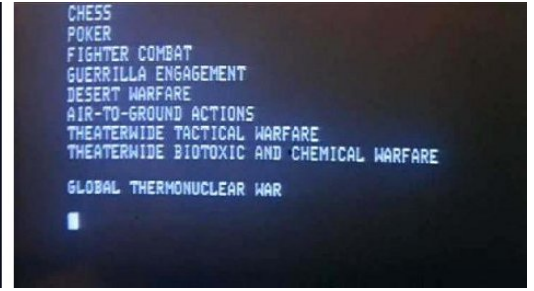


CS 105 (C++)

Assignment 5: Guessing Game

-WarGames (1983)**I. Overview**

In this assignment, you will implement a text-based number guessing game.

The user will be prompted for the top and bottom of a number range, and a random number in that range (inclusive) will be generated. The user will then be prompted for guesses, and will be given feedback until the number is guessed.

At the heart of your code will be a class that generates the random number and gives feedback on guesses.

The following are a few technical details you will need to use in your code.

- C++ has a built-in boolean type. It is called `bool`, and its legal values are the keywords `true` and `false`.
- To get input from the user, you will use `cin`. This is analogous to `cout`, but uses the right-shift operator instead of the left-shift operator. For example, to read from standard input into an integer `n`, you could use `cin >> n;`
- Be sure to use the new non-`.h` version of `<iostream>` for your input and output, as described in this [hello, world discussion](#), along with the required `using` statement.
- To generate a random number in C++ with `rand()`, use the C++ version of `<stdlib.h>`, which is `<cstdlib>`. This header is analogous to `<iostream>` in that it has no `.h`, and it relies on the same `using` statement (although including this `using` statement once will cover both header files).

The following are some important details of how your program must be implemented.

- You will create a class called `Game`, with the following characteristics.
 - This class will include `public` member function `generateNumber`. This function will receive two integer arguments to define the lower and upper bounds (inclusive) for the new random number and will return nothing.
 - The random number created by `generateNumber` will be stored in a `private` integer data member of `Game` called `randomNumber`.
 - `Game` will also include the `public` member function `tryGuess`, which takes an integer guess as its argument, and returns a `bool` with the value `true` for a correct guess and `false` for an incorrect guess. In addition to returning the proper value, `tryGuess` presents the user with the feedback on whether their guess was too high or too low.
 - There will be no other data or function members in `Game`.
 - You must choose one of the member functions to be defined inside the class, and one to be defined outside the class.
- `main()` will include the following.
 - Prompt the user for low and high limits of the number range.
 - Use an instance of `Game` to produce the random number and store it internally. (Note that the number's value is never known outside of `Game`.)
 - Repeatedly prompt the user for guesses until the correct answer is found, using the `Game` object to provide feedback for the guesses directly to standard out.

II. Grading

The following is a list of specific assignment requirements, along with the grade value for each (out of a total of 10 points for the assignment).

- Minimum Requirements

- Proper random number generation.
- Proper input and output.
- Proper game operation.
- Your work must be submitted in a single file called `main.cpp`.
- This file must compile on a department UNIX machine with the following command:

```
g++ main.cpp -o a5
```
- Before evaluation, your code must be submitted via `turnin`, using the following command on a department UNIX machine:

```
turnin --submit dlessin a5 main.cpp
```
- **Graded Elements**
 - Proper use of `bool` type.
 - Proper use of C++ Standard Library headers (as described above).
 - Proper use of `cin`.
 - Proper use of `cout`.
 - Proper use of `rand()`.
 - Proper use of access specifiers in `Game`.
 - One member function defined inside `Game` and one defined outside.

III. [The More You Know](#)

The following are some additional items that may be very important for you to know about this assignment.

- In many ways, this project should be pretty straightforward to implement. There is not much logic or computation to be done. It is primarily an opportunity to exercise some of the basic concepts and syntax of defining and using classes in C++.
- Your program does not have to be very long at all. (My implementation was about 50 lines altogether.) If yours is becoming particularly long or complex, you're probably doing more work than you have to.