**WXES1115/WXES1117 Data Structures**

**Lab 5: Generics**

In class discussion: Tutorial Q1 & Q2

Q1:

Create a generic class called Container that accepts one parameter, T. Create a no-arg constructor. Declare a private variable, t of type T. Create a method, add that returns nothing, accepting a parameter of generic type. Initialize this parameter to the initially declared variable. Create a generic method called retrieve() that returns the initially declared variable.

Create a main method within the Container class. Create two containers of type Integer and String. Append two objects to the containers, one of type Integer having value 50, another of type String having value Java. Display the Integer and String objects from the respective containers.

Q2 : Create a class called MyArray that has two methods, a main method that creates 3 arrays of

a) integer containing the numbers 1,2,3,4 and 5

b) string containing names, Jane, Tom and Bob

c) character containing alphabet, a, b and c

and a generic method listAll that displays the list of arrays.

1. a) Modify the following program to become a generic class.

public class StorePair {

private int first, second;

public StorePair(int first, int second) {

this.first = first;

this.second = second;

}

public int getFirst() {

return first;

}

public int getSecond() {

return second;

}

public void setPair(int first, int second) {

this.first = first;

this.second = second;

}

public String toString() {

return "first = " + first + " second = " + second;

}

}

b) Create a test program that creates two instances of the StorePair generic class, each having a pair of different Integer values.

c) Override the Object equals() method in the StorePair class.

d) Have the StorePair class implement the Comparable interface. Provide an implementation of compareTo().

e) Invoke the compareTo() that compares the first values of the two Integer instances created in (b) in the test program.

1. Create a generic class called MyGeneric that accepts one parameter. Declare a variable called e for the type parameter. Create a no-arg constructor. Create a constructor that accepts one generic parameter. Create a setter and getter method for the generic type.

Create a test program that creates two instances of generic class of type String called strObj and of type Integer called intObj. Set a value for each of these objects. Display these values using the getter method.

1. In a class called CompareMax, create a generic static method called maximum where the generic type extends the Comparable interface, which receives three parameters. Find the maximum of three values invoked by the main method.
2. Provide a declaration and implementation of the generic method minmax() that takes in an array of generic type and returns a string with the following format: Min = <minValue> Max = <maxValue>. For instance,

Integer[] intArr = {5, 3, 7, 1, 4, 9, 8, 2};

String[] strArr = {“red”, “blue”, “orange”, “tan”};

String intStr = minmax(intArr); //intStr = “Min = 1 Max = 9

String str = minmax(strArr); //str= “Min = blue Max = tan

\*Hint : use Comparable interface to compare the values

1. Create a class called FindMax that contains the following:

Create a Circle class that uses the Comparable interface. Declare the radius variable and a single parameterized constructor that accepts this variable.

In your main program, create 3 different objects of type array for integers that stores the following values, 1,2,3; a list of string that stores red, green, blue and a circle object of radius 3, 2.9 and 5.9. Invoke the max method as below:

public static <E extends Comparable<E>> E max(E[] list)

The max method above returns the maximum value in an array.

1. Write a generic method that returns the minimum element in a two-dimensional array. Below is the method signature:

public static <E extends Comparable<E>> E min(E[][] list)

1. Write the following method that shuffles an ArrayList containing following numbers (14, 24, 4, 42, 5). Below is the method signature:

public static <E> void shuffle(ArrayList<E> list)