|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LEARNING PROFILE FOR Interest | | | | | |
| *Name* | *:* | *Tyler Lucas* | *Due Date* | *:* | *N/A* |
| *Student ID* | *:* | *3305203* | *Submission Date* | *:* | *N/A* |

**1. Problem Statement:**

Textbook example program.

**2. Description of the Code:**

Calculates interest earned on investment given annual interest rate, compound type (periods per year; e.g. 12 for monthly), and accrual period in years.

**3. Errors and Warnings:**

Table : List of Errors and Warnings Encountered in the Program

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Errors / Warnings** | **Details** | **How I solved them** |
| 1 | E1 | method exp in class Math cannot be applied to given types;  required: double  found: double, double  reason: actual and formal argument lists differ in length  I mixed up Math.exp() and Math.pow(). | Changed Math.exp() to Math.pow(). |
| 2 |  |  |  |

**4. Sample Input and Output:**

description

[Provide some test cases with sample input and output of your program.]

**5. Discussion:**

The first error, where a class couldn’t be found in the project, was caused by setting either or both the class and the main method to “private”. I first ran into this error when attempting the *HelloWorld* sample program[[1]](#footnote-1). Having read the class Style Guide in which it says “Create private fields with getters/setters rather than leaving fields public,” as well as Controlling Access to Members of a Class[[2]](#footnote-2) in which it says “Use private unless you have a good reason not to,” I mistakenly thought this applied to the main class and method as well, as I’m still not sure what the differences are between a class, method, and object. Searching online didn’t reveal a solution right away, as few experienced programmers would think that something this simple could go awry, but I eventually found the answer here: <https://goo.gl/P2OdMJ>. Of course, the next page in the textbook had the answer as well:

The word “public” in the first line of main() means that this routine can be called from outside the program. This is essential because the main() routine is called by the Java interpreter, which is something external to the program itself.[[3]](#footnote-3)

I’ll have to keep an eye out to see how to implement other access levels at these top levels.

1. (Eck, 2014, p. 21) [↑](#footnote-ref-1)
2. (Oracle, 2015) [↑](#footnote-ref-2)
3. (Eck, 2014, p. 22) [↑](#footnote-ref-3)