slip8

```
interface Shape {

   double area();

}


 class Circle implements Shape {

    final double radius;


   public Circle(double radius) {

       this.radius = radius;

   }


   public double area() {

       return 3.14 * radius * radius;

   }

}


 class Sphere implements Shape {

  final double radius;


   public Sphere(double radius) {

       this.radius = radius;

   }


   public double area() {

       return 4 *3.14* radius * radius;

   }
}

public class s8q1 {
   public static void main(String[] args) {

       Circle circle = new Circle(5);
       System.out.println("Area of the Circle: " + circle.area());


       Sphere sphere = new Sphere(5);
       System.out.println("Area of the Sphere: " + sphere.area());
   }
}
```

```
import java.io.File;

class s8q2 {
    public static void main(String[] args) {

        String directoryPath = "D:\\JAVA";

        File directory = new File(directoryPath);

          if (directory.exists() && directory.isDirectory()) {

            File[] files = directory.listFiles();

            System.out.println("Text files in the directory:");

            for (File file : files) {
                if (file.isFile() && file.getName().endsWith(".txt")) {
                    System.out.println(file.getName());
                }
            }
        } else {
            System.out.println("The specified path is not a valid directory.");
        }
    }
}
```

```
def Repeateditems(inputtuple):
    repeated_items=set()

    for item in inputtuple:
        if inputtuple.count(item)>1:
            repeated_items.add(item)
    return repeated_items

tuple1=(1,2,3,4,5,2,3,4,5)
repeated=Repeateditems(tuple1)
```

```
print("original:", tuple1)

print("repeated items:",repeated)
```

```
class StringManipulator:

    def __init__(self):

        self.user_string = ""


    def get_String(self):



        self.user_string = input("Enter a string: ")


    def print_String(self):



        print("Uppercase String:", self.user_string.upper())


    def reverse_string(self):

        words = self.user_string.split()

        reversed_words = ' '.join(reversed(words))

        print("Reversed String in Lowercase:", reversed_words.lower())


s1= StringManipulator()


s1.get_String()
```

```
s1.print_String()


s1.reverse_string()
```