

Your Name: _____ Student Number: _____

Lab 8a: to be done in class with a partner – due at the end of this lesson (10 marks)

ProvinceTerritory

Create a class called ProvinceTerritory. It contains two instance variables: a String name and an int population. Create accessors and mutators too. The constructor takes the name and population as parameters and sets them as long as the name isn't null and the population isn't negative.

Canada

Create a class called Canada. It contains an array of 13 ProvinceTerritory references:

	Name	Population
	Ontario	12,851,821
	Quebec	7,903,001
	British Columbia	4,400,057
	Alberta	3,645,257
	Manitoba	1,208,268
	Saskatchewan	1,033,381
	Nova Scotia	921,727
	New Brunswick	751,171
	Newfoundland and Labrador	514,536
	Prince Edward Island	140,204
	Northwest Territories	41,462
	Yukon	33,897
	Nunavut	31,906
	Canada	33,476,688

Populate the array inside the Canada constructor. The following code will be useful:


```
public static final int BC = 0;
public static final int AB = 1;
etc...
```

```
provinces[BC] = new ProvinceTerritory("british columbia", 4400057);
provinces[AB] = new ProvinceTerritory("alberta", 3645257);
etc...
```

Create a method called **public int getTotalPopulation()** which uses a loop to add up all of the individual populations to calculate Canada's total population.

Create a method called **public String getLowestPopulation()** which uses a loop to determine and return the name of the province/territory that has the lowest population.

Create a method called **public int getPopulation(String province)** which returns the population of the province (the parameter); if there is no such province, return a constant called `NO_SUCH_PROVINCE`, which is an int set to -1. If the parameter is null or empty, throw an `IllegalArgumentException`.

NOTE: every method is responsible for null checks: is the array null? Is the Province null? Is the name null? Check before calling any properties (e.g. `.length` or `.toUpperCase()`).

Demonstrate your completed project to your instructor. When your instructor is satisfied, your paper will be signed and you can go home. Lab 8b (below) is due at the next lesson, and there is also a quiz next day.

Checked by: _____

NOTE: keep this paper for your instructor to verify your grades later in the course.

NOTE: EVERY SINGLE STUDENT MUST SUBMIT THIS LAB AND GET HIS OR HER PAPER SIGNED....

Lab 8b: at home, alone – due Thursday, Nov. 8, 2018 at midnight (15 marks)

Continue with the Canada class you made in lab 8a, above.

Here is an example method called **public String[] getProvincesWithPopulationBetween(int min, int max)** which returns an array of the names of all provinces/territories that have a population between min and max (inclusive):

```
public String[] getProvincesWithPopulationBetween(int min, int max){
    int i = 0;
    int j = 0; // the new array index
    int numOfProvWithPop = 0;
    String[] matchingProvinces;

    while(i < provinces.length){
        if((provinces[i].getPopulation() >= min) &&
            (provinces[i].getPopulation() <= max)){
            numOfProvWithPop++;
        }
        i++;
    }

    if(numOfProvWithPop > 0){
        matchingProvinces = new ProvinceTerritory[numOfProvWithPop];
    }else{
        // no matches
        return null;
    }

    i = 0; // start looking from the beginning again
    while(i < provinces.length){
        if((provinces[i].getPopulation() >= min) &&
            provinces[i].getPopulation() <= max){
            matchingProvinces[j] = provinces[i];
            j++;
        }
        i++;
    }
    return matchingProvinces;
}
```


Create a method called **public boolean isProvinceInCanada(String name)** which returns true if there is a province/territory in Canada with the given name (the parameter); otherwise returns false. If the parameter is null or empty, throw an `IllegalArgumentException`.

Create a method called **public String[] getProvincesWhoseNameContains(String substring)** which returns an array of the names of all provinces/territories whose name contains substring (the parameter). Hint: use the `String` class's **contains()** method. If the parameter is null or empty, throw an `IllegalArgumentException`.

Create a method called

public ProvinceTerritory[] getMoreProvincesWhoseNameContains(String substring)

which returns an array of the names of all provinces/territories whose name contains substring (the parameter). Hint: use the `String` class's **contains()** method. If the parameter is null or empty, throw an `IllegalArgumentException`.

Create a method called **public String[] getProvincesWhoseNameStartsWith(char letter)** which returns an array of the names of all provinces/territories whose name starts with letter (the parameter). Hint: use the `String` class's **startsWith()** method or use the `String` class's **charAt(0)** method. If the parameter is null or empty, throw an `IllegalArgumentException`.

NOTE: every method is responsible for null checks: is the array null? Is the Province null? Is the name null? Check before calling any properties (e.g. `.length` or `.toUpperCase()`).

Submission (15 marks)

Test your code and submit it to the Lab 8 folder on D2L (Activities -> Assignments -> Lab8) by **Thursday, Nov. 8, 2018 at midnight**. The submission must include the **ProvinceTerritory.java** and **Canada.java** files from your BlueJ project.

Mark Breakdown:

- 0 points if the `ProvinceTerritory` or `Canada` classes do not compile

- (3 points) Valid JavaDoc (class and constructor), good names, data types, constants (i.e., no magic numbers)
 - 1 point taken off for each violation
- (12 points) Meets the specifications in the description and returns correct results for each method
 - Your class will be evaluated against a set of tests to make sure the methods return the correct results for a given set of inputs

Province/Territory

Create a class called `ProvinceTerritory`. It contains two instance variables: a `String` name and an `int` population. Create accessors and mutators for the constructor takes the name and population as parameters and sets them, as long as the name isn't null and the population isn't negative.

Canada

Create a class called `Canada`. It contains an array of 13 `ProvinceTerritory` references.

Name	Population
Ontario	12,851,821
Quebec	7,923,241
British Columbia	4,400,057
Alberta	3,645,257
Manitoba	1,206,268
Saskatchewan	1,073,330
Northwest Territories	92,722
Yukon	34,131
New Brunswick and Labrador	1,145,335
Prince Edward Island	141,214
Atlantic Provinces	1,145,335
Yukon	34,131
Nunavut	34,131
Chad	12,851,821

Populate the array inside the `Canada` constructor. The following code will be useful: