CORNERSTONE
International Community College of Canada

Module 2
Java Development

Assignment 7- MADP202- July11, 2019

Due: Friday July 19, 6:30 pm:

Submission: Please create a corresponding pull-request to the correct Github branch

Late submission Penalty Policy:

Lack of submission of 3 or more assignments: 50% of your entire assignments. (7.5% of your course mark).
Lack of submission of 5 or more assignments: 100% of your entire assignments (15% of your course mark).
Remember you need to get at least 70% to pass the course.

Topics:

Generic Types in Java

Problem 1:

Write a generic method to count the number of elements in a collection that have a specific property (for example, Students with GPA above average, Books with specific categories). You need to:

- Define and implement a generic class with a generic method called: *countNumberOfElementsWithSpecificProperty*
- Design and implement a POJO class an element like Book, Student, Building, Account and define 3 properties for this class.
- Create an ApplicationDriver class (with a main method) and test your implementation.
- Note: You might need to create classes or interfaces in addition to the three classes mentioned above.

What is the time and space complexity order of your solution?

Problem 2:

Will the following class compile? If not, why?

```
1.  public final class Algorithm {
2.      public static <T> T max(T x, T y) {
3.          return x > y ? x : y;
4.      }
5.  }
```

What is the time and space complexity order of this method?

Problem 3:

Write a generic method to exchange the positions of two different elements in an array.

- Follow the same procedure as described in problem one. Just name the generic method *exchangeElements*

Problem 4:

Write a generic class with the following three methods and properties:

- The class has a generic property of type List.
- *addItemToList*: add a generic object to the list
- *removeItemFromTheList:* remove a generic object from the list
- *performOperation*: Performs an operation on the list and returns a list of all items which satisfies a specific search criteria. If no item is found return nill.

Note: Follow the same steps as defined in problem one.

Problem 5:

How do you invoke the following method to find the first integer in a list that is relatively prime to a list of specified integers?

```
1.  public static <T>
2.      int findFirst(List<T> list, int begin, int end, UnaryPredicate<T> p)
```

Note that two integers *a* and *b* are relatively prime if gcd(*a, b*) = 1, where gcd is short for greatest common divisor.