<https://github.com/ChrisPuzzo/COMP830IndProject>

The following is the class diagram for my implementation of the game code:

A screenshot of a computer game

Description automatically generated

To start, the questionDict dictionary is used to hold all the questions. It starts with a key, generally numbered 1-15 with two special keys called “done” and “wrong”. Each key-value pair consists of a key and a list that has 3 values: the question, the next question if the answer is yes, the next question if the answer is no. Done and Wrong are the final two questions, where the code either guesses correctly or incorrectly and then asks if the user wants to finish or restart the game.

The code only uses one class because that was all that it needed in my opinion. Having a class for each part of it would add a lot of unnecessary complexity to it.

In the class there are 5 methods. The first one that I am going to talk about is the showQuestion class. The show question class does what it says, it shows what the question is. It displays the question and then takes user input. Depending on what the user answers it either redisplays the question (if they choose something besides (1) or (2)), calls the answerYes method, or calls the answerNo method.

Both the answerYes and answerNo methods do basically the same thing, except one is used for no answers and the other is used for yes answers. While it might not have been necessary to have both it helped make the code more readable in my opinion. What they do is take the key of the next questions and re-call the showQuestion method again but this time with the new question.

stopGame is a pretty simple method in that it is used for one thing, stopping the game using the exit() method.

gameStart is a method mean to be called to start the game. It is used when the class is instanced. In java this would have been the only public class.

RUNS:

1. A black screen with white text

   Description automatically generated
2. A screenshot of a computer screen

   Description automatically generated
3. A screenshot of a computer

   Description automatically generated
4. A screenshot of a computer

   Description automatically generated

When designing this program, I designed it with different questions at first and then implemented the actual questions later. It was easier to test on 5 shorter questions than with the 17 or so longer ones from the assignment. Because of how it was designed I believe this code could be used to make 20 questions type game for any questions the user could want. As for adding in additional information, you would need to change up the code a little bit depending on what you want to display but it would be doable. For example if you wanted to put in pictures with each question you would have to have the code display the picture, which would probably mean adding in another method. However, I don’t believe this would be too hard to accomplish and would be easily doable if needed.

I did not follow a specific design pattern as much as I did the basic principles of OOP. For example, I made sure that every method in the class had a reason to exist and that I didn’t have one method that did everything.