SIR SYED UNIVERSITY OF ENGINEERING & TECHNOLOGY
SOFTWARE ENGINEERING DEPARTMENT

Spring 2021

Object Oriented Programming (SWE-103T)
Assignment # 2

Semester: II **Batch: 2020F**
Due Date: 30-May-2021 **Max Points: 10**

Instructions:

- Attempt all questions
- Do not copy or cheat from any one, make your own effort. If any assignment found copied it will straight away be rejected.
- Take the help from internet or any other online sources is allowed. If the data has been taken from any online source, then please organize it properly, do not copy and paste exactly.
- There is no maximum page limit.
- Assignment should be in MS Word document. The Word file should be submitted by converting it to a single PDF file.
- Mention your full name and roll number on the first / front page. The page numbers must be mentioned at the right bottom corner in the whole assignment.
- The maximum font size for text is 12 and for heading is 14. The font could be Arial or the Time New Roman.

Name: Abdul Moiz Chishti
Roll NO: SE20F-022

Question 1 **[3 points]**

Describe the concept of inheritance? In general how many types of inheritance can be applied in programming languages? Which type of inheritance java supports? Explain your answer with suitable examples

Concept Of Inheritance:

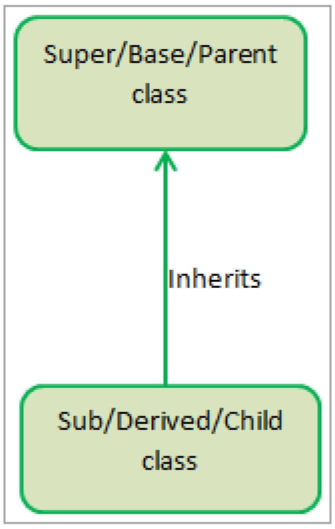
Inheritance is the process of creating a new Class, called the Derived Class , from the existing class, called the Base Class . The Inheritance has many advantages, the most important of them being the reusability of code. Rather than developing new Objects from scratch, new code can be based on the work of other developers, adding only the new features that are needed. The reuse of existing classes saves time and effort.

However, inheritance may be implemented in different combinations in Programming languages and they include:

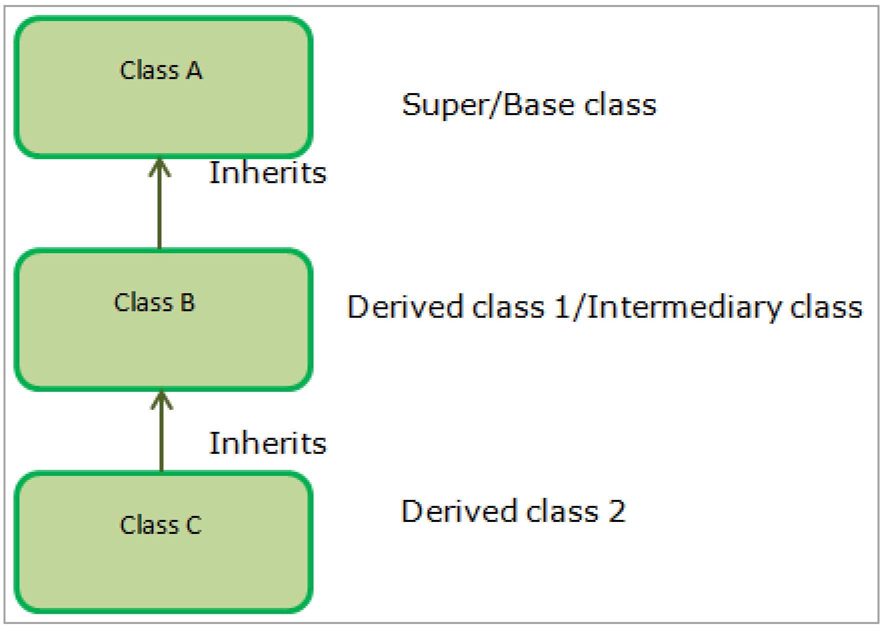
- Single Inheritance
- Multi Level Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multipath inheritance
- Multiple Inheritance

Java supports only single, multilevel, and hierarchical type of inheritance using classes. Java does not support multiple and hybrid inheritance with classes.

