| CST 281 | OBJECT ORIENTED PROGRAMMING | CATEGORY | L | T | P | CREDIT | YEAR OF INTRODUCTION |
|---|---|---|---|---|---|---|---|
| | | MINOR | 3 | 1 | 0 | 4 | 2019 |

**Preamble**: This is the programming course for awarding B.Tech. Minor in Computer Science and Engineering with specialization in ***Software Engineering*****.** 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 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 (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 outcome**

| | 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 UML – Basic Object Oriented concepts, UML (Unified Modeling Language) 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.

### Module 3

**More features of Java:**

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.

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***.

## Module 4

**Advanced features of Java:**

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

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 Class – ArrayList. Accessing Collections via an Iterator.



## Module 5

**GUI Programming, Event Handling and Multithreaded Programming:**

Swing fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Exploring Swing - JFrame, JLabel, JButton, JTextField.

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, Suspending, Resuming and Stopping Threads.

**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 prepare the rank list of students based one their performance in the first Semester B.Tech. Degree examination at APJ Abdul Kalam Technological University. The output should be stored in a file.

**Course Outcome 3 (CO3):** Write a program to demonstrate how event handling and exception handling are supported in Java..

**Course Outcome 4 (CO4):** Write a program to demonstrate the start, run, sleep and join methods in Thread class..

**Model Question Paper**

QP CODE: PAGES:3

Reg No:_____

Name:_____

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

**THIRD SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH & YEAR**

**Course Code: CST 281**

**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 why Java is considered to be secure and portable.
2. Describe the concept of association among classes with an example.
3. Explain the different arithmetic operators in Java.
4. Explain the use for command line arguments with a suitable Java program
5. Explain the use of CLASSPATH with an example.
6. What are the different types of exceptions?
7. Explain file handling features available in Java.
8. Write a simple program to read and print an integer value in Java.
9. Explain the concept of *main thread* in multi-threading.
10. Explain any two Event classes in Java.

**Part B**

**Answer any one question completely from each module**

11.

(a) Describe in detail polymorphism, abstraction and inheritance with suitable examples. (9)

(b) What is Java Virtual Machine? (5)

**OR**

12.

(a) Compare and contrast Functional Oriented and Object Oriented approach by considering a simple bus ticket reservation system. (5)

(b) What is a class diagram? Explain with an example. (9)

13.

(a) Explain primitive data types in Java. How are they different from other data types? (8)

(b) Explain variables and arrays in Java. (6)

**OR**

14.s

(a) Using a suitable Java program explain the concept of methods and constructors. (8)

(b) Explain the keyword ***super*** and its usage in Java. (6)

15.

(a) Using a table, explain the effect of access specifiers in inheritance. (6)

(b) Describe in detail about exception handling using **try** block and **catch** clause in Java with the help of a suitable Java program. (8)

**OR**

16.

(a) What is an interface in Java? Explain with a suitable example. (8)

(b) Explain ***throw***, ***throws*** and ***finally*** constructs with the help of a Java program. (6)

17.

(a) Explain ***ArrayList*** collections framework. Also explain the use of iterator in accessing collections.

(8)

(b) Bring out difference between "==" and ***equals()*** method with the help of a sample program

(6)

**OR**

18.

(a) Compare Byte Streams and Character Streams. Write a program to demonstrate the usage of the ***PrintWriter*** class.

(8)

(b) Explain any three String constructors with the help of sample code for each.

(6)

19.

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

(7)

(b) Describe in detail the creation of a thread using the Runnable interface.

(7)

**OR**

20.

(a) What are the differences between a process and a thread?

(4)

(b) Write a Graphical User Interface (GUI) based Java program to implement a simple calculator supporting the operations addition, subtraction, multiplication and division. Use Swing controls to implement GUI. There may be three text boxes, the first two for operands and the last for result. Add four buttons for the above operations. Write neat comments in your program to show how you handle events.

