No. of Printed Pages : 5

MCS-024

MCA (Revised) / BCA (Revised)

Term-End Examination

December, 2017

04950

MCS-024 : OBJECT ORIENTED TECHNOLOGIES AND JAVA PROGRAMMING

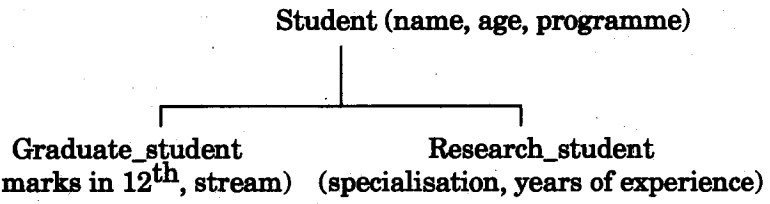
Time : 3 hours *Maximum Marks : 100*

(Weightage 75%)

Note : *Question no. 1 is* ***compulsory*** *and carries 40 marks. Attempt any* ***three*** *questions from the rest of the questions.*

---

1. (a) Consider the following class hierarchy :



A university has two types of students — graduate students and research students. The University maintains the record of name, age and programme of every student. For graduate

students, additional information like percentage of marks and stream, like science, commerce, etc. is recorded; whereas for research students, additionally, specialisation and years of working experience, if any, is recorded. Each class has a constructor. The constructor of subclasses makes a call to constructor of the superclass. Assume that every constructor has the same number of parameters as the number of instance variables. In addition, every subclass has a method that may update the instance variable values of that subclass. All the classes have a function display_student_info( ), the subclasses must override this method of the base class. Every student is either a graduate student or a research student.

Perform the following tasks for the description given above using Java :

(i) Create the three classes with proper instance variables and methods, with suitable inheritance. *5*

(ii) Create at least one parameterised constructor for each class. *3*

(iii) Implement the display_student_info( ) method in each class. *3*

(iv) Create an appropriate main method with at least two objects of each subclass. Can you create an object of the superclass ? Justify your answer. 3

(v) Write the code in main that uses objects and overriding to show polymorphism. 3

(b) What is multithreading ? What are the advantages of multithreading ? What is a main thread in the context of Java ? 5

(c) Write a Java program that accepts the input from the keyboard and writes it to a text file. 5

(d) Explain how an event is handled in Java with the help of an example program. 6

(e) Assume that a database named "Student" exists with attributes student_ID, student_name and programme. Write a Java program segment which will execute a SELECT query and display the resultant records. You need not write the connection command. 4

(f) What is Bytecode in Java ? What are the advantages of using bytecode in Java ? 3

2. (a) List the salient features of object oriented programming approach that distinguishes it from the procedural programming. *4*

(b) Explain different forms of inheritance with the help of diagrams. What are the advantages of using inheritance ? *8*

(c) Explain the following with the help of an example of each : *8*

(i) Dynamic Initialization

(ii) Operator Precedence

(iii) Switch Statement

(iv) Array Initialization

3. (a) What is Static Method in Java ? Explain with the help of an example. *4*

(b) Explain with the help of an example program, how objects can be passed as parameters in Java. *5*

(c) What is a Package in Java ? How are they related to CLASSPATH ? Explain with the help of an example program. *6*

(d) What is an Interface in Java ? How are they different from ABSTRACT classes ? *5*

4. (a) Explain with the help of an example program, how interthread communication is performed in Java using wait( ) and notify( ) and other methods. *6*

(b) How is Character class in Java different to String class ? Explain how you will compare two objects of String class. Also write a program that converts lowercase characters to uppercase characters of a string. *8*

(c) What is the purpose of the following Stream classes ? *6*

(i) PrintStream

(ii) RandomAccessFile

(iii) ByteArrayInputStream

(iv) FilterOutputStream

5. Explain the following with the help of a/an diagram/example/program, if needed : *20*

(a) Paint( ) Method of Applet

(b) HTML Applet Tag

(c) Button

(d) Checkbox Group

(e) Grid Layout (No need of writing program)

(f) Container

(g) Uses of RMI

(h) POST Method