**QUESTION NO :- 01**

public interface MyFirstInterface {

int x = 10;

void display();

}

------------------- 1 ------------------

public interface MyInterface {

int variable1 = 10;

public static final int variable2 = 20;

}

Public: Both variables are implicitly public, meaning they can be accessed from outside the interface.

Static: Both variables belong to the interface itself, not to any specific instance of a class implementing the interface.

Final: Both variables are implicitly final, and their values cannot be changed once assigned.

So there is no practical difference between declaring variables with or without the public static final keywords in an interface.

------------------- 2 ------------------

public interface MyInterface {

void method1();

abstract void method2();

}

In Java interfaces, all methods are implicitly abstract, regardless of whether you explicitly use the "abstract" keyword or not. So, there is no difference between declaring methods with or without the "abstract" keyword in an interface.

------------------- 3 ------------------

public class InterfaceImplemented implements MyFirstInterface {

@Override

public void display() {

// Try to change the value of x

x = 20; // This will result in a compilation error

System.out.println("Value of x: " + x);

}

}

This code will result in a compilation error due to the attempt to change the value of "x" inside the "display()" method.

The reason is that in Java interfaces, all variables are implicitly public, static, and final, which means they are constants and their values cannot be modified after initialization. When you try to change the value of "x" inside the "display()" method or any other method in the implementing class, the Java compiler will generate an error because you cannot reassign a value to a final variable.

2.

Interface Speaker

package com.mycompany.practicals;

public interface Speaker {

public void speak();

}

Class Priest

package com.mycompany.practicals;

public class Priest implements Speaker {

@Override

public void speak(){

System.out.println("preachhh");

}

}

Class Politician

package com.mycompany.practicals;

public class Politician {

public void speak(){

System.out.println("Politicalll");

}

}

Class Lecturerr

package com.mycompany.practicals;

public class Lecturerr {

public void speak(){

System.out.println("Lecturing");

}

}

Main Method

package com.mycompany.practicals;

public class Practicals {

public static void main(String[] args) {

Priest p= new Priest();

p.speak();

Politician po=new Politician();

po.speak();

Lecturerr le= new Lecturerr();

le.speak();

}

}

Output:

--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.1.0:exec (default-cli) @ Practicals ---

preachhh

Politicalll

Lecturing

------------------------------------------------------------------------

BUILD SUCCESS

------------------------------------------------------------------------

3.The code builds successfully but nothing runs.

Method in the parent class Student, Display() method is declared as Final. Hence it cannot be overridden by the child class.

Variable Marks is also declared Final which makes it impossible to change the amount which has been assigned to that variable.

Student class itself is declared as Final. A class which has been declared Final cannot be extended by a subclass.

4.

Abstract Class Shape

package com.mycompany.lab5;

abstract class Shape {

protected double area;

abstract double calculatearea();

public void display(){

System.out.println("Area: " +calculatearea());

}

}

Class Rectangle

package com.mycompany.lab5;

public class Rectangle extends Shape{

protected double l;

protected double w;

public void setvalue(double l, double w) {

this.l=l;

this.w=w;

System.out.println("Rectangle");

}

@Override

public double calculatearea(){

return area=l\*w;

}

}

Class Circle

package com.mycompany.lab5;

public class circle extends Shape {

protected double r;

public void setvalue(double r){

this.r=r;

System.out.println("Circle:");

}

@Override

public double calculatearea(){

return area= 3.14\*r\*r;

}

}

Main Method

package com.mycompany.lab5;

public class Lab5 {

public static void main(String[] args) {

Rectangle r1=new Rectangle();

r1.setvalue(5.00, 7.00);

r1.calculatearea();

r1.display();

circle c1=new circle();

c1.setvalue(5.00);

c1.calculatearea();

c1.display();

}

}

Output

--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.1.0:exec (default-cli) @ Lab5 ---

Rectangle

Area: 35.0

Circle:

Area: 78.5

------------------------------------------------------------------------

BUILD SUCCESS