

SEMESTER S3

OBJECT ORIENTED PROGRAMMING

(Common to CS/CA/CD/AM/CB/CN/CU/CG)

| | | | |
|----------------------------------|----------|-------------|----------------|
| Course Code | PBCST304 | CIE Marks | 60 |
| Teaching Hours/Week (L:T:P:R) | 3:0:0:1 | ESE Marks | 40 |
| Credits | 4 | Exam Hours | 2 Hrs. 30 Min. |
| Prerequisites (if any) | None | Course Type | Theory |

Course Objectives:

1. To teach the core object-oriented principles such as abstraction, encapsulation, inheritance, and polymorphism, robust error-handling using exception mechanisms to ensure program reliability.
2. To equip the learner to develop object oriented programs encompassing fundamental structures, environments, and the effective utilization of data types, arrays, strings, operators, and control statements for program flow in Java.
3. To enable the learner to design and develop event-driven graphical user interface (GUI) database applications using Swing and database connection components.

SYLLABUS

| Module No. | Syllabus Description | Contact Hours |
|------------|---|---------------|
| 1 | Introduction to Java: Structure of a simple java program; Java programming Environment and Runtime Environment (Command Line & IDE); Java compiler; Java Virtual Machine; Primitive Data types and Wrapper Types; Casting and Autoboxing; Arrays; Strings; Vector class; Operators - Arithmetic, Bitwise, Relational, Boolean Logical, Assignment, Conditional (Ternary); Operator Precedence; Control Statements - Selection Statements, Iteration Statements and Jump Statements; Functions; Command Line Arguments; Variable Length Arguments; Classes; Abstract Classes; Interfaces. [<i>Use proper naming conventions</i>] OOP Concepts :- Data abstraction, encapsulation, inheritance, polymorphism, Procedural and | 10 |

| | | |
|---|--|----|
| | <p>object oriented programming paradigm; Microservices.</p> <p>Object Oriented Programming in Java :- Declaring Objects; Object Reference; Introduction to Methods; Constructors; Access Modifiers; <i>this</i> keyword.</p> | |
| 2 | <p>Polymorphism :- Method Overloading, Using Objects as Parameters, Returning Objects, Recursion. Static Members, Final Variables, Inner Classes.</p> <p>Inheritance - Super Class, Sub Class, Types of Inheritance, The <i>super</i> keyword, protected Members, Calling Order of Constructors. Method Overriding, Dynamic Method Dispatch, Using <i>final</i> with Inheritance.</p> | 8 |
| 3 | <p>Packages and Interfaces – Packages - Defining a Package, CLASSPATH, Access Protection, Importing Packages.</p> <p>Interfaces - Interfaces v/s Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interface(s).</p> <p>Exception Handling - Checked Exceptions, Unchecked Exceptions, <i>try</i> Block and <i>catch</i> Clause, Multiple catch Clauses, Nested <i>try</i> Statements, <i>throw</i>, <i>throws</i> and <i>finally</i>, Java Built-in Exceptions, Custom Exceptions.</p> <p>Introduction to design patterns in Java : Singleton and Adaptor.</p> | 9 |
| 4 | <p>SOLID Principles in Java (https://www.javatpoint.com/solid-principles-java)</p> <p>Swings fundamentals – Overview of AWT, Swing v/s AWT, Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Swing Packages, Event Handling in Swings, Swing Layout Managers, Exploring Swings–JFrame, JLabel, The Swing Buttons, JTextField.</p> <p>Event handling – Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Event Model.</p> <p>Developing Database Applications using JDBC – JDBC overview, Types,</p> | 10 |

| | | |
|--|--|--|
| | Steps, Common JDBC Components, Connection Establishment, SQL Fundamentals [<i>For projects only</i>] - Creating and Executing basic SQL Queries, Working with Result Set, Performing CRUD Operations with JDBC. | |
|--|--|--|

Suggestion on Project Topics

Student should Identify a topic to be implemented as project having the following nature

- i. It must accept a considerable amount of information from the user for processing.*
- ii. It must have a considerable amount of data to be stored permanently within the computer - as plain files / using databases..*
- iii. It must process the user provided data and the stored data to generate some output to be displayed to the user.*

Examples : -

1. Design and implement the Circulation function in a Library Management System using Object-Oriented Programming (OOP) principles in Java and limited use of SQL. The system should manage the operations of a library, such as book & user management, borrowing and returning books.

