You will be implementing a project in C++. The project is a text-based adventure game based on a storyline. You can define whatever storyline you want. The game will focus on a character overcoming obstacles to reach the final goal.

The game will have 3 levels. Each level should have 2 challenges. The challenges will represent obstacles. The project will use all the concepts that have been taught so far.

The project will include the following:

A **character.h** file where you define a class called Character. The character class has following members:

Date Members:

* name: string
* health: int
* damage: int

Member functions:

* Constructor: Character(string n);
  + Takes a string and assigns it to the name data member. Initialize health to 50, and damage to 5.
* void updateHealth(Character c);
  + Decrements the health of the current character by c's damage points.
* void updateDamage();
  + Increments damage points by 10.
* int getHealth();
  + returns the health of a character
* int getDamage();
  + returns the damage of a character
* string getName();
  + returns the name of a character
* void print();
  + Prints name and health of a character

For the main part, you can divide your code into multiple files, or just work on a single main.cpp file.

You will need to implement the following functions outside the class definition. These are not member functions of Character class:

* winner function: Finds the character who won (the character which has higher health points), and prints the name and health of that character
* level1 function: This function contains the challenges for level 1. You can implement both the challenges inside this function, or define separate functions for the challenges and call them inside this function.
* level2 function: Similar to level 1, this function will contain challenges for level 2
* level3 function: Similarly, this function will contain challenges for level 3
* main function: This is the function where you will call level1, level2, and level3 functions

You will need to implement the following aspects of the game:

* Declare two integer variables **MAX\_SCORE** and **GAME\_SCORE**. Initialize them both to 0.
  + A player can play multiple games
  + GAME\_SCORE will hold the score of each game.
  + MAX\_SCORE will hold the maximum of the scores among all games.
* Declare two Character objects. The first will be the **Player** and the other will be the **Oracle**. You can give whatever names you want to your characters. They both will start with health at 50 and damage at 5.
* The Player will be the one who solves the challenges. The Oracle does not solve any challenges.
* Before the player attempts any challenge, you need to check if it has at least 10 health points available. The game cannot continue if the player does not have any health points left.
* You don't need to have any checks on Oracle's health. The Oracle's health can even be negative.

The course of the game will be as follows:

1. At the start of your game, your GAME\_SCORE will be 0.
2. You will call your level1 function. The player attempts the first challenge. Once the player finishes the challenge (win/lose doesn't matter) they can go to the next challenge.
3. Then you call your level2 and level3 functions accordingly.
4. Everytime the player wins a challenge, do the following:
   * The oracle loses health based on player's damage (use the updateHealth function)
   * The player gains damage points (use the updateDamage function)
   * Increment the GAME\_SCORE by 10
5. And, everytime the player loses a challenge, do the following instead:
   * The player loses health based on oracle's damage (use the updateHealth function)
   * The oracle gains damage points (use the updateDamage function)
   * Do NOT update the GAME\_SCORE.
6. After the game ends (this can happen if the player completes level3, or player's health is less than 10), do the following:
   * Update the MAX\_SCORE to the GAME\_SCORE, if the current GAME\_SCORE is greater than MAX\_SCORE
   * Call the winner function and print details of the winning character
   * Ask the user if they want to restart the game. If yes, go back to Step 1 and DO NOT reset any character's health. Otherwise, end the game and print the MAX\_SCORE.

**Submission instructions:**

DO NOT send me your code files as submissions. I will not be accepting your code files as submission. You can definitely ask for help during class or office hours, or share your code if you are stuck somewhere and I will help you with it. Just remember, showing the code when asking for help is not same as submission.

The submission will be a presentation. You don't have to make a Powerpoint presentation for this.

There will be two parts to the presentation: Code run (where you play the game and show as many aspects as possible of the game), and Q/A (where I ask you one or two questions in the code. These questions will be very basic, but will be used to determine if you wrote the code on your own or copied from somewhere. These help students who put in the effort, get credit for their effort).

The final project is worth 15% of your grade. 10 points will be based on the code behaving as expected, and the rest 5 points will be based on your responses to the Q/A.

The final project presentation is scheduled for December 6, 2024. You need to give the presentation in-person in my office. This is not a team project, and needs to be implemented individually by each one of you.

If you will not be able to come on the day, you can present earlier in my office during office hours. No presentation after December 6th will be accepted. So, if I did not receive your presentations by then, I will be assigning a 0.

You can also present earlier in my office if you are done early. In that case, you don't need to come on the 6th of December.

The TA will not be taking presentations. You can only present to me.