

Name: Sachini Dissanayake

ID: 27357

Questions	Code
Exercise 1	<pre>// Interface declaration interface MyFirstInterface {     int x = 10; // Integer type variable (implicitly public, static, and final)      // Abstract method (implicitly public and abstract)     void display(); }</pre>
1	<pre>// Interface declaration interface MyFirstInterface {     int x = 10; // Implicitly public, static, and final     void display(); // Abstract method, implicitly public and abstract }</pre> <p>There's no significant difference between declaring the variables because the variable will be implicitly public, static, and final in both cases, making it a constant value in any class that implements this interface.</p>
2	<pre>// Interface declaration interface MyFirstInterface {     int x = 10;     void display(); // Implicitly abstract and public }</pre> <p>There is no difference between declaring the method display() with/without the abstract keyword inside the interface. In both cases, the method will be implicitly abstract and public.</p>
3	<pre>class InterfaceImplemented implements MyFirstInterface {     // Implementing the abstract method from the interface     @Override     public void display() {         // You cannot change the value of x here as it's implicitly final.         // Uncommenting the next line will result in a compilation error.         // x = 20;         System.out.println("Value of x: " + x);     } }</pre>

	In the InterfaceImplemented class, you cannot change the value of x inside the display() method because it is implicitly final in the interface. All variables declared inside an interface are implicitly final, which means they cannot be reassigned once initialized. If you try to change the value of x in the display() method, you will get a compilation error
--	---
