

Institute of Engineering & Management
Department of Computer Science & Engineering
Object Oriented Programming (IT) Lab for 3rd year 5th semester 2018
Code: CS594D

Date: 28/08/18

WEEK-8

Assignment-1

Problem Statement: Create an abstract class Shape with two abstract methods, area() & disp(). Now design three concrete classes Rectangle, Circle & Triangle can compute area and display its separately.

Now check the following and find out the reasons:

- i) Abstract class can be final
- ii) Abstract method can be final
- iii) Abstract method can be static

Source code:

```
import java.util.Scanner;

abstract class Shape{
    protected double x, y, area;
    Shape(){x=y=0;}
    Shape(double x){this.x=x;}
    Shape(double x, double y)
    {
        this.x = x;
        this.y = y;
    }
    abstract void area();
    abstract void disp();
}

class Rectangle extends Shape{
    Rectangle(double x, double y){super(x,y);}
    Rectangle(){super();}
    void area(){this.area = x*y;}
    void disp()
    {
        System.out.println("Area of Rectangle: "+area);
    }
}

class Triangle extends Shape{
    Triangle(double x, double y){super(x,y);}
    Triangle(){super();}
    void area(){this.area = 0.5*x*y;}
    void disp()
    {
        System.out.println("Area of Triangle: "+area);
    }
}

class Circle extends Shape{
    Circle(double x){super(x);}
    Circle(){super();}
    void area(){this.area = Math.PI*x*x;}
```

```

    void disp()
    {
        System.out.println("Area of Circle: "+area);
    }
}

class Main
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        Shape sp;
        System.out.print("Enter the value of x & y: ");
        double x = sc.nextDouble(), y = sc.nextDouble();
        sp = new Rectangle(x,y);
        sp.area(); sp.disp();
        sp = new Triangle(x,y);
        sp.area(); sp.disp();
        sp = new Circle(x);
        sp.area(); sp.disp();
        sc.close();
    }
}

```

Screen-Shot:

```

rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/shape$ javac a1.java
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/shape$ java Main
Enter the value of x & y: 2.5 3
Area of Rectangle: 7.5
Area of Triangle: 3.75
Area of Circle: 19.634954084936208
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/shape$ java Main
Enter the value of x & y: 3 4
Area of Rectangle: 12.0
Area of Triangle: 6.0
Area of Circle: 28.274333882308138
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/shape$

```

Assignment-2

Problem Statement: Give an example to show that multiple inheritance is not possible with explanation. Show that this can be implemented using Interface.

Source code:

```

class Class1{
    void print1(){
        System.out.println("1st Class");
    }
}

class Class2{
    void print2(){
        System.out.println("2nd Class");
    }
}

```

```

class Main extends Class1, Class2{
    public static void main(String args[])
    {
        Main obj = new Main();
        obj.print1();
        obj.print2();
    }
}

```

Screen-Shot:

```

rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/Multiple inheritance$ javac a2.java
a2.java:13: error: '{' expected
class Main extends Class1, Class2{
                        ^
1 error
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/Multiple inheritance$ █

```

Thus multiple inheritance is not possible in Java

Assignment-3

Problem Statement: Let's consider the example of vehicles like bicycle, car, bike, they have common properties like color, gear etc. and functionalities like speedup, change gear, speed_down(or apply brake) .Implement the functionalities of all these vehicles using interface.

Source code:

```

import java.util.Scanner;

interface Vehicle{
    void speedUp(double in);
    void changeGear(int gear);
    void speedDown(int dec);
}

class Car implements Vehicle{
    double speed;
    int gear;
    Car(){
        speed = 0;
        gear = 0;
    }
    Car(double s, int g){
        speed = s;
        gear = g;
    }
    public void speedUp(double in){
        speed += in;
    }
    public void changeGear(int gear){
        this.gear = gear;
    }
    public void speedDown(int dec){
        speed -= dec;
    }
}

class Bike implements Vehicle{

```

```

double speed;
int gear;
Bike()
{
    speed = 0;
    gear = 0;
}
Bike(double s, int g)
{
    speed = s;
    gear = g;
}
public void speedUp(double in)
{
    speed += in;
}
public void changeGear(int gear)
{
    this.gear = gear;
}
public void speedDown(int dec)
{
    speed -= dec;
}
}

```

```

class Bicycle implements Vehicle{
    double speed;
    int gear;
    Bicycle()
    {
        speed = 0;
        gear = 0;
    }
    Bicycle(double s, int g)
    {
        speed = s;
        gear = g;
    }
    public void speedUp(double in)
    {
        speed += in;
    }
    public void changeGear(int gear)
    {
        this.gear = gear;
    }
    public void speedDown(int dec)
    {
        speed -= dec;
    }
}

```

```

class Main
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of speed & gear: ");
        double s = sc.nextDouble();
        int g = sc.nextInt();
    }
}

```

```

        Car cr = new Car(s,g);
        Bike bk = new Bike(s,g);
        Bicycle bc = new Bicycle(s,g);
        cr.speedUp(5); bk.speedUp(3); bc.speedUp(1);
        cr.speedDown(2); bk.speedDown(4); bc.speedDown(2);
        cr.changeGear(4); bk.changeGear(2);
        System.out.println("The value of speed & gear of\n Car: "+cr.speed+"
"+cr.gear);
        System.out.println(" Bike: "+bk.speed+" "+bk.gear);
        System.out.println(" Bicycle: "+bc.speed+" "+bc.gear);
        sc.close();
    }
}

```

Screen-Shot:

```

rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/vehicle$ javac a3.java
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/vehicle$ java Main
Enter the value of speed & gear: 6 1
The value of speed & gear of
Car: 9.0 4
Bike: 5.0 2
Bicycle: 5.0 1
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/vehicle$ java Main
Enter the value of speed & gear: 5.7 3
The value of speed & gear of
Car: 8.7 4
Bike: 4.699999999999999 2
Bicycle: 4.7 3
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/vehicle$ █

```

Assignment-4

Problem Statement: Can we add any static method in an interface? Explain your answer.

Source code:

```

interface inter1{
    static void foo(){
        System.out.println("New Feature of Java 8");
    }
}

class Main{
    public static void main(String args[]){
        inter1.foo();
    }
}

```

Screen-Shot:

```

rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/Static_interface$ javac a4.java
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/Static_interface$ java Main
New Feature of Java 8
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/Static_interface$ █

```

Thus, we can add static method in an interface in java 8 or newer versions. But in previous version of java we cannot add static method in a interface.

Assignment-5

Problem Statement: Create two interface, each with one method, inherit a new interface from the two, adding a new method. Create a class by implementing the new interface also inheriting from a concrete class . Now write three methods, each of which takes one of the three interface as an argument. In main() create an object of your class and pass it to each of the methods.

Source code:

```
interface Inter1{
    void fool(Main obj);
}

interface Inter2{
    void foo2(Main obj);
}

interface Inter3 extends Inter1, Inter2{
    void foo3(Main obj);
}

class Class1{
    int gem = 8;
    public void fool(Main obj){
        System.out.println("Gem from 1st interface: "+obj.gem);
    }
    public void foo2(Main obj){
        System.out.println("Gem from 2nd interface: "+obj.gem);
    }
    public void foo3(Main obj){
        System.out.println("Gem from 3rd interface: "+obj.gem);
    }
}

class Main extends Class1 implements Inter3{
    public static void main(String args[]){
        Main obj = new Main();
        obj.fool(obj);
        obj.foo2(obj);
        obj.foo3(obj);
    }
}
```

Screen-Shot:

```
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/interface$ javac a5.java
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/interface$ java Main
Gem from 1st interface: 8
Gem from 2nd interface: 8
Gem from 3rd interface: 8
rana@ranajit:~/Git/College_programs/5th SEM/Java/Assignment 8/interface$ █
```