Java Standard Edition

Duration: 90 Minutes **Marks: 50**

Section I [Multiple Choice Questions] : **1 Marks Each**

1. Which sentence defines the association concept in OO?

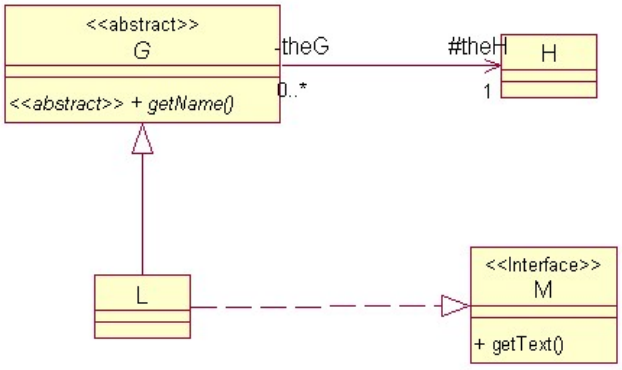
A. Association is a way to form new classes (instances of which are called objects) using classes that have already been defined, where data and behavior of existent classes are derived by the new class.

B. Association defines a relationship between classes of objects which allows one object instance to delegate responsibilities to another object on its behalf.

C. Association is behavior that varies depending on the class in which the behavior is invoked.

D. Association allows multiple functions taking different types to be defined with the same name.

2. Take a look at the UML diagram, please pick the correct options below.



A. class G extends L ……
class M implements L ……

B. class L implements G ……
class M extends L ……

C. class L extends M and implements G ……

D. class L extends G and implements M ……

E. None of the above

3. Which Order class properly represents the relationship "Order has a customer who is a Customer"?

A. class Order extends Customer {}

B. class Order implements Customer {}

C. class Order { private int Customer; }

D. class Order { private Customer customer; }

4. A blueprint for a software object is called a ___.

A. Method
B. Class
C. Object
D. All Of them listed above.

5. Hiding internal data from the outside world, and accessing it only through publicly exposed methods is known as data ___.

E. Encapsulation
F. Abstraction
G. Polymorphism
H. Binding

6. A software object's state is stored in ___.

I. Methods
J. Attributes
K. Class
L. None of the above

7. What is the correct ordering for the import, class and package declarations when found in a single file?

A. package, import, class

B. class, import, package

C. import, package, class

D. package, class, import

8. What is the parameter specification for the public static void main method?

A. String args []

B. String [] args

C. Strings args []

D. String args

9. Which of these are legal identifiers. Select the three correct answers.

A. number_1

B. number_a

C. $1234

D. -volatile

10. Which of the following are keywords in Java. Select the two correct answers.

A. friend

B. NULL

C. implement

D. synchronized

E. throws

11. Which of the following are true. Select the three correct answers.

A. A static method may be invoked before even a single instance of the class is constructed.

B. A static method cannot access non-static methods of the class.

C. Abstract modifier can appear before a class or a method but not before a variable.

D. final modifier can appear before a class or a variable but not before a method.

E. Synchronized modifier may appear before a method or a variable but not before a class.

12. Which of the following declares an abstract method in an abstract Java class?

A. public abstract method();

B. public abstract void method();

C. public void abstract Method();

D. public void method() {}

E. public abstract void method() {}

13. Which of the following statements regarding abstract methods are true?

A. An abstract class can have instances created using the constructor of the abstract class.
B. An abstract class can be extended.
C. A subclass of a non-abstract superclass can be abstract.
D. A subclass can override a concrete method in a superclass to declare it abstract.
E. An abstract class can be used as a data type.

14. How can you force garbage collection?

A. grabage collection cannot be forced.
B. Call System.gc().
C. Call Runtime.gc().
D. set the reference to null.

15. Consider the following line of code:

```
int x[]=new int[25];
```

After execution, which statement is not true?

A. x.length is 25.
B. x[0] is null.
C. x[24] is 0
D. x[24] is 0.0

16. After execution of the code below what are the values of x, a, and b?

```
1: int x,a=6,b=7;
2: x=a++ + b++;
```

A. x=15, a=7, b=8
B. x=15, a=6, b=7
C. x=13, a=7, b=8
D. x=13, a=6, b=7

17. What are the reasons to create an instance of the FileInputStream and FileOutputStream class?

A. To determine whether the file exist.

B. To obtain the properties of the file such as whether the file can be read, written, or is hidden.

C. To rename the file.

D. To delete the file.

E. To read/write data from/to a file

18. Which method(s) must a serializable class implement?

A. It must always implement both readObject and writeObject

B. It must implement either readObject or writeObject, or both, depending upon the desired behavior

C. No methods.

19. Which of the following statements loads the JDBC-ODBC driver?

A. Class.forName(oracle.jdbc.driver.OracleDriver)

B. Class.forName("oracle.jdbc.driver.OracleDriver ")

C. Class.loadClass(oracle.jdbc.driver.OracleDriver)

D. Class.loadClass("oracle.jdbc.driver.OracleDriver")

20. Name the Different Types of Drivers.

Section II [Predict Output] : **1 Marks Each**

1. Predict the output for the following snippet of Code.

```
// filename Test.java
class Test {
    void show()
        {
                System.out.println(" Test::Show() Called");
        }
}

public class Main {
    public static void main(String args[]) {
        Test t;
        t.show();
    }
}
```