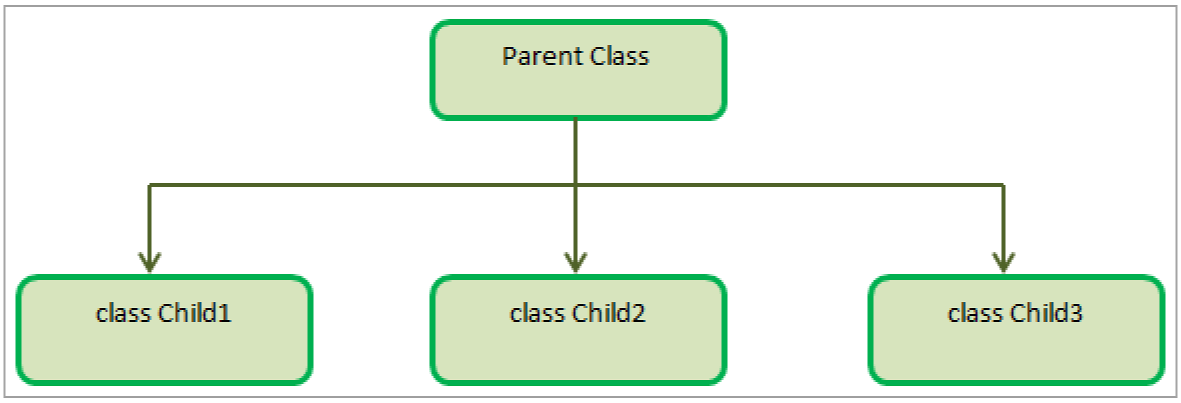
Single Inheritance In Java:



Multilevel Inheritance In Java:



Hierarchical Inheritance In Java:



Question 2 [2 points]

What are the difference between abstract class and interface? Explain your answer with suitable example?

Abstract class	Interface
1) Abstract class can have abstract and non-abstract methods.	Interface can have only abstract methods. Since Java 8, it can have default and static methods also.
2) Abstract class doesn't support multiple inheritance.	Interface supports multiple inheritance.

3) Abstract class can have final, non-final, static and non-static variables.	Interface has only static and final variables.
4) Abstract class can provide the implementation of interface.	Interface can't provide the implementation of abstract class.
5) The abstract keyword is used to declare abstract class.	The interface keyword is used to declare interface.

9)Example:
public abstract class Shape{
public abstract void draw();
}
	Example:
	public interface Drawable{
	void draw();
	}

Question 3 **[5 points]**

Construct java abstract class Shape containing **private** information of radius (int) and height (int). The class contains the following for calculations:

- Default constructor setting radius and height to 5.
- A parameterized constructor setting radius and height to values provided in parameters.
- An abstract method volume () returning type double.
- An abstract method surface_area () returning type double

Then construct a child class Sphere inherit from Shape class. The class contains the following:

- Default constructor calling parent class default constructor to set radius and height to 5.
- A parameterized constructor calling parent class constructor for setting radius and height to values provided in parameters.
- Provide implementation of abstract volume method to calculate and return Volume of Sphere (V=4/3*PI*R^3).
- Provide implementation of surface area method to calculate and return Surface Area of Sphere (SA=4*PI*R^2).

Then construct a child class Cylinder inherit from Shape class. The class contains the following:

- Default constructor calling parent class default constructor to set radius and height to 5.
- A parameterized constructor calling parent class constructor for setting radius and height to values provided in parameters.
- Provide implementation of abstract volume method to calculate and return Volume of Cylinder (V=PI*R^2*H).
- Provide implementation of surface area method to calculate and return Surface Area of Sphere (SA=2*PI*R^2 + 2*PI*R*H)

Make sure to use Math and Scanner class for computing results and inputs. Then construct a Testing class containing main() method and create **ONE** object of Sphere and Cylinder class. Initialize both objects using different constructor. Finally compute and display Volume of Sphere and Cylinder. Similarly compute and display Surface Area of Cylinder.

