# Practical 05

## Exercise 01

1. public interface MyFirstInterface

{

int x = 10; // With or without 'public static final' - Both are implicitly present

void display();

}

1. public interface MyFirstInterface

{

int x = 10;

void display(); // With or without 'abstract' - It is always treated as an abstract method

}

1. public class InterfaceImplemented implements MyFirstInterface

{

@Override

public void display()

{

x = 20; // Trying to change the value of x - This will result in a compile-time error

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

}

}

## Exercise 02

package com.mycompany.peactica5;

public interface speaker

{

public void speak();

}

package com.mycompany.peactica5;

public class priest implements speaker

{

@Override

public void speak()

{

System.out.println("Iam Priest");

}

}

package com.mycompany.peactica5;

public class lecture implements speaker

{

@Override

public void speak()

{

System.out.println("Iam Lecture");

}

}

package com.mycompany.peactica5;

public class politician implements speaker

{

@Override

public void speak()

{

System.out.println("Iam Politician");

}

}

package com.mycompany.peactica5;

public class Peactica5

{

public static void main(String[] args)

{

priest p1=new priest();

lecture i1=new lecture();

politician po1=new politician();

p1.speak();

i1.speak();

po1.speak();

}

}

## Exercise 03

**The compilation error** occurs because the **display()** method in the **Student** class is declared as **final**, but it does not have a method body (implementation). In Java, when you declare a method as **final**, you are stating that it cannot be overridden by subclasses. However, a **final** method must have an implementation in the class where it is declared. Since there is no implementation for the **display()** method, the code will not compile.

## Exercise 04

package com.mycompany.ex04;

abstract public class shape

{

abstract double calarea();

public void display()

{

System.out.println("Area"+calarea());

}

}

package com.mycompany.ex04;

public class circle extends shape

{

private double radius;

@Override

double calarea()

{

return 22/7\*(radius\*radius);

}

public circle(double radius)

{

this.radius=radius;

}

}

package com.mycompany.ex04;

public class rectangle extends shape

{

public double height;

public double weight;

@Override

double calarea()

{

return height\*weight;

}

public rectangle(double height, double weight)

{

this.height=height;

this.weight=weight;

}

}

package com.mycompany.ex04;

public class Ex04 {

public static void main(String[] args)

{

circle c1=new circle(15);

rectangle r1=new rectangle(5,5);

c1.display();

r1.display();

}

}