Exercise 01:

Answer

01.

```
interface MyFirstInterface {
   int x = 10; // Integer type variable
   void display(); // Abstract method
}
```

I.

```
// Declaring the variable without public static final keywords

interface MyFirstInterface {
   int x = 10;
}

// Declaring the variable with public static final keywords

interface MyFirstInterface {
   public static final int x = 10; // Also implicitly public, static, and final
}
```

Yes, Interface variables are constants, and their values cannot be changed. Making them implicitly final enforces this constraint, ensuring that implementing classes cannot modify the values of these variables.

(or)

The final modifier ensures the value assigned to the interface variable is a true constant that cannot be re-assigned by program code

II.

// Declaring the abstract method without the abstract keyword.

```
interface MyFirstInterface {
   void display(); // Implicitly abstract
}

// Declaring the abstract method with the abstract keyword

interface MyFirstInterface {
   abstract void display(); // Also implicitly abstract
}
```

No, when declaring an abstract method inside an interface, using the abstract keyword is optional. There is no difference between declaring the abstract method with or without the abstract keyword. Both approaches are equivalent, and the method will be treated as abstract in either case.

III.

```
interface MyFirstInterface {
   int x = 10;

      abstract void display();
}

class InterfaceImplemented implements MyFirstInterface {

int x = 20; // This is a separate instance variable in the class, not related to the interface variable
x

   // Overriding all the abstract methods
   public void display() {
      x = 50;
      System.out.println("New Value of x is: " + x);
   }
}

public class Main {
   public static void main(String[] args) {
      InterfaceImplemented abj = new InterfaceImplemented();
      abj.display();
      //or we can call only value of x
      System.out.println("Value of x outside InterfaceImplemented: " + abj.x);
   }
}
```

We can change the value of the instance variable x inside the implementing class. However, we cannot change the value of the interface variable x.
Interface variables are implicitly public static final, making them constants, and their values cannot be modified once assigned.