility studies, aligning strongly with the proficiency objective.

CO3:PO5: Weightage 3 (Strongly Relates) Ethical considerations are crucial in the collection and analysis of data requirements and functional requirements, forming a strong relationship with the proficiency objective.

CO4:PO5: Weightage 2 (Moderately Related) While ethical principles are important in all aspects of IT, the direct link to database normalization may be more moderate.

CO5:PO5: Weightage 3 (Strongly Relates) Implementing validation techniques involves ethical considerations in ensuring data accuracy and security, aligning strongly with the proficiency objective.

CO6:PO5: Weightage 2 (Moderately Related) Ethical considerations may be indirectly involved in front-end design, making the relationship moderate.

CO7: PO5: Weightage 3 (Strongly Relates) Building websites with ethical considerations aligns strongly with the proficiency objective.

Justification of Mapping of PO6 with All CO'S

CO1: PO6: Weightage 3 (Strongly Relates) Functioning effectively as an individual and in a team requires the ability to identify and define problems in real-life case studies, forming a strong relationship with the proficiency objective.

CO2: PO6: Weightage 3 (Strongly Relates) Working in a team and mastering feasibility studies go hand in hand, aligning strongly with the proficiency objective.

CO3: PO6: Weightage 3 (Strongly Relates) Effective teamwork involves the collaborative collection and analysis of data and functional requirements, forming a strong relationship with the proficiency objective.

CO4: PO6: Weightage 2 (Moderately Related) While teamwork is crucial in IT projects, the direct link to database normalization may be more moderate.

CO5:PO6: Weightage 3 (Strongly Relates) Teamwork is essential in implementing validation techniques, aligning strongly with the proficiency objective.

CO6:PO6: Weightage 3 (Strongly Relates) Front-end screen designing often involves collaborative work in a team, strongly relating to the proficiency objective.

CO7: PO6: Weightage 3 (Strongly Relates)
Building websites requires effective teamwork, aligning strongly with the proficiency objective

----

Justification of Mapping of PO7 with All CO'S

CO1: PO7: Weightage 3 (Strongly Relates) Identifying opportunities and pursuing them to create value and wealth often starts with identifying and defining problems in real-life case studies, forming a strong relationship with the proficiency objective.

CO2:PO7: Weightage 3 (Strongly Relates) Developing innovation, employability, and entrepreneurial skills is closely tied to mastering feasibility studies, aligning strongly with the proficiency objective.

CO3:PO7: Weightage 3 (Strongly Relates) Innovation and entrepreneurial skills often involve the collection and analysis of data and functional requirements, forming a strong relationship with the proficiency objective.

CO4:PO7: Weightage 2 (Moderately Related) While entrepreneurship may involve IT projects, the direct link to database normalization may be more moderate.

CO5:PO7: Weightage 3 (Strongly Relates) Implementing validation techniques is relevant to creating value and wealth, strongly relating to the proficiency objective.

CO6:PO7: Weightage 3 (Strongly Relates) Front-end screen designing with modern technologies is crucial for innovation and entrepreneurial skills, forming a strong relationship with the proficiency objective.

CO7:PO7: Weightage 3 (Strongly Relates) Building real-life websites aligns with the goal of creating value and wealth, strongly relating to the proficiency objective.