Practical 5: Inheritance & Abstract Classes

Exercise 01:

```
package com.mycompany.main;
public interface MyFirstInterface
{
   int x=10;
   void display();
}
package com.mycompany.main;
public class InterfaceImplemented implements MyFirstInterface
{
   public void display()
   {
      System.out.println("Value of X: "+x);
   }
}
package com.mycompany.main;
public class Main {

   public static void main(String[] args)
   {
      InterfaceImplemented obj = new InterfaceImplemented();
```

```
        obj.display();
        }
}
```

1. Declaring the variable x without the public static final keywords is the same as declaring it with those keywords. All variables in interfaces are implicitly public, static, and final. As a result, including or excluding these keywords makes no difference in this circumstance.

2. It is equivalent to declaring an abstract method in the interface with or without the abstract keyword. All methods in interfaces are implicitly abstract, and the abstract keyword is not necessary.

3. The value of x in the InterfaceImplemented class cannot be modified since it is marked as final in the interface. Within the interface, the x variable is effectively a constant, and any effort to change its value will result in a compilation error.

Exercise 02:

Exercise 03:

Exercise 04:

```
package com.mycompany.abstractclassex;
public abstract class shape
{
   public abstract double calculateArea();
   public void display()
   {
      System.out.println("Shape Details");
      System.out.println("Area: "+calculateArea());
   }
}
package com.mycompany.abstractclassex;
public class Circle extends Shape
{
   private double radius;
   public Circle1(double radius)
   {
         this.radius=radius;
```

```
    }
    public double calculateArea()
    {
        return Math.PI*radius*radius;
    }
}
package com.mycompany.shape;
public class Rectangle extends Shape
{
  private double length;
  private double width;
  public Rectangle1(double length, double width)
  {
     this.length=length;
     this.width=width;
  }
  public double calculateArea()
  {
     return length*width;
  }
}
package com.mycompany.abstractclassex;
```

```
public class AbstractClassEx
{
    public static void main(String[] args)
    {
     Circle circle=new Circle(6);
     circle.display();
     Rectangle rectangle=new Rectangle(5,7);
     rectangle.display();
    }
}
```