

Exercise 01:

1. Declare an interface called "MyFirstInterface".

```
public interface MyFirstInterface {  
  
    int x = 10;  
  
    void display();  
  
}
```

Declaring variable with public static final keywords

```
public interface MyFirstInterface {  
  
    public static final int x = 10; // Variable x (public, static, and final)  
  
    void display(); // Abstract method (implicitly public and abstract)  
  
}
```

Declaring variable without public static final keywords

```
public interface MyFirstInterface {  
  
    int x = 10;  
  
    void display();  
  
}
```

- There is no difference between declaring the variable with and without the **public static final** keywords in Java interfaces, as all variables in interfaces are implicitly public, static, and final.

2. With abstract keyword:

```
public interface MyFirstInterface {  
  
    abstract void display(); // Abstract method (implicitly public)  
  
}
```

Without abstract keyword:

```
public interface MyFirstInterface {  
  
    void display(); // Abstract method (implicitly public)  
  
}
```

- There is no difference between these two approaches as all methods declared in an interface are implicitly abstract whether or not the abstract keyword is explicitly used.

3. Implement the "MyFirstInterface" in a class "InterfaceImplemented" and try to change the value of x inside this method

```
public class InterfaceImplemented implements MyFirstInterface {  
  
    public void display() {  
  
        System.out.println("Value of x: " + x);  
  
    }  
  
}
```

- it is not possible to change the value of the variable x inside the class Interface Implemented as x is declared as a constant variable (implicitly public, static, and final) in the interface MyFirstInterface, and constants cannot be modified after they are assigned a value.