**CIS 181 - Lab 9**

**Generic Programming**

## ****Objectives****

* To gain a better understanding of generic programming.
* To gain a better understanding of implementation of generic programming in Java.
* To be familiar with generic class, generic method, and generic interface.

## ****Description****

A generic type is a generic class or interface that is parameterized over types. Generics are a facility of generic programming that were added to the Java programming language in 2004 within version J2SE 5.0. They were designed to extend Java's type system to allow "a type or method to operate on objects of various types while providing compile-time type safety". Generic class, generic method, can generic interface can be described as below.

 A **class** is generic if it declares one or more type variables. These type variables are known as the type parameters of the class. It defines one or more type variables that act as parameters. A generic class declaration defines a set of parameterized types, one for each possible invocation of the type parameter section. All of these parameterized types share the same class at runtime.

 A **method** is generic if it declares one or more type variables. These type variables are known as the formal type parameters of the method. The form of the formal type parameter list is identical to a type parameter list of a class or interface.

 An **interface** is generic if it declares one or more type variables. These type variables are known as the type parameters of the interface. It defines one or more type variables that act as parameters. A generic interface declaration defines a set of types, one for each possible invocation of the type parameter section. All parameterized types share the same interface at runtime.

In this lab, three types of generics are related to the following source files.

1. Generic class:

GenericClass.java

GenericDemo01.java

1. Generic method:

GenericMethod.java

GenericDemo02.java

1. Generic Interface:

GenericInterface.java

GenericInterfaceImplementation1.java

GenericInterfaceImplementation2.java

GenericDemo03.java

You only need to add code in three source files: GenericDemo01.java, GenericDemo02.java, and GenericDemo03.java. Other source files related to generic class, generic method, and generic interface have been completed

**Exercises**

1. Download the following code

* GenericDemo01.java
* GenericDemo02.java
* GenericDemo03.java
* GenericInterface.java
* GenericClass.java
* GenericMethod.java
* GenericInterfaceImplementation1.java
* GenericInterfaceImplementation2.java

1. In Eclipse, create a new Java project called "Lab 9", and import the above 8 files into a “generic” package. Three files GenericDemo01.java, GenericDemo02.java, and GenericDemo03.java. are needed to be modified in your lab.
2. Please complete the missing section of code in the source file GenericDemo01.java. After you complete the programming of this file, compile and run the program of GenericDemo01.java. It should print out the following message:

Can be any type

111

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

11111

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

make sure it is String type

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

Note that missing code is filled out between a pair of comments:

// coding starting here

// coding ending here

1. Please complete the missing section of code in the source file GenericDemo02.java. After you complete the programming of this file, compile and run the program of GenericDemo02.java. It should print out the following message:

11

string type

true

33.3333

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

can be any data type

333

false

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

Note that missing code is filled out between a pair of comments:

// coding starting here

// coding ending here

1. Please complete the missing section of code in the source file GenericDemo03.java. After you complete the programming of this file, compile and run the program of GenericDemo03.java. It should print out the following message:

receive string as a parameter

can only transmit string type here

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

Generic class, accepting any type of data

transmit string

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

transmit integer type

Generic class, accepting any type of data

111

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

Note that missing code is filled out between a pair of comments:

// coding starting here

// coding ending here

1. Submit your completed program (GenericDemo01.java, GenericDemo02.java, and GenericDemo03.java) to the myCourses!