**OOP Lab 13**

|  |  |
| --- | --- |
| Due Date: | June 15, 23 : 59 |

* **Submit your assignment using the following file format:**
* LabNumber\_StudentName\_Student\_ID.zip

Example: Lab13\_Hongkildong\_201620505.zip

* The zip file will contain source code file that contains codes of classes.

**Objectives**

* **How to use generic classes and interfaces that implements the basic data structures in computer Science**
* **Understand the advantage and disadvantage of each generic data structure classes**

**Exercises (7x5 =35%)**

**Q1**.Modify lines **16-25 in Fig. 16.3** by using **asList()** method of **Arrays** class . This method converts static array **colors** and **colors2** into generic data structure such as LinkedList**.**  After converting **color** and **color2 arrays,** the LinkedList () constructor uses them as an argument to initialize **list1** and **list2** reference variables.

**Q2**. Write a program that reads in a series of **first names** and eliminates duplicates by storing them in a Set. Allow the user to search for a first name. **The partial code is given in the file “ Q2Code”. Hence, complete the partial code by adding your own code.**

**Q3**. Use a **HashMap** to create a reusable class for choosing one of the **13 pre-defined colors** in a class Color. The names of the colors should be used as keys, and the predefined Color objects should be used as values. Place this class in a package that can be imported into any Java program. User your new class in an application that allows the user to select a color and draw a shape in that color. **The partial code is given in the file “Q3Code”. Hence, complete the partial code by adding your own code.**

**Q4** Write an application to implement the three Set operations: interaction, union, and difference. You should build three methods corresponding to the operations. In your main method, test these methods on two **HashSets** of strings. **The partial code is given in the file”Q4Code”. Hence, complete the partial code by adding your own code.**

**Q5.** Write a program that has a method **ListConcatenate**() which receives two objects of LinkedList, and returns a new LinkedList that contains elements of the first list followed by elements of the second list. In the main method, test this method on two LinkedLists of strings.**The partial code is given in the file “Q5Code”. Hence, complete the partial code by adding your own code.**

**Q6**. Write a program that uses a String method split to tokenize a line of text input by the user and places each token in a TreeSet. Print the elements of the TreeSet. [Note: This should cause the elements to be printed in ascending sorted order). **The partial code is given in the file “Q6Code”. Hence, complete the partial code by adding your own code.**

**Q7**. The output of Fig. **16.15** shows that PriorityQueue orders Double elements in ascending order. Rewrite Fig. 16.15 so that it orders Double elements in descending order (i.e., 9.8 should be the highest-priority element rather than 3.2). **The partial code is given in the file “Q7Code”. Hence, complete the partial code by adding your own code.**