Jasque Saydyk and Michael Horton

Professor Vanderberg, Swenson, and Trice

CS 126L Section 2

7 February 2017

**Lab 03 - Math Quiz**

## 1. Problem Statement

This program is a simple math quiz where the user can choose between one of three levels of difficulty, then the program tells them how well they performed. The user should be able to put in any input without paying attention to case-sensitivity. The program should inform the user of any errors they've inputted, and allow the user to continue or quit if they choose to.

Requirements

- User can choose how many questions they will be asked
- Three distinct levels for the user to choose
    - Beginner - addition or subtraction, between 1 to 10, random
    - Intermediate - addition, subtraction, multiplication, or division, between 1 to 25, random
    - Advanced - one of five questions from prelab, with random range of numbers that makes sense
- End conditions
    - Well done! - if user answered more than ⅔ of the questions correctly
    - You need more practice - if user got ⅔ questions correct
    - Please ask your math teacher for help! - if less than ⅓ questions correct
- All inputs should work regardless of case used

Additional Requirements

- Allow the user to restart or end the game when it ends
- Notify the user with an error if they give the game invalid input
- Allow the user to reenter the input if they get an error

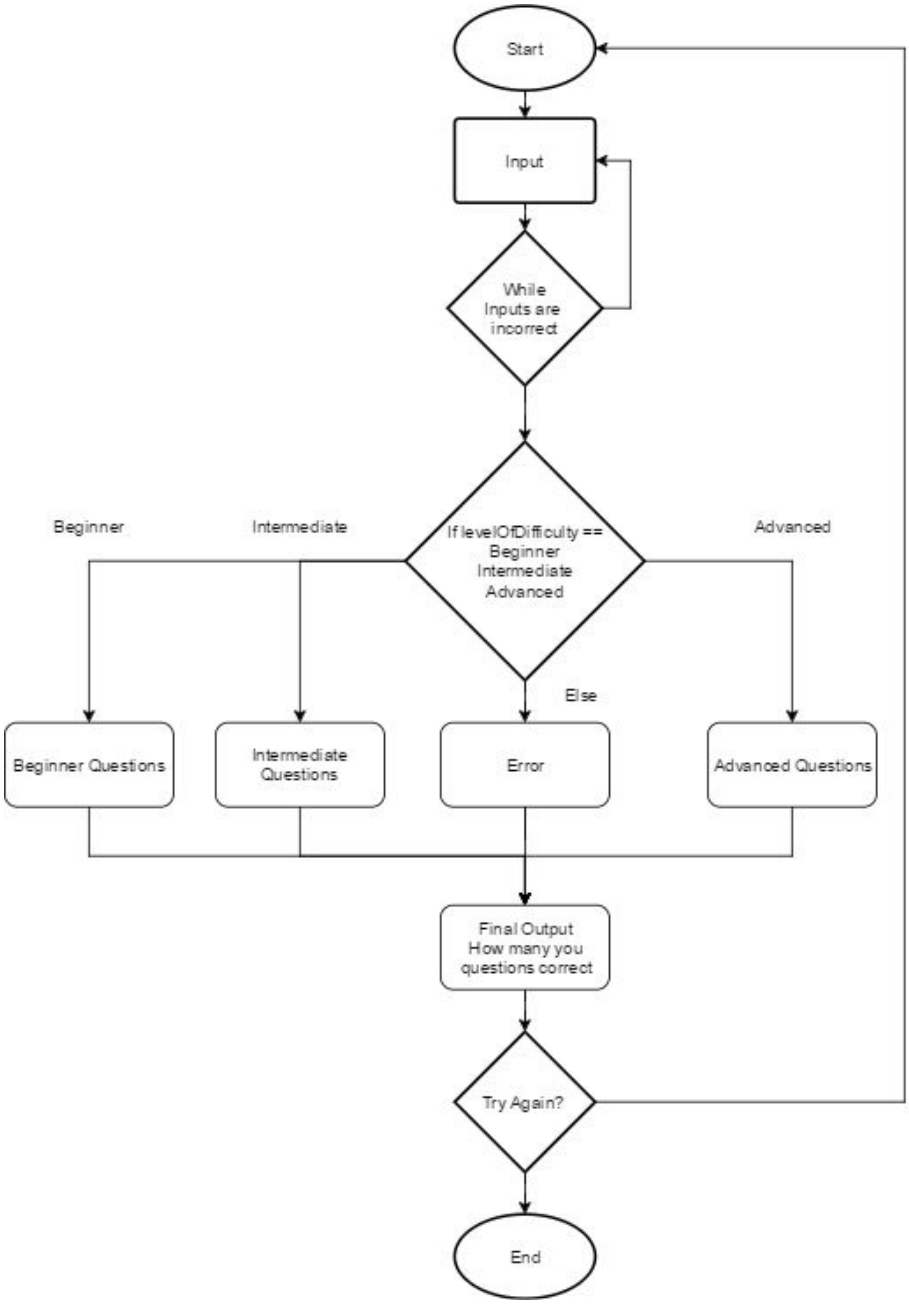
## 2. Planning

The first thing to point out is that since we are going to allow the user to restart the game when it ends and give them an error when they give an invalid input, this is going to require the game to operate from within a while loop. The various difficulties of the game will take place in various nested if trees.

Due to the additional requirements added onto this program, the best way to implement it is to have a separate class for each of the difficulties and have the advanced questions potentially in a separate text file. This would greatly reduce the main file's length and make it significantly more readable. However, we will not implement the project this way due to some constraints on knowledge in our team.

The alternative implementation is to have a series of nested while loops, for loops, and if trees in one file, which is what we will be doing.

Below is a rough, general flow diagram of the project



### 3. Implementation and Testing

Implementation was straight-forward. The only nuance I had to add that differed from the planning stages is that only the initial input of the number of questions the user wanted and the difficulty of the questions had to be in a while loop, along with the try again prompt for there to be adequate error checking. Also, any int cast is wrapped in try catch blocks to catch any ValueErrors if the user inputted a string instead.

I also moved the paragraphs for the advanced questions to the top of the program as to not clutter the already cluttered while loops and if trees.

```
How many question's would you like to answer? 2

What difficulty do you want? (Beginner, Intermediate, Advanced) beginner

What's 9 minus 7? 3

No, I'm afraid the answer is 2.

What's 5 plus 10? 15

That's right ----- well done.

I asked you 2 questions. You go 1 of them right.
You need more practice

Try Again?(Y/N) y

How many question's would you like to answer? 1

What difficulty do you want? (Beginner, Intermediate, Advanced) advanced

The Persians have three archers for every calvary, whereas the Macedonians
have three cavalry for every five hoplites they have. The Persians have 22187 calvary,
whereas the Macedonians have 24617 Hoplites. Who has the larger army?
(Enter 0 for the Persians, enter 1 for the Macedonians): 0

No, I'm afraid the answer is 1.
Please ask your math teacher for help!

Try Again?(Y/N) n
>>>
```

### 4. Reflection

For this problem we were given and it's intended size, we believe we arrived at the most appropriate solutions. However it must be noted that this is not the best way to implement the project, and that if this project were to be expanded in scope in any way, an object oriented approach would be a vastly superior solution.