Requirements

I. Class Design

- Book: Attributes like title, author, ISBN, genre, and status (available/borrowed).
- User: Attributes like user ID, name, contact information, and a list of borrowed books.
- Library: Attributes like a list of books and a list of users.
- Librarian: Inherits from User, with additional functionalities like adding/removing books and managing users.
- BorrowTransaction: Attributes like transaction ID, book, user, borrow date, and return date

II. Functionalities

a. Book Management:

- Add, remove, and update book details.
- Search books by title, author, ISBN, and genre.

b. User Management:

- Register new users.
- Search users by user ID and name.

c. Borrowing and Returning:

- Borrow a book: Check if the book is available and if the user can borrow more books.
- Return a book: Update the book's status and remove it from the user's borrowed list.

III. Deliverables

1. Design Document: Describe the classes, their attributes, methods and relationships.
 2. Source Code: Well-documented Java code implementing the described functionalities.
 3. User Manual: Instructions on how to set up, run and use the system.
 4. Test Cases: A suite of test cases demonstrating the functionality of the system.
2. Design and implement an Online Payment Processing System using Object-Oriented Programming(OOP) principles in Java, with a focus on dynamic polymorphism. The system should support different types of payment methods and demonstrate polymorphism in processing payments.

Requirements

a. Class Design

- Payment: An abstract base class with common attributes and an abstract method for processing payments.
- CreditCardPayment: Inherits from Payment, with specific implementation for processing credit card payments.
- PayPalPayment: Inherits from Payment, with specific implementation for processing PayPal payments.
- BankTransferPayment: Inherits from Payment, with specific implementation for processing bank transfer payments.
- PaymentProcessor: A class to manage and process different types of payments.

b. Functionalities

- Add Payment Method: Add new payment methods (CreditCardPayment, PayPalPayment, BankTransferPayment) to the system.
- Process Payment: Demonstrate dynamic polymorphism by processing payments using different methods.

c. Deliverables

- Design Document: Describe the classes, their attributes, methods and relationships.
- Source Code: Well-documented Java code implementing the described functionalities.

- User Manual: Instructions on how to set up, run and use the system.
- Test Cases: A suite of test cases demonstrating the functionality of the system.

Course Assessment Method
(CIE: 60 marks, ESE: 40 marks)

Continuous Internal Evaluation Marks (CIE):

| Attendance | Project | Internal Ex-1 | Internal Ex-2 | Total |
|------------|---------|---------------|---------------|-------|
| 5 | 30 | 12.5 | 12.5 | 60 |

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

| Part A | Part B | Total |
|--|---|-------|
| <ul style="list-style-type: none"> • 2 Questions from each module. • Total of 8 Questions, each carrying 2 marks (8x2 =16 marks) | <ul style="list-style-type: none"> • 2 questions will be given from each module, out of which 1 question should be answered. • Each question can have a maximum of 2 subdivisions. E • Each question carries 6 marks. (4x6 = 24 marks) | 40 |

Course Outcomes (COs)

At the end of the course students should be able to:

| Course Outcome | | Bloom's Knowledge Level (KL) |
|----------------|---|------------------------------|
| CO1 | Explain the process of writing, compiling, and executing basic Java programs, including their structure and components, to demonstrate proficiency. | K2 |
| CO2 | Utilize object-oriented programming principles in the design and implementation of Java applications. | K3 |
| CO3 | Develop and manage Java packages and interfaces, enhancing code modularity and reusability. | K3 |
| CO4 | Implement error handling using Java's exception mechanisms and leverage interfaces for modular applications. | K3 |
| CO5 | Develop event-driven Java GUI applications with database connectivity using Swing and JDBC. | K3 |

Note: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create

CO-PO Mapping Table:

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| CO1 | 3 | 3 | | | | | | | | | | 3 |
| CO2 | 3 | 3 | 3 | | | | | | | | | 3 |
| CO3 | 3 | 3 | 3 | | 3 | | | | | | | 3 |
| CO4 | 3 | 3 | 3 | | 3 | | | | | | | 3 |
| CO5 | 3 | 3 | 3 | | 3 | | | | | | | 3 |

