Section: A Name: Colin Morris-Moncada

**EGR222 HW3 Worksheet**

Do this worksheet BEFORE YOU IMPLEMENT anything in IntelliJ. This worksheet is designed to give you opportunity how you want to design and structure your program before you jump into implementation. This will help you to come up with better cleaner solution with the 7 goals we talked about during class.

This worksheet focuses on high-level structure rather than some specific functionalities (such as handling capitalization, specific height of graphics, etc.)

1. What are the 7 goals that you want to achieve in all of your programs?

1) Functionality

2) NO Redundancy

3) Efficiency (performance)

4) NO Method Chaining

5) Readability

6) Good Style

7) Single Responsibility

1. Read HW3 spec carefully first before moving on to any of below questions.
2. Think how many class constants you will hire. Note that there are at least 3 required class constants you need to have.   
   (See Stylistic Guidelines in page 4)

1. What should be the 3 modifiers for class constant?

\_\_final\_\_ \_\_\_final\_\_\_\_\_\_\_ \_\_\_\_\_final\_\_\_\_\_\_

1. Required class constants

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Name | Purpose | Default Value |
| Int | STARTING\_YEAR | Start year of data | 1890 |
| Int | DECADE\_WIDTH | Width of decade on panel | 60 |
| Int | LEGEND\_HEIGHT | Height of legend rectangles | 30 |

1. What are the 3 major tasks that your program needs to do?
2. Prompt the user and get \_\_\_\_\_**name\_\_\_\_**and \_\_\_\_\_\_**gender**\_\_\_\_\_\_\_ from the user.
3. If exists, output to console \_\_r**anking**\_\_\_ and \_\_**meaning**\_\_ of name and gender requested.

If not, output to console \_\_\_\_\_**” ”**\_**not found**\_.

1. If 2) exists, also output graphically to Drawing Panel the same info as 2).

Think about for above 4-1, 4-2, 4-3 where you would place your code.

Should it go inside main method, or should you separate out as a method?

1. Let’s start with main.
2. What is the requirement says for the main method? (See Stylistic Guidelines)
   1. **Main should not draw on a drawing panel not read lines of input from a file.**
3. Then what should be your main method look like? Write down pseudo-code.

public static void main(String[] args){

printMessage(MESSAGE\_PREFIX);

userInput();

outputToConsole();

outputToGraphics();

}

1. Write down a few methods that you are going to call in main method.

(Note, that you don’t want to call too many methods in main. Just like HW1 example, just cone(); body(); code() was enough and all the details can be handled by cone() and body() method individually.

For each method, put a header info such as

*public static void printMessage(String msg) {}*

public static void printMessage (String msg){}

public static void userInput (){}

public static void outputToConsole (){}

public static void outputToGraphics(){}

1. See second paragraph in page 3 of the HW3 spec above Graphical Output.

In this HW3, there are two types of input one is names.txt and the other is meaning.txt. Take a look at their format.

1. What is common in these files?

Each has multiple lines of content. Each line represents one name with corresponding traits such as popularity or interesting facts about that name. Each trait is separated by whitespace.

1. What should the method for searching for both files do? What should this method return? Will returning boolean better? int? Something else?

The method should use a while loop to make sure there is still lines to be read from the input file.

Put a header info for this method and pseudo code here.

*public static return\_type yourMethodName(Your parameters … ) {*

*…*

*Write your pseudo code here*

*}*

public static void

1. See page 3, Graphical Output requirement. And also the last bullet point in Stylistic Guidelines on page 4.

Your main method should call 1 graphical method, according to 4-3.

What are the four things this graphical method should output?

1. Panel with size \_\_\_\_\_\_\_\_\_\_ EQUATION GOES HERE \_\_\_\_\_\_\_\_\_\_\_ (It will be better not to use hard-coded value) with white background
2. Top gray legend box
   1. fixed
3. Bottom gray legend box
   1. fixed
4. Top black line
   1. fixed
5. Bottom black line
   1. fixed
6. Green bars
   1. dynamic
7. Year string on top of the bottom gray legend box
   1. dynamic
8. Rank string on top of the green bar
   1. dynamic
9. Among 7) which are fixed and which are not?
10. Design methods and method calls to handle graphics output and write header and pseudo code for each method