

CIS 36A – Java Programming I

Lab 2 & 3: Greeter

All classes must follow good design and coding practices, including code indenting, and appropriate comments.

Introduction

In this lab, you'll use the pre-written class definition of a class called **Greeter**. **Greeter** is a fully-functional class definition which does nothing except contain a *main* method!

Don't worry about fully understanding classes and class definitions yet, since that will be covered in chapter 4. You will simply write your Java code inside the *main* method of **Greeter**, to accomplish the lab's requirements, and then export and submit a .JAR file with sources, as always.

Requirements

1. Your project, when run, will ask the user three questions, by printing them to the screen. After each question, your project will take keyboard input from the user, using an instance of **Scanner**. Each response given by the user should be saved to a variable of an appropriate type!

Make sure that your instance of **Scanner** uses the appropriate method! For example, if you expect a value of type *double* from the user, use the instance method *getDouble()*.

Here are the three questions, each followed by the type of variable you expect to receive from the user's keyboard response:

1. What is your name? (**String**)
2. How old are you? (*int*)
3. What is the diagonal length, in cm, of the screen you're reading this on? (*double*)

2. Your project will then use the responses you took from the user (and saved into variables), and print four lines of text to the user. Here are the three lines of text you'll print:

1. "Hi, <NAME>", where <NAME> is what the user gave for their name.
If the user types your name, print an additional line:
"What a great name!"
2. If the user is five years old or less, print:
"Wow, you're young!"
If the user is older than or equal to one hundred, print:
"Wow, you're old!"
3. "Triple that diagonal is <TRIPLEDIAG>!", where <TRIPLEDIAG> is the value the user entered multiplied by three.

Example Output

Here are two examples of what the screen might look like after your program is run. For clarity, I have put the user input in red.

Example output 1, in which user enters a name that is not the same name as the programmer:

```
What is your name? Foo
How old are you? 110
What is the diagonal length, in cm, of the screen you're reading
this on? 287.42
Hi, Foo!
Wow, you're old!
Triple that diagonal is 862.26!
```

Example output 2, in which user enters a name that is the same name as the programmer:

```
What is your name? Michael
How old are you? 50
What is the diagonal length, in cm, of the screen you're reading
this on? 1.0
Hi, Michael!
What a great name!
Triple that diagonal is 3.0!
```