CSCI 1300 CS1: Starting Computing
Ashraf, Cox, Spring 2020
Project 3: Create your own project

For the Final Project, you will implement a turn-based, text (adventure) game in C++, or another similar interactive concept. The minimum requirements for the project can be found below. From known games, suggested examples include Where in the World is Carmen Sandiego?, Final Fantasy, and MUD. Of course, you are encouraged to come up with a novel and unique idea in order to distinguish your project from the others.

Requirements

The only (minimum) requirement for this final project is that you include the following programming features in a C++ program:

Implementation requirements:
- 3 user-defined classes (one class can be a driver class)
- 4+ data members in two of the classes
- 1+ array of objects from a class that you created.
- 6+ if-else statements
- 4+ loops (while loops, for loops, do-while, in total)
- 2+ nested loops
- File IO (both reading from a file and writing to a file)

Project feature requirements:
- The project must have interactive components (ask the player for inputs, create menus for choices, and so on). It's a game!
- Game stats should be displayed at each turn. It's more exciting and meaningful! Also, these stats help debug the code.
- Your project must include:
  - At least 5 menu options (other than Quit/Exist)
  - At least two of these options must have a second layer of menu options
  - At least 2 menu options(primary or secondary layer of the menu) should include a random component, at least one each from the following:
    - The value of the variable is selected at random from a certain range of values (i.e. select a value at random between 1 and 6)
    - A probability value determines one of the outcomes (i.e. there is a 60% change a certain event will occur)

Timeline

Wednesday, April 8: Mandatory project design meeting. You need to have a meeting with professors/TAs/CAs/LAs to discuss your project ideas and code skeleton. You should bring your class files and code (a list of classes and methods you will implement) to the meeting. During the meeting, they will go over your classes to see if they're feasible and meet requirements.

This meeting is mandatory. Prior to Monday, March 30,you must have reserved a slot on **the project 3 design meeting scheduler** on Moodle and have the meeting date reserved as a day on or before Wednesday, April 8.

Wednesday, April 8, 6:00pm: Submit class files & Code Skeleton. Your .h files should be complete with all the data members and member functions you will be using for each class. For the class implementation .cpp files, you should fully implement simple functions like your getters and setters. For more complex functions you can include function stubs with detailed comments.

For example, if we were stubbing a function to implement bubble sort and return the number of swaps we might give in our code skeleton:

```
/*
  1. Compare adjacent elements. If the first is greater than the
     second, swap them.
  2. Do this for each pair of adjacent elements, starting with the
     first two and ending with the last two. At this point the last
     element should be the greatest.
  3. Repeat the steps for all elements except the last one.
  4. Repeat again from the beginning for one less element each time,
     until there are no more pairs to compare.
*/
int bubble_sort(int arr[], int size)
{
    int swaps = 5;
    return swaps;  // function returns expected type (int)
}
```

Thursday, April 23, 6:00 pm: Final Deliverables. Your project will be due on Thursday, April 23, at 6:00 pm. You must submit a zip file to Moodle, including all .h (or .hpp) and .cpp files. The submission should compile and run. TAs will be also grading comments and style.

Monday, April 27, 6:00 pm: Project Report. Write a 1-2 page report containing the following reflection questions:

1. How did you prepare for the project?
2. How did you develop our code skeleton? In what way(s) did you use your code skeleton?
3. Reflect on how you could have done better or how you could have completed the project faster or more efficiently?
4. In addition, write a paragraph answering the following question, in the context of the Project in CSCI 1300:
   Did you have any false starts, or begin down a path only to have to turn back when figuring out the strategy/algorithm for your Final Project program? Describe in detail what happened, for example, what specific decision led you to the false starts, or, if not, why do you think your work had progressed so smoothly. In either case, give a specific example.

The report should be a 1-inch margin, single space, 12pt font size, times new roman. Submit a report as PDF to Moodle.

Interview grading. Sign up, between April 15 and April 22, for the interview grading slot on Moodle. The interviews will take place between April 23 and April 30. If you don't sign-up between April 15 and April 22 and you miss your interview, then no points will be awarded for the project. The schedulers for interview grading will be available a couple of days before the deadline of this project.

During the interview grading, TAs will be playing your game and asking questions about your game. They will also ask about your implementations and conceptual questions.

Extra Credit

Option1(3pt): Implement a sorting algorithm. Write your implementation of sorting algorithm --- do not use a Library function or any outside resources) and apply it to a task in your program

Option2(7pt): Present your project in a video. Make a 5 minute (+ or - 1 min) video explaining:
- The project idea
- Implementation ana approach
- A demonstration of the working project

Work Alone or in a Group of no more than two.

For the final project, you are allowed to work with (at most) another student that is currently in CSCI1300 this semester. You may also choose to work by yourself if you so choose. If you decide to work as a team, we expect that you will be contributing to the project equally. TAs will be grading your work as a group, so both parties will receive the same grade.

Collaboration

All work for this assignment (and of course in general) must be original work. Every project (group or single) must be original and not similar to ANY other source (internet, paid tutor, fellow student, past years projects, etc). You may collaborate and brainstorm with other people, but each assignment submission is to be the unique creation of the author(s). Your code must be original and unique. Your work may not include code taken from online resources like Chegg or StackExchange, or from other students (past or present), even with modification. Any such instances constitute Academic Dishonesty (passing off others' work as your own) and will earn you a 0 on the assignment and a trip to the Honor Code Council. If a group project is found to be non-unique and deemed worthy of being sent to Honor Code Council then BOTH parties are guilty (even if one party decides to "take the blame"). Choose your groups wisely! If you aren't sure if something is okay, then please just ask!

Project 3 points

Project 3 is 15% of your final grade. Here is a summary of the points.

5 points - attending the project meeting with a TA.
5 points - for submitting your code skeleton and class files on time.
5 points - submitting the project report as described earlier.

20 points - Implementation requirements.
The implementation requirements are specified on the 1st page of this write-up.

30 points - interview grading.
- TA's questions about your project
- Algorithms descriptions, comments, good style
- Code compiles

35 points - project functionality
- The game plays at outlined in the project description
- Your solution accounts for user error

Project 3 total: 100 points

Note: if your code does not compile, you cannot score above 40 points for the project.