Source Code:

Shape Class:

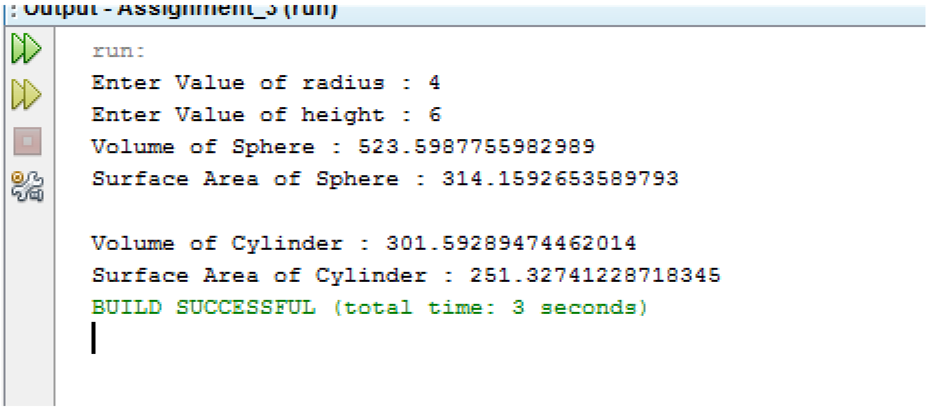
```
import java.util.Scanner;
abstract class Shape{
    Scanner sc=new Scanner(System.in);
    protected int radius,height;
    Shape(){
        radius=5;
        height=5;
    }
    Shape(int r,int h){
        System.out.print("Enter Value of radius : ");
        r=sc.nextInt();
        System.out.print("Enter Value of height : ");
        h=sc.nextInt();
        radius=r;
        height=h;
    }
    abstract double volume();
    abstract double suface_area();
}
class Sphere extends Shape{

    Sphere(){
        radius=5;
        height=5;
    }
    Sphere(int radius,int height){
        super(radius, height);
}

    @Override
    double volume() {
        double volume=(4.0/3.0)*(Math.PI)*(Math.pow(radius, 3));
        return volume;
    }

    @Override
    double suface_area() {
       double surface_area=(4*(Math.PI)*(Math.pow(radius, 2)));
```

```
        return surface_area;
    }
}
class Cylinder extends Shape{

    Cylinder(){
        radius=5;
        height=5;
    }
    Cylinder(int radius,int height){
        super(radius, height);
}

    @Override
    double volume() {
        double volume=(Math.PI)*(Math.pow(radius, 2))*(height);
        return volume;
    }

    @Override
    double suface_area() {
       double surface_area=(2*(Math.PI)*(Math.pow(radius, 2))) + (2*(Math.PI)*radius*height);
       return surface_area;
    }
}
```

Main Class:

```
public class Assignment_3 {
    public static void main(String[] args) {
        Sphere sp=new Sphere();
        Cylinder cy=new Cylinder(0,1);
        System.out.println("Volume of Sphere : "+ sp.volume());
        System.out.println("Surface Area of Sphere : "+sp.suface_area());
        System.out.println("\nVolume of Cylinder : "+cy.volume());
        System.out.println("Surface Area of Cylinder : "+cy.suface_area());
    }
}
```

Output:



```
run:
Enter Value of radius : 4
Enter Value of height : 6
Volume of Sphere : 523.5987755982989
Surface Area of Sphere : 314.1592653589793

Volume of Cylinder : 301.59289474462014
Surface Area of Cylinder : 251.32741228718345
BUILD SUCCESSFUL (total time: 3 seconds)
```