Lab Session

Practical 05: Inheritance & Abstract Classes

SGCM Dematapitiya - 29056

Exercise 01:

1.

* With public key word

public interface MyFirstinterface{

public int x;

// Abstract method

void display();

}

* Without public keyword

public interface MyFirstinterface{

int x;

// Abstract method

void display();

}

* There is no difference between these two approaches. Public keyword is optional.

2.

* With abstract keyword

public interface MyFirstinterface{

// Abstract method

abstract void display();

}

* Without abstract keyword

public interface MyFirstinterface{

// Abstract method

void display();

* when declaring an abstract method within an interface, it is not required to use the abstract keyword. The method is abstract by default if the abstract keyword is not present. There is no difference between defining an abstract method within an interface with or without the abstract keyword.

3.

* In the "InterfaceImplemented" class, we overrode the "display()" function from the "MyFirstInterface" interface.

So, attempting to modify the value of the variable "x" within the "display()" metho will result in a compilation error.

This issue occurs because interface variables are implicitly treated as public static final, which means they are constants whose values cannot be changed.

Any attempt to change the value of "x" after it is declared in the interface will result show error.

public class InterfaceImplemented implements MyFirstInterface {

@Override

public void display() {

x = 20; // Invalid (constant)

System.out.println("The value of x is: " + x);

}

}