**CIS 181 - Lab 10**

**Interface and Polymorphism**

## ****Objectives****

* To gain a better understanding of Interface and Polymorphism.
* To gain a better understanding of overriding methods.
* To become familiar with abstract method which is used for inheritance.

## ****Description****

Polymorphism means "many forms" (poly = many, morphism = form). When a single function can be applied to various types of arguments, pure polymorphism occurs. There is one function and multiple explanations. When we have many different functions that are all represented by the same name, another extreme situation occurs: this situation is called overloading. Between these two methods are the overriding method and the deferred method. The overriding function and overloading function are different. Overriding function is related to polymorphism, which is belonged to dynamic binding and occurs during running time. The overload function is belonged to static binding, which occurs during compilation time.

The interface formalizes polymorphism. Interfaces allow us to define polymorphism in a declarative way with independence of implementation. If two elements implement the same interface, they are polymorphic for a set of behaviors. You always hear that polymorphism is the great benefit of object-oriented programming, but without interfaces, it cannot be implemented, verified or even expressed in a non-mandatory or language-specific way. The formalization of the interface removes the mystery and provides us with a good way to accurately describe what polymorphism has been trying to do. The interface is testable, verifiable and accurate.

As stated in the official document (in the below link) of Java language, “Implementing an interface allows a class to become more formal about the behavior it promises to provide. Interfaces form a contract between the class and the outside world, and this contract is enforced at build time by the compiler. If your class claims to implement an interface, all methods defined by that interface must appear in its source code before the class will successfully compile.”.

<https://docs.oracle.com/javase/tutorial/java/concepts/interface.html>

In this lab, you will write your own code to practice and experience interface and polymorphism using Java.

**Exercises**

1. Download the following code

* Demo01.java
* IP01.java
* Main.java
* USB.java
* Keyboard.java
* Mouse.java
* Laptop.java

1. In Eclipse, create a new Java project called "Lab 10", and import the above 2 files into a “interfacepolymorphism” package.
2. Compile and run the program of Demo01.java and IP01.java. You don’t need to modify any code in source files of Demo01.java and IP01.java. They are used to help you be familiar with interface and its implementation. It should print out the following message:

method 1

method 2

method 3

method 4

1. Let’s focus on other 5 source files.

Main.java

USB.java

Keyboard.java

Mouse.java

Laptop.java

USB.java provides an interface, which has two methods: open() and close(). They are not implemented in USB interface, since they are used for being implemented in classes of Keyboard and Mouse. Both Keyboard and Mouse classes implement the interface USB. You don’t need to modify any code in USB.java.

For Keyboard.java file, you need to override both open() and close() methods, and output corresponding messages indicated in the source file comments. There is another method click(), which has been implemented in Keyboard.java file. The method click() is belonged to the class Keyboard. It is not an overriding method.

Similarly, you implement open() and close() methods in Mouse.java file. The corresponding output messages are given in Mouse.java file as comments.

For the LapTop.java file, you don’t need to modify any code. It defines a LapTop class, which has multiple methods and attributes. Read those methods and attributes in the LapTop class. You can call those methods and use those attributes in the Main class later.

Lastly, Main.java should be modified and some missing code should be filled out. You need to create interfaces for keyboard and mouse, respectively. Hint: the method useUsb() defined in the class LapTop will be used here. The method open() will also be used here.

Please refer code presented in the Demo01.java and IP01.jave to write overriding methods. For source files Keyboard.java, Mouse.java, and Main.java, note that you add source code between two comments below.

// coding starts here

// coding ends here

1. After you fill out the source files Keyboard.java, Mouse.java, and Main.java, run your program of Main.java and should print out exactly the following messages:

start laptop

-------------------------------

Use USB connected devices

Keyboard USB connected successfully

-------------------------------

Use USB connected devices

mouse USB opened and connected successfully

-------------------------------

mouse USB closed and disconnected

Keyboard USB disconnected

shut down laptop

1. Submit your completed program (Keyboard.java, Mouse.java, and Main.java) to the myCourses!