ASSESSMENT – 3 - OOPS Principles - Sagar - FISERV

* Required

1

Name: Batch : Email id: *

Enter your answer

2

Java doesn't support……. Inheritance [with classes] (1 Point)

- a) Multi level Inheritance
- b) Hierarchial Inheritance
- c) Multipath Inheritance
- d) Single Inheritance

3

Pick the Incorrect statement from the below options (1 Point)

- a) super keyword presents implicitly in superclass

b) super keyword refers both superclass and subclass memory

c) super keyword explicitly required to access whenever both superclass subclass names are same

d) super keyword will be considered as ObjectReference

4

What is the output of the below program?

```
class A{
        A(int x){
        System.out.println("x= "+x);
}
}
class B extends A{
        B(int y)
        {
                System.out.println("y= "+y);
}
public static void main(String args[]){
        B ob = new B(5);
}
 }
```

(1 Point)

a) x=5, y=5

b) x=5

◯ c) y=5

◯ d) compile error

5

Order of less restrictive to more restrictive of Access Modifiers (1 Point)

◯ a) public □ No Name □ protected □ private

◯ b) private □ No Name □ protected □ public

◯ c) public □ protected □ No Name □ private

◯ d) private □ protected □ No Name □ public

6

What is the output of the below program?

```
class Father {


        public void car() {

                System.out.println("Father's Car");

        }

}


class Son extends Father {
```

```
        public void car() {
                System.out.println("Son's Car");
        }
}

public class Sample {

        public static void main(String[] args) {

                Father john = new Son();
                john.car();
        }
}
```

(1 Point)

- a) Father's Car
- b) Son's Car
- c) There is an ambiguity, so no one Car
- d) Compiler Error

7

Pick the Incorrect statement w.r.t abstraction (1 Point)

- a) Abstract class can have combination of Abstract methods and Non Abstract methods

b) In Abstract class, abstract method must have abstract keyword, otherwise compile error occurs

c) In Interface, abstract method must have abstract keyword, otherwise compile error occurs

d) Abstract class can have all Non Abstract methods

8

How to create object of the inner class? (1 Point)

a) OuterClass.InnerClass innerObject = outerObject.new InnerClass();

b) OuterClass.InnerClass innerObject = new InnerClass();

c) InnerClass innerObject = outerObject.new InnerClass();

d) OuterClass.InnerClass = outerObject.new InnerClass();

9

What is the Output of the below program

```
public class Test {
    public int a=0;
    class innerClass
    {
        public int x=1;
        void innermethod(int x)
        {
            System.out.println("value of x = " + x);
```

```
        System.out.println("value of this.x =  "+ this.x);
        System.out.println("value of Test.this.x = " + Test.this.a);
    }
  }
  public static void main( String args[] )
  {
    Test t=new Test();
    Test.innerClass im=t.new innerClass();
    im.innermethod(55);
  }
} (1 Point)
```

- value of x = 55 value of this.x = 0 value of Test.this.x = 1
- value of x = 1 value of this.x = 0 value of Test.this.x = 55
- value of x = 55 value of this.x = 1 value of Test.this.x = 0
- value of x = 0 value of this.x = 55 value of Test.this.x = 1

10

Write the Output of the below program?

```
abstract class Car {
    public Car(String name) {
    super();
```

```
        System.out.print("2");
    }


    {
        System.out.print("3");
    }
}


public class BlueCar extends Car {
    {
        System.out.print("4");
    }

    public BlueCar() {
        super("blue");
        System.out.print("5");
    }

    public static void main(String[] gears) {
        new BlueCar();
    }
}
```
(1 Point)

Enter your answer

11

What is the Output of below program?

```
class Color {
    int red, green, blue;

    void Color() {
        red = 10;
        green = 10;
        blue = 10;
    }

    void printColor() {
        System.out.println("red: " + red + " green: " + green + " blue: " + blue);
    }
}

public class Test {
    public static void main(String[] args) {
        Color color = new Color();
```

```
        color.printColor();
    }
}
```
(1 Point)

Enter your answer

12

Coding Questions:

You are given an interface *AdvancedArithmetic* which contains a method signature *int divisor_sum(int n)*.

You need to write a class called MyCalculator which implements the interface.

***divisorSum* function just takes an integer as input and return the sum of all its divisors.**

For example divisors of 6 are 1, 2, 3 and 6, so *divisor_sum* should return 12.

(1 Point)

Enter your answer

13

Please Implement the code with below Inputs

Create below Classes

- **Person.java**

- Student.java

- Teacher.java

- Subject.java

- ClassRoom.java

- Test.java

Implement Hierarchial Inheritance

-Person.java is a Super Class

-Student.java and Teacher.java are Sub Classes

a) In "Person.java" file

-create below instance variables with private access modifier id, age, gender

id (Datatype is int)

age (Datatype is int)

gender (Datatype is char)

-Write a Parameterized Constructor and initialize all these '3' instance variables

-Write getter methods for these '3' instance variables

-write a instance method - without return type and without arguments

-In this instance method - print all '3' values

(1 Point)

Enter your answer

14

b) In "Student.java" file

-Create instance variables rollNumber and classRoom variables with private access modifier

rollNumber (Datatype is int)

classRoom (Datatype is ClassRoom)

-Write a Parameterized Constructor call Super Class Parameterized Constructor explicitly and also initialize these '2' instance variables

- Override the super class method in Sub Class (SuperClass and SubClass method names must be same)

- In this overriding method Print the "RollNumber"

- Also In this method Call the Super Class method using Super keyword, It should print super class instance variables data (1 Point)

Enter your answer

15

c) In "Teacher.java" file

-Create instance variables experienceYears and specialization variables with private access modifier

experienceYears (Datatype is int)

specialization (Datatype is Subject)

-Write a Parameterized Constructor call Super Class Parameterized Constructor explicitly and also initialize these '2' instance variables

- Override the super class method in Sub Class (SuperClass and SubClass method names must be same)

- In this overriding method Print the "experienceYears"

- Also In this method Call the Super Class method using Super keyword, It should print super class instance variables data (1 Point)

Enter your answer

16

d) In "Subject.java" file

-Create instance variables subjectId and subjectNames variables with private access modifier

subjectId (Datatype is int)

subjectNames (Datatype is char array)

-Write a Parameterized Constructor - Initialize the SubjectId and SubjectNames variables data

-Write the getter methods for these two variables - subjectId and subjectNames (1 Point)

Enter your answer

17

e) In "ClassRoom.java" file

-Create instance variables classId and classNames variables with private access modifier

classId (Datatype is int)

classNames (Datatype is char array)

-Write a Parameterized Constructor - Initialize the classId and classNames variables data

-Write the getter methods for these two variables - classId and classNames (1 Point)

Enter your answer

18

(1 Point)

Enter your answer