IB Computer Science

Student_____

Case Study: Populations

Date: _____

Objectives:

- Construct and implement algorithms utilizing 2-dimensional arrays
- Define and use constants as a programming construct
- Analyse the use of variables, constants and operators in algorithms.

Overview:

In this programming lab assignment you will review the Population Study Problem, review the Java program code that has been provided to you, answer questions about your understanding and interpretation of the information provided to you, and complete some Java programming tasks. Read the directions of each section carefully.

The Population Study Problem

A population study divides a metropolitan area into seven regions: A–G.

The following table shows the current population (in millions) of the regions.

Region	Current population (millions)
A	2.3
B	2.1
C	1.2
D	1.4
E	1.5
F	1.1
G	0.8

Two one-dimensional arrays, `Region` and `Curr_Pop`, are used to hold this data.
For example, `Region[0] ='A'`. The population in region A is 2.3 million and 2.3 is found in `Curr_Pop[0]`.

IB Computer Science

Student_____

Case Study: Populations

Date: _____

Part I:

Directions: Java modules have been created for you. Set up a programming environment and review the programs as specified:

1) Setting up Cloud 9
    a. In your Cloud9 firstname_java work space create a directory named population.
    b. Open a terminal session and make the population directory your current directory.
    c. Start an FTP session to download the required files. A userid and password will not be needed. The first letter below is a lower case L

```
lftp ftp.hwmath.net
```

    d. Issue the following ftp commands, one at a time:

```
get Population.java
get Trace.class
exit
```

    e. Verify that you have both the Population.java and Trace.class files in your population directory.
2) Program Review #1
    a. Review the way final variables are declared and used in the Population.java program. How many final variables are declared? _____
    b. The array declarations were integrated from Noah's program. Review how the one-dimension arrays Region and Curr_Pop are declared and initialized. Complare this to the population chart on page 1.
    c. Review the method **print_population()** and trace the output that would be displayed based on how the Region and Curr_Pop arrays are initialized:

IB Computer Science

Student_____

Case Study: Populations

Date: _____

The numbers in the following table represent expected **percentages** of yearly migration from one region to another, obtained by analysing historical migration data. For example, it is expected that 0.32% of the current population of region B will move to region C.

The diagonal entries represent a region's internal growth rate. For example, the population of region C is expected to increase by 1.2% as a result of the births and deaths of people currently living in region C.

To / From	A	B	C	D	E	F	G
A	**1.10**	0.21	0.21	0.05	0.20	0.20	0.29
B	0.30	**1.20**	0.32	0.25	0.20	0.09	0.31
C	0.25	0.22	**1.20**	0.35	0.30	0.23	0.12
D	0.10	0.33	0.36	**1.30**	0.09	0.12	0.20
E	0.20	0.22	0.24	0.35	**1.00**	0.20	0.21
F	0.12	0.21	0.13	0.21	0.22	**1.40**	0.31
G	0.05	0.03	0.30	0.20	0.23	0.26	**0.90**

3) Program Review #2

a. Review the declaration and initialization of the 2-dimensional array migration.
b. Notice in particular how neatly Noah lines up the rows and columns to match the table above.

4) Using the tables above and the declarations in Population.java, answer each of the following questions. You can use a calculator but do not compile or run any programs to answer these questions.

a. State the **percentage** of the population of region G that are expected to move to Region A _____
b. Determine the number of people from region B who are expected to move to region E. _____

4.c Describe how the change in population of region F in one year could be determined (There are three components that are needed to consider]

1.

2.

3.

IB Computer Science

Student_____

Case Study: Populations

Date: _____

The Trace Module

You have been provided with the Trace.class file which is used for tracing and debugging a program. A trace level can be set using the calls to set_trace_off(), set_trace_low(), set_trace_medium(), and set_trace_high().

Messages sent using the method trace(String s) will be displayed as long as the trace level is not set to off.

Messages send using the method trace(int level, String s) will be displayed as long as the trace level specified by the level parameter is less than or equal to the current trace level. This is helpful when you want some messages to be displayed, but not all messages.

Part II Additional Program Review

5. Tracing the main program
   a. See how the Trace t object is declared at the top of Population.java and initialized in the main method. Notice also that the main method is the last method defined in the Population.java file.
   b. The calls to print_population() and print_totalPopulation() should produce the output you should expect after completing the previous lab steps and answering the questions.
   c. Notice that the main method calls methods question_4a(), question_4b(), and question_4c(). Review the questions asked in 4.a, 4.b, and 4.c and then look at how those questions are answered in each of the corresponding methods.
   d. Complie Population.java and execute it – compare the output produced with the answers that you provided? Explain any differences in the space below:

IB Computer Science

Student_____

Case Study: Populations

Date: _____

Part III Algorithm and Programming Assignment

Review Question 4 questions, the java code provided, and the program output carefully.

Construct the algorithm that will predict the population in each region after 10 years. You should assume that the yearly migration percentages given in the table remain the same over the 10 years.

Complete the code in the main method and **implement your algorithm** in method **`calc_migration().`**

Utilize the `Trace` class as needed to help verify the correctness of your algorithm and your code.

Complete the following table with the populations of each region after 10 years:

Region	Population after 10 years (millions)
A	
B	
C	
D	
E	
F	
G	