

ASSIGNMENT-2

- ① Write a inheritance program to implement an interface called shape. Now, create three classes circle, square, Triangle. The shape interface should contain three methods Radius, length, height and Base?

```
class Interface shape {  
    double radius();  
    double length();  
    double height();  
    double base();  
}
```

```
class circle implements shape {  
    private double radius;  
    public circle (double radius) {  
        this.radius = radius;  
    }  
}
```

@override

```
public double radius() {  
    return radius;  
}
```

}

@override

```
public double length() {  
    return 2 * math.PI * radius;  
}
```

}

@Override

```
public double base() {
```

```
    return side;
```

```
}
```

```
}
```

```
class Triangle implements Shape {
```

```
    private double base;
```

```
    private double height;
```

```
    public Triangle (double base, double height) {
```

```
        this.base = base;
```

```
        this.height = height;
```

```
}
```

@Override

```
public double radius() {
```

```
    return 0;
```

```
}
```

@Override

```
public double length() {
```

```
    return base + 2 * Math.sqrt(base * base + height * height);
```

```
}
```

@Override

```
public double height() {
```

```
    return height;
```

```
}
```

@Override

```
public double base() {
```



```
return base();
```

```
}
```

```
}
```

```
public class main {
```

```
    public static void main (String[] args) {
```

```
        circle circle = new circle (5);
```

```
        square square = new square (4);
```

```
        Triangle Triangle = new triangle (3, 6);
```

```
        System.out.println ("circle length." + circle.length());
```

```
        System.out.println ("square length" + square.length());
```

```
        System.out.println ("triangle length" + triangle.length());
```

```
    }
```

```
}
```

- ② write a program to illustrate polymorphism, create a class called parent contains 3 methods TV(), radio(int button, string, play(int)); phone(int, phnum, double price, string). Now create another class called child. This child class should override and overload the methods in parent class.

```
class parent {
```

```
    void TV() {
```

```
        System.out.println ("watching TV");
```

```
}
```



```
void radio (int button, string station) {  
    System.out.println ("Tuning to station" + station + "  
    on button" + button);  
}
```

}

```
void phone (int button, string phn num, double price,  
    string model) {
```

```
    System.out.println ("calling" + phn num + "using  
    phone model" + model + " (button: " + button + "  
    price: " + price + ")");  
}
```

}

}

```
class child extends parent {
```

```
@Override
```

```
void TV() {
```

```
    System.out.println ("child is watching TV");  
}
```

}

```
@Override
```

```
void radio (int button, string playlist) {
```

```
    System.out.println ("child is playing playlist" +  
    playlist + " on radio button" + button);  
}
```

}


```
void phone (int button, string phn num, double price,  
            string model) {
```

```
    System.out.println ("child " calling "+ phn num +  
        "using phone mode "+ model + "(Button" + button +  
        ", price : " + price + ")");
```

```
}
```

```
public class main {
```

```
    public static void main (String[] args) {
```

```
        parent parentobj = new child ();
```

```
        parentobj.Tv ();
```

```
        parentobj.radio (2, "Rock");
```

```
        parentobj.phone (1, "123-456-7890", 199.99,  
            "Smart phone");
```

```
}
```

```
}
```