CS1083 Assignment #3 - Winter 2021

Due: Friday, February 5th before 4:00pm (Atlantic), submitted in the Assignment 3 dropbox in Desire2Learn. (Read the submission instructions at the end of this document carefully).

The purpose of this assignment is to provide you with practice with polymorphism

This assignment is to be done individually. If you have questions, direct them to a tutor/assistant during a help session in the "Faculty of Computer Science Student Success Centre" team or to your course instructor.

1. Inheritance and Polymorphism:

Use the given classes (**Shape**, **EquilTriangle**, **Circle**, **Square**) and the driver program **Test1**:

A. Run the program using the given "hard-coded" data. What is the result of using the **equals()** method of the base **Object** class? How is it determining equality? Report the results.

B. Write an **equals()** method for **EquilTriangle**, **Circle** and **Square** that determines equal based on the shape's area (What class should it go in?). Run the **Test1** program again and report the results.

C. Create a **Test2** driver class to hold the following shapes in an array, compare each shape against the others (only once) using your **equals()** method and report the results.

```
new Circle(2);
new Circle(2);
new Circle(3);
new Square(Math.PI);
new EquilTriangle(2);
```

2. Interfaces and Polymorphism:

A. Implement the **Comparable** interface for the various shapes where the **compareTo()** method uses the value of perimeter/area as the basis for comparison. Implement in **Find.java** the following methods using **one of the following two ways** to handle generic types:

1. As in the textbook (see the Sorting class in Chapter 10): **public class Find<T>**; This means the methods will take **Comparable<T>** arrays and array elements may need to be cast to **T** (see examples in Chapter 10). *Note: this approach will generate a warning about "unchecked or unsafe operations" when you compile Find.java, which can be ignored.*

- **`public Comparable getLargest(Comparable<T>[] a);`**
  Returns the largest object in the array
- **`public boolean isPresent(Comparable<T>[] a, T key);`**
  Returns whether a given object (**key**) is present in the array (using **compareTo()**)
- **`public int presentNTimes(Comparable<T>[] a, T key);`**
  A count of the number of occurrences of a given object (**key**) in the array (using **compareTo()**).
- **`public boolean isSorted(Comparable<T>[] a);`**
  Returns whether the array is sorted in ascending order

2. **public class Find<T extends Comparable<T>>**; this means the methods will take arrays with elements of type **T**, there's no need to cast the argument, and the compiler will generate no warning.

- **`public T getLargest(T[] a);`**
  Returns the largest object in the array
- **`public boolean isPresent(T[] a, T key);`**
  Returns whether a given object (**key**) is present in the array (using **compareTo()**)
- **`public int presentNTimes(T[] a, T key);`**
  A count of the number of occurrences of a given object (**key**) in the array (using **compareTo()**).
- **`public boolean isSorted(T[] a);`**

Returns whether the array is sorted in ascending order

Create a driver program **Test3** that uses the same objects as in **Test2** as well as the following object to search and count: `new Square(4);` *Note: you can ignore the warning about "unchecked or unsafe operations" when you compile Find.java.*

B. To illustrate how general the methods in **Find.java** are, create a **Cat** class (instance variables: **name** and **weight**) that implements **Comparable**. Compare cats based on name first, and if the names are the same, compare them using weight. Create a driver program **Test4** that uses two arrays of 3 cats, one sorted in ascending order and one unsorted, as well as a cat to search and count, to test the methods in **Find.java**.

Your electronic submission (submitted via Desire2Learn) will consist of two files:

i. a written report. This should begin with a title page that includes: the course (CS 1083), your section (FR01B, FR02B, FR03B), the assignment number, your full name, and your UNB student number. That should be followed by two sections, with each part clearly identified with a section heading. Include:

   a. The answer to the question in 1A

   b. The modified source code for **Shape**

   c. The **Test2** driver program

   d. The output of **Test2**

   e. Source code for **Find**

   f. The test driver program **Test3**

   g. The output of **Test3**

   h. Source code for **Cat**

   i. The test driver program **Test4**

   j. The output of **Test4**

   This written report should be prepared using a word processor; we recommend using Microsoft Word (i.e. create a .docx file for your report). Copy & paste your java source code & required output into the report document. Add appropriate headings for each part. Fix up the formatting where necessary, adjusting line breaks & page breaks to ensure that your document is easy to read. Use a monospaced font for your code to maintain proper indentation.) Once the report is complete and you've checked it all over, save the .docx file for your own records, and then **save a second copy in pdf format for submission**. (Note: Be sure to open that file in a pdf viewer to verify that the pdf was generated correctly.) **The SINGLE pdf file containing your report will be submitted to the appropriate assignment drop box on Desire2Learn**. (It is important that you submit a pdf file and NOT the original Word document. This pdf will allow the marker to write comments directly on your work to give you better feedback.)

Note: name this report as follows: **YourName_CS1083_As3_Report.pdf**

ii. an archive file (**.zip**) that contains your Java source code and input files for this assignment. Make sure that your archive includes all .java files (in case the marker wishes to compile & run your code to test it). You should not include the report document or the .class files in your archive. This archive should be submitted as a **single file** to the appropriate drop box on Desire2Learn.

Note: name this archive file as follows: **YourName_CS1083_As3_Archive.zip**