Note: 1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation

| Text Books | | | | |
|------------|---|--|-----------------------|------------------|
| Sl. No | Title of the Book | Name of the Author/s | Name of the Publisher | Edition and Year |
| 1 | Java: The Complete Reference | Herbert Schildt | Tata McGraw Hill | 13/e, 2024 |
| 2 | Introduction to Java Programming, Comprehensive Version | Y Daniel Liang | Pearson | 10/e, 2014 |
| 3 | Head First Design Patterns | Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra | O'Reilly Media | 1/e, 2004 |

| Reference Books | | | | |
|-----------------|---|---------------------------|-----------------------|------------------|
| Sl. No | Title of the Book | Name of the Author/s | Name of the Publisher | Edition and Year |
| 1 | Head First Java: A Brain Friendly Guide | Kathy Sierra & Bert Bates | O'Reilly | 3/e, 2022 |
| 2 | JAVA™ for Programmers | Paul Deitel | PHI | 11/e, 2018 |
| 3 | Clean Code : A Handbook of Agile Software Craftsmanship | Robert C. Martin | Prentice Hall | 1/e, 2008 |
| 4 | Programming with Java | E Balagurusamy | McGraw Hill Education | 6/e, 2019 |
| 5 | Java For Dummies | Barry A. Burd | Wiley | 8/e.2022 |
| 6 | Effective Java | Joshua Bloch | Pearson | 3/e, 2018 |

| Video Links (NPTEL, SWAYAM...) | |
|--------------------------------|--|
| Module No. | Link ID |
| 1 | https://nptel.ac.in/courses/106105191 (Lecture no: 9, 10, 1, 2, 3, 4) |
| 2 | https://nptel.ac.in/courses/106105191 (Lecture no: 1, 7, 8, 11, 12, 13, 14, 15, 16) |
| 3 | https://nptel.ac.in/courses/106105191 (Lecture no: 17, 18, 19, 20, 21, 22, 23, 24, 25, 26) |
| 4 | https://nptel.ac.in/courses/106105191 (Lecture no: 43, 44, 45, 46, 47, 50, 51, 52, 53, 54, 55) |

PBL Course Elements

| L: Lecture (3 Hrs.) | R: Project (1 Hr.), 2 Faculty Members | | |
|---|---------------------------------------|--|--|
| | Tutorial | Practical | Presentation |
| Lecture delivery | Project identification | Simulation/ Laboratory Work/ Workshops | Presentation (Progress and Final Presentations) |
| Group discussion | Project Analysis | Data Collection | Evaluation |
| Question answer Sessions/ Brainstorming Sessions | Analytical thinking and self-learning | Testing | Project Milestone Reviews, Feedback, Project reformation (If required) |
| Guest Speakers (Industry Experts) | Case Study/ Field Survey Report | Prototyping | Poster Presentation/ Video Presentation: Students present their results in a 2 to 5 minutes video |

Assessment and Evaluation for Project Activity

| Sl. No | Evaluation for | Allotted Marks |
|--------------|---|----------------|
| 1 | Project Planning and Proposal | 5 |
| 2 | Contribution in Progress Presentations and Question Answer Sessions | 4 |
| 3 | Involvement in the project work and Team Work | 3 |
| 4 | Execution and Implementation | 10 |
| 5 | Final Presentations | 5 |
| 6 | Project Quality, Innovation and Creativity | 3 |
| Total | | 30 |

1. Project Planning and Proposal (5 Marks)

- Clarity and feasibility of the project plan
- Research and background understanding
- Defined objectives and methodology

2. Contribution in Progress Presentation and Question Answer Sessions (4 Marks)

- Individual contribution to the presentation
- Effectiveness in answering questions and handling feedback

3. Involvement in the Project Work and Team Work (3 Marks)

- Active participation and individual contribution
- Teamwork and collaboration

4. Execution and Implementation (10 Marks)

- Adherence to the project timeline and milestones
- Application of theoretical knowledge and problem-solving
- Final Result

5. Final Presentation (5 Marks)

- Quality and clarity of the overall presentation
- Individual contribution to the presentation
- Effectiveness in answering questions

6. Project Quality, Innovation, and Creativity (3 Marks)

- Overall quality and technical excellence of the project
- Innovation and originality in the project
- Creativity in solutions and approaches