**CptS 121 - Program Design and Development**

**Programming Assignment 1: Equation Evaluator**

**Assigned:** Friday, January 17, 2025

**Due:** Friday, January 24, 2025 by midnight

**I. Learner Objectives:**

At the conclusion of this programming assignment, participants should be able to:

* Analyze a basic set of requirements for a problem and derive logical solutions to them
* Declare variables
* Apply C data types and associated mathematical operators
* Comment a program according to class standards
* Logically order sequential C statements to solve small problems
* Compose a small C language program
* Compile a C program using Microsoft Visual Studio 2022
* Execute a program
* Create basic test cases for a program

**II. Prerequisites:**

Before starting this programming assignment, participants should be able to:

* Access Microsoft Visual Studio 2022 Integrated Development Environment (IDE)
* Summarize topics from Hanly & Koffman Chapters 1 – 2 including:
	+ The steps of the software development method
	+ C language elements (preprocessor directives, reserved words, and standard identifiers)
	+ The standard C data types
	+ The general form of a high-level program

**III. Overview & Requirements:**

Write a C program that evaluates the equations provided below. The program must prompt the user for inputs to each equation and evaluate them based on the inputs. All equations should be placed into a single .c file. This means you should ***NOT*** have 7 Visual Studio projects or 7 .c files. All variables, except for the *plaintext\_character, encoded\_character* and variable *a*, are floating-point values. The *plaintext\_character* and *encoded\_character* variables are characters and the *a* variable is an integer. Note: the atomic mass of phosphate equation does not require any user input. The *PI* and *mass constants* used in the atomic mass calculation must be defined as constant macros (#defined constants). Error checking is not required for your program. You do not need to check for faulty user input or ***dividing by zero***.

1. Newton’s Second Law of Motion: force = mass \* acceleration
2. Volume of a cylinder: volume\_cylinder = PI \* radius2 \* height
3. Character encoding: encoded\_character = (plaintext\_character – ‘7’) + ‘A’
4. Atomic mass of phosphate: phosphate\_atomic\_mass = 3 \* hydrogen\_mass + 4 \* oxygen\_mass + phosphorous\_mass, where hydrogen\_mass = 1.008 amu / atom, oxygen\_mass = 16 amu / atom, and phosphorous\_mass = 30.97 amu / atom
5. Height of projectile in meters: height = -16 \* time2 + initial\_velocity \* time + initial\_height
6. Current through circuit in Amperes: current = square root of (power / resistance)
7. General equation: y = (3 / 4) \* x – z / (a % 2) + PI (recall: *a* is an integer; note: programmatically explicitly typecast the 3 / 4 quantity to a floating-point number to achieve 0.75)

**IV. Expected Results:**

The following console window illustrates inputs and outputs that are appropriate for your program. Your program must display the results in a similar form as shown in the window. The window shows possible results for the given input tests, for the first two equations only.



Note: you will need to display the results for all of the equations!

**V. Submitting Assignments:**

1. Using Canvas <https://canvas.wsu.edu/>, please submit your solution to the correct “Programming Assignments” (PA) folder. Your solution should be zipped into a .zip file with the name <your last name>\_PA1.zip and uploaded. To upload your solution, please navigate to your correct Canvas ***lab*** course space. Select the “Assignments” link in the main left menu bar. Navigate to the correct PA submission folder. Click the “Start Assignment” button. Click the “Upload File” button. Choose the appropriate .zip file with your solution. Finally, click the “Submit Assignment” button.
2. Your project should contain your C source file (which must be a .c file).
3. Your project must build properly. The most points an assignment can receive if it does not build properly is 65 out of 100.

**VI. Grading Guidelines:**

This assignment is worth 100 points. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

* 5 pts for correct declaration of constant macros
* 35 pts for proper prompts and handling of input (5 pts/equation)
* 49 pts for correct calculation of results based on given inputs (7 pts/equation)
* 11 pts for adherence to proper programming style established for the class and comments