(10)

| Teaching Plan | | |
|---|---|---|
| **Module 1 (Introduction)** | | **(8 hours)** |
| 1.1 | Approaches to Software Design- Functional Oriented Design, Object-Oriented Design, Case Study of Automated Fire Alarm System. | 1 hour |
| 1.2 | Object Modeling Using UML – Basic object oriented concepts | 1 hour |
| 1.3 | Basic object oriented concepts | 1 hour |
| 1.4 | UML diagrams, Use case model | 1hour |
| 1.5 | Class diagram, Interaction diagram | 1hour |
| 1.6 | Activity diagram, State chart diagram | 1hour |
| 1.7 | Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. JVM, Java compiler, Bytecode | 1hour |
| 1.8 | Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues | 1hour |
| **Module 2 (Core Java Fundamentals)** | | **(12 hours)** |
| 2.1 | Primitive Data types - Integers, Floating Point Types, Characters, Boolean | 1 hour |
| 2.2 | Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class. | 1 hour |
| 2.3 | Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence. | 1 hour |
| 2.4 | Control Statements - Selection Statements, Iteration Statements and Jump Statements. | 1 hour |
| 2.5 | Object Oriented Programming in Java - Class Fundamentals, Declaring Objects | 1 hour |
| 2.6 | Object Reference, Introduction to Methods | 1 hour |
| 2.7 | Constructors, ***this*** Keyword | 1 hour |
| 2.8 | Method Overloading, Using Objects as Parameters | 1 hour |

| | | |
|---|---|---|
| 2.9 | Returning Objects, Recursion | 1 hour |
| 2.10 | Access Control, static Members | 1 hour |
| 2.11 | Final Variables, Inner Classes | 1 hour |
| 2.12 | Command-Line Arguments, Variable Length Arguments | 1 hour |
| **Module 3 (More features of Java)** | | **(8 hours)** |
| 3.1 | Inheritance - Super class, Sub class, the keyword super, protected Members, | 1 hour |
| 3.2 | Calling Order of Constructors, Method Overriding, the Object class, | 1 hour |
| 3.3 | Abstract Classes and Methods, Using final with Inheritance | 1 hour |
| 3.4 | Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages | 1 hour |
| 3.5 | Interfaces | 1 hour |
| 3.6 | Exception Handling - Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause | 1 hour |
| 3.7 | Multiple ***catch*** Clauses, Nested ***try*** Statements | 1 hour |
| 3.8 | ***throw***, ***throws*** and ***finally*** | 1 hour |
| **Module 4 (Advanced features of Java)** | | **(8 hours)** |
| 4.1 | Input/Output - I/O Basics, Reading Console Input | 1hour |
| 4.2 | Writing Console Output, PrintWriter Class | 1hour |
| 4.3 | Object Streams and Serialization | 1hour |
| 4.4 | Serialization, Working with Files | 1hour |
| 4.5 | Working with Files | 1hour |
| 4.6 | Java Library - String Handling – String Constructors, String Length, Special String Operations | 1hour |
| 4.7 | Character Extraction, String Comparison, Searching Strings, Modifying Strings Using valueOf( ), Comparison of StringBuffer and String. | 1hour |
| 4.8 | Collections framework – Collections overview, Collections Class – ArrayList. Accessing Collections via an Iterator. | 1hour |

| Module 5 (GUI Programming, Event Handling and Multithreaded Programming) | | (9 hours) |
|---|---|---|
| 5.1 | Swings fundamentals, Swing Key Features | |
| 5.2 | MVC, Swing Controls, Components and Containers | |
| 5.3 | Exploring Swing –JFrame, JLabel, JButton, JTextField. | |
| 5.4 | Event handling - Event Handling Mechanisms, Delegation Event Model | 1hour |
| 5.5 | Delegation Event Model, Event Classes | 1hour |
| 5.6 | Sources of Events, Event Listener Interfaces, Using the Delegation Model | 1hour |
| 5.7 | Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread | 1hour |
| 5.8 | Creating Multiple Threads | 1hour |
| 5.9 | Suspending, Resuming and Stopping Threads. | 1hour |