OBJECT ORIENTED PROGRAMMING WITH JAVA 8– LAB 6

Q1. Create an abstract class Vehicle with one abstract function color(). Create two sub classes Car and Bus from the Vehicle class. Invoke the function through the instance of Car and Bus. Also use the abstract class reference that invokes that function in main.

Program:

```
abstract class Vehicle
 {
  abstract void color();
 }
 class Car extends Vehicle
 {
  void color()
   {
     System.out.print("The color of the car is Red");
   }
 }
 class Bus extends Vehicle
 {
  void color ()
    {
      System.out.println("The bus color is Blue");
    }
 }
 class Vehiclemain
 {
   public static void main(String args[])
    {
      Vehicle car = new Car();
          Vehicle bus = new Bus();
          Car carInstance = new Car();
      Bus busInstance =new Bus();
          car.color();
          bus.color();
          carInstance.color();
          busInstance.color();
    }
 }
```

Output:

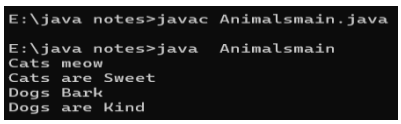
```
E:\java notes>javac Vehiclemain.java

E:\java notes>java Vehiclemain
The color of the car is RedThe bus color is Blue
The color of the car is RedThe bus color is Blue
```

Q2. Create an abstract class Animals with two abstract methods eat() and makeNoise(). Now create a class Cats where makeNoise() method is overridden which prints "Cats meow" and a class Dogs where makeNoise() method is overridden which prints "Dogs bark", both inheriting the class Animals. Now create an object for each of the subclasses and call their respective methods.

Program:

```
abstract class Animals
{
  abstract void eat();
  abstract void makeNoise();
}
class Cats extends Animals
{
  void makeNoise()
    {
       System.out.println("Cats meow");
    }
  void eat()
    {
       System.out.println("Cats are Sweet");
    }
}
class Dogs extends Animals
{
  void  makeNoise()
   {
     System.out.println("Dogs Bark");
   }
  void eat ()
   {
     System.out.println("Dogs are Kind");
   }
}
class Animalsmain
{
  public static void main(String [] args)
   {
      Animals cat = new Cats();
      Animals dog = new Dogs();
      cat.makeNoise();
      cat.eat();
      dog.makeNoise();
      dog.eat();
    }
}
```

Output:



Q3. Write an interface called Result with a method checkStatus(int mark) that returns a
boolean value. The checkStatus() method should return true if the mark is greater than or equal to 50 else false. Write another interface called Classify with a method checkGrade(int average) which returns a string. The checkGrade() method must return “First class” when the parameter average is 60 or more, “ Second class” when average is 50 or more but below 60, “Third class” when average is less than 50. Write a class called Exam which implements both Result and Classify. Now create an object for Exam class and call their respective methods.

Program :

```
            interface Result
{
   boolean checkStatus(int mark);
}
interface Classify
{
  String checkGrade(int average);
}
class Class1 implements Result,Classify
{
  public boolean checkStatus(int mark) {
      return mark >=50;
     }
  public String checkGrade(int average)
     {
      if (average>= 60) {
        return "First Class";
      }
      else if (average>=50) {
        return "Second Class";
      }
      else {
         return "Third Class";
           }
    }
 }
class ResultClassify
{
 public static void main(String args[])
  {
    Class1 c = new Class1();
    int mark = 65;
    boolean passed = c.checkStatus(mark);
    int average = 68;
    String grade = c.checkGrade(average);
    System.out.println("Marks :" + mark + ",Status :" + (passed? "Passed" : "Failed"));
    System.out.println("Average : " +average + ",Grade : " +grade);
  }
}
```

Output:

```
E:\java notes>java ResultClassify
Marks :65,Status :Passed
Average : 68,Grade : First Class
```

Q4**.** Create one interface named Sample which contains a variable named x and a method meth1(). Implement the interface in a class and implement the method to print the value of the variable. Invoke the method using interface reference**.**

Program:

```
interface Sample
{
  int x = 1;
  void meth1();        //abstract method meth1
}
class sampleImpl implements Sample {
  public void meth1() {
     System.out.println("X : " + x);
   }
}
class SampleMain
{
   public static void main(String [] args)
    {
      sampleImpl s = new sampleImpl();
      Sample interfaceRef = s;
      interfaceRef.meth1();
    }
}
```

Output:

```
E:\java notes>javac SampleMain.java

E:\java notes>java SampleMain
X : 1
```

Q5. Create an interface, with one abstract method to find square of a number and a default method to find cube of a number. Write the implementing class and invoke the two methods.

Program:

```
interface Number
{
int Square(int num);    // abstract method to find square of number
default int cube(int num)   //default method to find cube of number
  {
    return num * num * num;
  }
}
class findSquareCube implements Number    // implementing class
{
```

```
  public int Square(int num) {
       return num * num;
      }
 }
class SquareCube
{
  public static void main(String args[])
   {
    findSquareCube f = new findSquareCube();
    int square = f.Square(9);
    System.out.println("Square of 9: " + square);
    int cube = f.cube(4);
    System.out.println("Cube of 4 :" + cube);
    }
}
```

Output:

```
E:\java notes>javac SquareCube.java

E:\java notes>java SquareCube
Square of 9: 81
Cube of 4 :64
```

Q6. Create a class named A with following data members:
protected int a
private int b
int c
public int d
Create a no argument constructor which initializes the variables with values a=10, b=20, c=30 and d=40. Create a subclass named B in another package with main function which creates an object of B and print the values of all variables.

Program:

```
package packageA;
public class A
{
 protected int a;
 private int b;
 int c;
 public int d;

 public A()
  {
   a = 10;
   b = 20;
   c = 30;
   d = 40;
  }
}
```

```
import packageA.A;
public class B extends A
{
  public static void main(String args[])
   {
     B b = new B();
     System.out.println("a : " + b.a);
     //System.out.println("a : " + b.a);
     //System.out.println("c : " + b.c);
     System.out.println("d : " + b.d);     }}
```

Output:

```
E:\java notes>javac B.java

E:\java notes>java B
a : 10
d : 40
```

Q7. Write a Java class called Sumn which contains a method that finds the sum of all numbers from 1 upto the given number n. This class is in the directory with path d:\yourname\pack. Write a main program to access this package and print the sum.

Program:

```
package RASHI.pack;

public class Sumn{
  public static int findSum(int n)
   {
      int sum = 0;
      for (int i = 1; i <= n ; i++) {
         sum += i;
       }
       return sum;
    }
}
```

```
import RASHI.pack.Sumn;
   package RASHI.pack;
   public class SumnMain
   {
     public static void main(String args[])
      {
        int n = 10;
        int sum = Sumn.findSum(n);
        System.out.println("Sum of the number from 1 to " + n + "is" + sum);
       }
   }
```