C5-R3: OBJECT ORIENTED METHODOLOGY

NOTE:

> **1. Answer question 1 and any FOUR questions from 2 to 7.**
> **2. Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1.

a) Give difference between reference, object and instance of class. Can we calculate the number of instances created in any Java program?

b) What is dynamic polymorphism in Java?

c) What is the difference between 'throw' and 'throws'? How can it help in Exception handling?

d) What is Exception? What is the importance of it? Explain the chaining Exception.

e) What is a servlet? What can you achieve using servlets?

f) What do you understand by distributed object computing? What are the advantages of distributed object computing?

g) How does CORBA facilitate distributed object computing?

(7x4)

2.

a) Explain the concepts encapsulation, generalization and polymorphism. Discuss, how the above concepts support reusability.

b) In UML, interaction between objects may be modelled using sequence diagram or collaboration diagrams. Describe and differentiate both diagrams. The class diagram and these interaction diagrams should be mutually consistent. Explain.

c) What are difficulties in the implementation of distributed processes and object parallelism? Discuss.

(6+6+6)

3.

a) Explain the concept of exceptions in Java. Using a small example illustrate how are they useful?

b) Explain method overloading and its usefulness. Give some examples of method overloading.

c) Explain, how socket programming is supported in Java. What is the difference between a server socket and a client socket? Explain, how you can create a server socket.

(6+6+6)

4.

a) Over the time you have got huge amount of music and video CD collection. Some of your music CDs were recorded by just one band/artist and some have several songs/pieces of different band/artists. Some popular songs are remade several times. Some times different artists involved with these remaking.

You found it difficult frequently to search for a particular song made by your favourite band or single artist. You were also tired of searching for your much loved movies.

Finally you had decided to label the CDs with unique serial numbers and develop a computer program to manage your requirements.

Initially, you had identified that the following requirements were important.

i) To add information about CDs; In general, this includes title of the CD, price, date purchased etc. and for music CD, the title of the songs (and its version, if necessary) and the corresponding artist or bands detail and etc. For the movies, the details of main actor/actress, director and your own grading, etc.

ii) To search for a song on its title, artist or band.

iii) To search for a movie on its grading (e.g. A(very good), B(good), etc.)

Prepare a 'use case' diagram for all the above requirements. If necessary, your diagram should comprise appropriate 'include', 'extends' and 'generalization' relationships. Further, write a detailed use case description for the second use case only.

b) Define the substitutability concept in object oriented programming with a suitable example in Java code. Also discuss about encapsulation and information-hiding concepts in object-oriented technology with suitable examples in Java code.

(10+8)

5.

a) Can we declare an array of an object in JAVA? What is the advantage of it?

b) What is InetAddress? What is the importance of it in networking? Explain networking in JAVA with example?

c) What is JDBC? How can it handle database? How PreparedStatement is different from Statement Object?

(6+6+6)

6.

a) Relational Database Management Systems (RDBMS) are still widely used to handle persistent objects. Using suitable examples, explain how a generalization specialization relationship may be resolved in Relational Data Modeling.

b) What is the difference between method overloading and method overriding? Explain with a suitable example.

c) Using a few examples explain, how the following types of relations among classes can be implemented in Java:

* association,

* generalization, and

* aggregation (is-part of)

(6+6+6)

7.

a) Explain AWT class hierarchy. What is event-driven programming? Using suitable examples explain AWT supports event-driven programming, with special reference to event registration, listening and handling.

b) What is CORBA? What are the advantages of having a CORBA-based application as compared to a client-server Java solution using sockets?

c) What do you understand by multiple inheritance? Give an example of multiple inheritance.

(6+6+6)