

CST 205	OBJECT ORIENTED PROGRAMMING USING JAVA	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	3	1	0	4	2019

Preamble: The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Inheritance, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course helps the learners to develop Desktop GUI Applications, Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Prerequisite: Topics covered under the course PROGRAMMING IN C (EST 102)

Course Outcomes: After the completion of the course the student will be able to

CO1	Write Java programs using the object oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism (Cognitive Knowledge Level: Apply)
CO2	Utilise datatypes, operators, control statements, built in packages & interfaces, Input/ Output Streams and Files in Java to develop programs (Cognitive Knowledge Level: Apply)
CO3	Illustrate how robust programs can be written in Java using exception handling mechanism (Cognitive Knowledge Level: Understand)
CO4	Write application programs in Java using multithreading and database connectivity (Cognitive Knowledge Level: Apply)
CO5	Write Graphical User Interface based application programs by utilising event handling features and Swing in Java (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

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

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test1 (Marks %)	Test2 (Marks %)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

SYLLABUS

Object Oriented Programming Using Java

Module 1

Introduction:

Approaches to Software Design - Functional Oriented Design, Object Oriented Design, Case Study of Automated Fire Alarm System.

Object Modeling Using Unified Modeling Language (UML) – Basic Object Oriented concepts, UML diagrams, Use case model, Class diagram, Interaction diagram, Activity diagram, State chart diagram.

Introduction to Java - Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.

Module 2

Core Java Fundamentals:

Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.

Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.

Control Statements - Selection Statements, Iteration Statements and Jump Statements.

Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, *this* Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Final Variables, Inner Classes, Command Line Arguments, Variable Length Arguments.

Inheritance - Super Class, Sub Class, The Keyword *super*, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, using *final* with Inheritance.

Module 3

More features of Java:

Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces.

Exception Handling - Checked Exceptions, Unchecked Exceptions, *try* Block and *catch* Clause, Multiple *catch* Clauses, Nested *try* Statements, *throw*, *throws* and *finally*.

Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Object Streams and Serialization, Working with Files.

Module 4

Advanced features of Java:

Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, using valueOf(), Comparison of StringBuffer and String.

Collections framework - Collections overview, Collections Interfaces- Collection Interface, List Interface.

Collections Class – ArrayList class. Accessing a Collection via an Iterator.

Event handling - Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model.

Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Synchronization, Suspending, Resuming and Stopping Threads.

Module 5

Graphical User Interface and Database support of Java:

Swings fundamentals - 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.

Java DataBase Connectivity (JDBC) - JDBC overview, Creating and Executing Queries – create table, delete, insert, select.

Text Books:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Rajib Mall, Fundamentals of Software Engineering, 4th edition, PHI, 2014.
3. Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11th Edition, Pearson, 2018.

Reference Books:

1. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
2. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
3. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
4. Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.
6. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

Sample Course Level Assessment Questions

Course Outcome1(CO1): For the following passage develop UML diagrams and then implement it as a Java program in accordance with your UML design.

Passage: College Office collects semester fee and college bus fee for each student. A clerk at the college office collects the fees from each student. The bus fee is calculated depending on the distance of the corresponding bus stop from the college. The semester fee varies depending upon the semester as well as branch of each student. Students are supposed to pay the fees in full. Economically backward students are eligible for 50% discount in semester fee. The consolidated fees receipt is issued to each student by the clerk, which contains the student name, admission number, semester and branch of student along with details of fees collected. Students can log in and view the details of fees remitted and dues if any. The system allows students and clerk level login to the system. Clerk is able to view reports of each class showing status of fees payment of each student.

Course Outcome 2(CO2): Write a Java program to evaluate a post fix expression containing two operands and a single operator using stack. Stack should be implemented as a separate entity so as to reflect OOP concepts.

Course Outcome 3(CO3): Write a program to demonstrate the start, run, sleep and join methods in Thread class.

Course Outcome 4(CO4): Write a GUI based program with separate buttons to add, delete and display student details i.e. name, student ID, current semester and branch of study based on student ID.

Course Outcome 5(CO5): Using Swing create a JFrame with a JLabel and two JButtons. Set the texts of JButtons as “Yes” and “No” respectively. Set the JLabel’s text to the text of the button currently being pressed. Initially the JLabel’s text is blank.

Model Question Paper

QP CODE:

PAGES:3

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRD SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 205

Course Name: Object Oriented Programming using Java

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Briefly explain the portable, secure and robust features of Java.
2. Describe the concepts of object and class with a suitable Java program.
3. Explain the concept of method overriding with an example.
4. What is the use of the keyword *final* in Java?
5. Explain the concept of streams.
6. Explain any two applications of Serialization.
7. Distinguish the usage of “==” and *equals()* method when comparing String type?
8. What are Collections in Java? Explain any one Collection interface in Java.
9. Explain any two properties of Swing components in Java.
10. Explain JLabel component. With suitable examples explain any two of its constructors.

Part B

Answer any one question completely from each module

11.

- (a) Describe in detail any three Object Oriented Programming principles. Illustrate with suitable examples.

(9)

- (b) What is Java Runtime Environment? What is the role of Java Virtual Machine in it? (5)

OR

12.

- (a) Compare and contrast Java standard edition and Java enterprise edition. (5)
- (b) Why is Java considered to be platform independent? What is the role of Bytecode in making Java platform independent? (9)

13.

- (a) Explain in detail the primitive data types in Java. (8)
- (b) Explain automatic type conversion in Java with an example. What are the two conditions required for it? (6)

OR

14.

- (a) Using a suitable Java program explain the difference between *private* and *public* members in the context of inheritance. (8)
- (b) Is it possible to use the keyword *super* within a static method? Give justification for your answer. (6)

15.

- (a) Explain in detail about byte streams and character streams with suitable code samples. (6)
- (b) Describe in detail about exception handling, *try* block and *catch* clause with the help of a suitable Java program. (8)

OR

16.

- (a) Explain object streams in Java. Explain the role of Serializable interface with a suitable code sample. (8)
- (b) Explain *throw*, *throws* and *finally* constructs with the help of a Java program. (6)

17.

(a) Describe in detail the creation of a thread using the Runnable interface and the Thread class with suitable examples. (10)

(b) Explain List Interface. Mention any two exceptions thrown by its methods. (4)

OR

18.

(a) Explain in detail the Delegation Event model for event handling in Java. (7)

(b) Write a simple program by extending appropriate class to demonstrate the working of threads in java. (7)

19.

(a) Write a Java program to demonstrate the use of JLabel and JButton by adding them to JFrame. (7)

(b) Explain step-by-step procedure of using Java DataBase Connectivity in Java programs. (7)

OR

20.

(a) Explain the class hierarchy of Java Swing components. (7)

(b) Write a Java Program to create a student table and to add student details to it using JDBC. (7)