Short Description

You are working on building a system to analyze the grades of university students. To do this, you have to build many small and inter-related programs that will help solve different problems in your system. Good luck!

Learning Outcomes

- Practice basic programming skills
- Discover new language features[1]
- Practice using conditional statements (*if*, *elif*, *else*)
- Understand how nesting of conditional statements impact the functionality of the program

Detailed Explanation and Things To Do

- Design and write code for each question in this lab exercise without using any built-in functions except print, input, and type conversion functions, if needed.
- You are **strongly encouraged** to complete questions 1 - 3 for extra practice and show them to your TAs during the lab session. However, you only need to **submit** the solution to question 4 .
- Your code must implement the major logical tasks, such as interacting with the user through input statements and displaying the relevant output.
- Make sure you are following the applicable code quality standards outlined in the Software Quality Tests. (Ignore the requirements relating to functions, repetition statements, and having a main function - none of those are needed in this exercise.)

Resources

- None

[1] To create this program, you may need to use a few programming language features that have not been used yet in class. Discovering new language features and how to use them is an integral part of problem-solving in computing science and an essential skill that you should learn. Think about what you need to do, search the web for python3 programming examples, and/or use the Python documentation to help you find the programming constructs that you need. If you get stuck, ask your TA for help/hints about the programming constructs you need to use.

Question 1

Your first task is simple. Your system should show if two students’ grades are equal. Write a program that takes two grades as inputs (floating point numbers), checks whether they are equal or not, and prints a corresponding message. For this question, you may assume that the user always enters a valid floating-point number between 0 and 4 (both endpoints inclusive). The output should be displayed according to the sample output below:

Sample Run 1:

```
What is the first grade? 3.5
What is the second grade? 2.5
3.5 and 2.5 are NOT equal!
```

Sample Run 2:

```
What is the first grade? 4
What is the second grade? 4
4 and 4 are equal!
```

Question 2

Given three grades (assuming all grades are floating-point numbers), your program must print the highest value of these grades. Write a program to do this according to the formatting shown in the sample output below. For this question, you may assume that the user always enters a valid floating-point number between 0 and 4 (both endpoints inclusive). You must set up your program such that it uses a conditional statement to compare the values given by the user (this lab is about practicing conditional statements).

Sample Run 1:

```
What is the first grade? 2.3
What is the second grade? 3.1
What is the third grade? 4.0
The highest grade is 4.0
```

Sample Run 2:

```
What is the first grade? 3.1
What is the second grade? 3.5
What is the third grade? 2.3
```

```
The highest grade is 3.5
```

Question 3

The system should analyze a given grade and describe it based on the following categorization: when the grade is less than 1, it is categorized as “Failure”. When it is below 1.3 but greater than or equal to 1, it is categorized as “Poor”. When it is below 2.3 but greater than or equal to 1.3, it is categorized as ”Satisfactory”. When it is below 3.3 but greater than or equal to 2.3, it is categorized as ”Good”. Finally, if the grade is greater than or equal to 3.3 it is categorized as ”Excellent”. If the grade is not in the valid range of 0 to 4.0, the program must categorize it as “Invalid”.
Write a program that takes a floating-point number grade as input and prints its category. In addition to this functionality, your code must be written in a specific way to help you gain a deeper understanding of conditional statements: in your program, you must not express ranges of grades explicitly using logical operators (ex. if grade> 1.3 and grade< 2.3); instead, you must have at most one comparison operation in each branch of conditional statements. In other words, only one comparison operation is allowed after an *if* (ex. if grade>= 1.3) or *elif* keyword (ex. elif grade< 2.3). You may assume that the user always enters a valid floating-point number between 0 and 4.

Sample Run 1:

```
Input the grade: 1.2
The grade is Poor!
```

Sample Run 2:

```
Input the grade: 3.2
The grade is Good!
```

Sample Run 3:

```
Input the grade: 4.0
The grade is Excellent!
```

Question 4

In a self-paced online course, students are assigned to different levels in the course based on the marks they receive on two quizzes that they take at the start of the course. The three levels are :
- Level 1 (Easy)

- Level 2 (Medium)
- Level 3 (Hard)

If a student is able to score greater than or equal to 80 in both quizzes, they are assigned to Level 3.

If a student is able to score greater than or equal to 50 in both quizzes they are assigned to Level 2.

If a student scores less than 50 in one of the two quizzes and greater than or equal to 50 in the other quiz, the student is asked to redo the quiz they scored less on.

If a student scores less than 50 in both quizzes, the student starts the course at level 1.

You may assume the user always enters a valid float value in the range of 0 and 100 (both endpoints inclusive).

Sample Run 1:

```
Enter marks for quiz 1 >90
Enter marks for quiz 2 >90
Level 3
```

Sample Run 2:

```
Enter marks for quiz 1 >60
Enter marks for quiz 2 >70
Level 2
```

Sample Run 3:

```
Enter marks for quiz 1 >40
Enter marks for quiz 2 >60
Redo quiz1
```

Sample Run 4:

```
Enter marks for quiz 1 >45
Enter marks for quiz 2 >30
Level 1
```

Sample Run 5:

```
Enter marks for quiz 1 >70
Enter marks for quiz 2 >47
Redo quiz2
```

Submission Information

- You are required to submit the solution for **question 4** of this lab exercise by the due date provided. For submission purposes, the file with your code should be named:

  ```
  level.py
  ```