Lab Sheet 6

Exercise 1

1. In Java, variables declared within an interface are implicitly public, static, and final. Therefore, explicitly using public static final keywords while declaring variables in an interface is redundant and unnecessary. Both x and y in the above example are equivalent. They are constants and cannot be changed once they are assigned a value.

2. In Java, all methods declared within an interface are implicitly abstract, even if the abstract keyword is not used. So, explicitly using the abstract keyword while declaring methods in an interface is optional and not required.

3. In Java, when a variable is declared within an interface, it is implicitly considered as a public static final constant. Constants are meant to have a fixed value that cannot be changed once assigned. In the "InterfaceImplemented" class, when we try to modify the value of x inside the display() method, it results in a compilation error because x is final and cannot be changed.

   Since x is constant and shared across all instances of the implementing class, it is considered a class-level constant rather than an instance-level variable. Therefore, it is common for all objects of the implementing class, and any attempt to modify its value will be disallowed by the compiler.

Exercise 2

MAIN

```
package com.mycompany.main;

public class Main
{
   public static void main(String[] args)
   {
      Speaker politician = new Politician();
      politician.speak();

      Speaker priest = new Priest();
      priest.speak();

      Speaker lecturer = new Lecturer();
      lecturer.speak();
   }
}
```

Politician Class

```
package com.mycompany.main;

class Politician implements Speaker
{
   @Override
   public void speak()
   {
      System.out.println("I am a politician.");
   }
}
```

Priest Class

```
package com.mycompany.main;

class Priest implements Speaker
{
   @Override
   public void speak()
   {
      System.out.println("I am a priest.");
   }
}
```

Lecturer Class

```
package com.mycompany.main;

class Lecturer implements Speaker
{
   @Override
   public void speak()
   {
      System.out.println("I am a lecturer.");
   }
}
```

Speaker Interface

```
package com.mycompany.main;

interface Speaker
{
   void speak();
}
```