2. Predict the output

```
public class Main {
    public static void main(String[] args) {
        Integer x = new Integer(400);
        Integer y = new Integer(400);

        if( x == y )
        {
                System.out.println("Same");
        }
        else
        {
                System.out.println(" Not Same ");
        }
    }
}
```

3. Predict the Ouput

```
class Test {

        static int count = 0;

        Test()
        {
                count++;
        }

    public static void main(String args[]) {
        Test t1 = new Test();
        Test t2 = new Test();

         System.out.println("Total " + count + " Objects Created ");
    }
}
```

4. How are Parameters Passed in Java ?

```
class Test
{
        public static void swap(int i, int j)
        {
                temp =i;
                i = j;
                j = temp;
                System.out.println(" I = " + i " , J = " + j );
        }
```

```
        public static void main(String args[])
        {
                Integer i = new Integer(10);
                Integer j = new Integer(20);

                swap(i , j );
        }
}
```

5. Copy Constructors in Java

```
class Complex
{
    private double re, im;

    public Complex(double re, double im)
    {
            this.re = re;
            this.im = im;
    }

    Complex(Complex c)
    {
        System.out.println(" Copy Constructor Called ");
        re = c.re;
        im = c.im;
    }

    @Override
    public String toString()
    {
        return "[ " + re + "  +  "  + im  + " i ] ";
    }
}

public class Main
{
```

```
        public static void main(String args[])
        {
            Complex c1 = new Complex(10,15);

            Complex c2 = new Complex(c1);

            Complex c3 = c2;

            System.out.println(c2);
        }
    }
```

6. Predict the Output of the following code:

```
import java.io.*;
public class Main
{
    public static void main(String args[])
    {
            FileReader file = new FileReader("C:\\Test\\a.txt");
            BufferReader fileInput = new BufferReader(file);

            for(int counter = 0; counter < 3;counter++)
            {
                    System.out.println(fileInput.readLine());
            }
            fileInput.close();
     }
}
```

7. Predict the Output

```
        class Base
        {
            Base()
            {
                    System.out.println(" Base Class Constructor Callled ");
            }
        }

        class  Derived extends Base
        {
            Derived()
            {
                    System.out.println(" Derived Class Constructor Callled ");
```

```
        }
}

public class Main
{
        public static void main(String args[])
        {
                Derived derivedRef = new Derived();
        }
}
```

8. What is the output?

```
class Base
{
        int x;

        Base(int _x)
        {
                x = _x;
        }
}

class Derived extends Base
{
        int y;

        Derived(int  _x,  int  _y)
        {
                super(_x);
                y = _y;
        }

        void display()
        {
                System.out.println(" X  = " + x  + " Y =  " + y);
        }
}

public class Main
{
        public static void main(String args[])
        {
                Derived d = new Derived(10,20);
                d.display();
        }
}
```

9. What is the output of this code?

```
abstract class Base
{
        public void fun()
        {
                System.out.println(" Base Fun() Called ");
        }
}
class Derived
{
        public void fun()
        {
                System.out.println(" Derived Fun() Called ");
        }
}
class Main
{
        public static void main(String args[])
        {
                Base b = new Derived();
                b.fun();
        }
}
```

10. What is the output of this program?

```
class Outer
{
        class Inner
        {
                public void display()
                {
                        System.out.println(" In a Nested Class Method ");
                }
        }
}

public class Main()
{
        public static void main(String args[])
        {
                Outer outer = new Outer();
                outer.Inner inner = new outer.Inner();
                inner.display();
        }
}
```

Section III [Descriptive Questions]: **4 Marks Each**

1. Write a Sample Application using JDBC API to perform Read Delete, Insert and Update Operations for **[Departments Table]**
2. What is an Inner Class ? Write a Sample Program to Illustrate Member Level Inner Class
3. Explain Different Types of Constructors in Java, with Sample to Demonstrate All the Constructors
4. Create a class to construct an ArrayList of Customer objects, and then provide a method to find and return a Customer object for a given ID. The Order class needs to be modified to contain an ArrayList of order items, requiring a method to add items into the ArrayList, and another method to remove the items.
5. Explain Java SE New Features **[Usage of Lambda Expression, Different Functional Interfaces and Modular Programming in Java]**