Lab 06 exercise 01

```
public interface MyFirstInterface {
    int x = 10;
  void display();
}
public class InterfaceImplemented implements MyFirstInterface {
    @Override
    public void display() {

        x = 20;

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

01. Variables are automatically public, static, and final in Java interfaces whether or not you explicitly declare them with these keywords. Therefore, it makes no difference whether the variable x is declared in the interface with or without the public static final keywords. The outcomes of both methods will be the same, and x will be regarded as a constant variable that cannot be changed once assigned.

02. There is no need to explicitly use the abstract keyword when specifying abstract methods in an interface since all methods in interfaces are inherently abstract. The show() function will be regarded as abstract and required to be implemented in the classes that implement the interface because both strategies will provide the same behaviour.

03. As was previously established, variables in interfaces are implicitly final, which means that once they have been assigned a value, they become constants and cannot be modified. As a result, there will be a compilation error if you attempt to modify the value of x inside the show() function of the "InterfaceImplemented" class. No effort shall be made to change a final variable,