

## **CSCE 1030: Project 1**

### **Due: 11:59 PM on Monday, October 09, 2023**

#### **PROGRAM DESCRIPTION**

In this project, you have to write a C++ program to design a game that involves guessing a number between two randomly generated numbers. You cannot see the randomly generated at start. If you guess correctly, you earn points. If you guess incorrectly, you lose points. You can keep on guessing as long as you have points remaining. You can also choose to display one of the two numbers (either the lower bound or the upper bound) to make the game easier. If you choose to display a number to get help, you earn less points for correct guesses and lose more points for incorrect guesses. You CANNOT, of course, display both bounds. If you COULD display both, there will be no guessing.

**DO NOT USE ARRAYS AND PROGRAMMER DEFINED FUNCTIONS IN YOUR PROGRAM. IF YOU DO SO, YOU WILL GET A ZERO. WE WILL LEARN ABOUT THOSE IN CHAPTERS 5 AND 6 LATER.**

**Note that a player can cheat in this game by repeatedly entering a correct guess. But we will ignore that scenario for this project.**

#### **PROGRAM REQUIREMENTS**

1. As with all projects in this course, your program's output will display your name, your EUID, your e-mail address, the department name, and course number. This means that your program will print this information to the terminal (see the sample output).
2. Declare an integer variable and initialize it to 100. This variable represents the number of points each player starts the game with.
3. Declare an enumeration constant with values displayLeft, displayRight, Guess, Change and Exit (in order) and assign integer values 1 to 5 to them respectively. These five items will correspond to the menu choice you will provide the player.
4. Using a suitable message, ask the player for the name of the player. The name may have multiple words.
  - Only alphabets (A-Z or a-z) and whitespaces are permitted in the account name.
    - If the player enters any other characters in the name, you need to generate an error message and ask for the name again.
    - Your program must keep on asking the player to enter the name until the player enters it correctly.
  - The player may type the name in either uppercase or lowercase, but you need to convert every initial to uppercase and every other alphabet to lowercase.
5. Generate two seeded random integers in the range of 100 to 200, inclusive. The first integer MUST BE less than the second integer. Using a loop of choice, keep on generating the integers until it is so.

6. Declare two integers and initialize them with -1 and display them. Since you cannot display the actual numbers being randomly generated, you will only display these integers. In case, the player wants to see the randomly generated integer, you will replace one of the -1 with the actual random number being generated.
7. Based on the enumerated constant data of Step 3, generate menu choice for the player. Using an integer variable, ask the player to select from the menu. See Sample Output.
8. Design a switch-case block with a default case, use the enumerated data constants of Step 3 as your cases. Based on the player input of Step 7, one of the cases will execute.
  - You must use a variable of your enumeration constant type as the switching expression.
  - If the player chooses to display the integer on the left (the lower bound), display the first randomly generated integer instead of -1 on the left.
    - Provide a message saying a correct guess will only earn 1 point and an incorrect guess will lose 10 points in the game.
  - If the player chooses to display the integer on the right (the upper bound), display the second randomly generated integer instead of -1 on the right.
    - Provide a message saying a correct guess will only earn 1 point and an incorrect guess will lose 10 points in the game.
  - In both cases, make sure the player cannot display both numbers/bounds.
  - If the player chooses to guess a number between the two bounds, ask the player to enter the guess.
    - If the guessed number is between the two bounds (i.e. the player guess correctly), increment the points, otherwise decrement the points.
    - If the player has displayed either of the two bounds, you need to increment by 1 point and decrement by 10 points, when appropriate.
    - If the player has NOT displayed any of the two bounds, you need to increment by 5 points and decrement by 5 points, when appropriate.
    - Update the player with the points balance after every guess.
    - Note that the guess CANNOT be the two bounds, it has to be strictly between the bounds to be a correct guess. For example, if the generated numbers are 150 and 160, both 150 and 160 are incorrect guesses.
  - If the player chooses to change the random numbers, regenerate the two seeded random numbers, still making the first number smaller than second number. Deduct 1 point for this choice and provide the player updated points balance.
  - If the player choose to Exit, display a suitable goodbye message using the name of the player and display the final points balance.
  - If the player enters a wrong choice, use the default case to provide an error message and ask the player to enter again.
9. You must use a suitable loop of choice to allow the player to repeat Steps 7 and 8 i.e. choosing from the menu and executing the corresponding case, until the use chooses to Exit or the points balance falls below zero (becomes negative).
10. Your program source code should be named **"euidProject1.cpp"**, without the quotes. where euid should be replaced by the EUID of the submitting student. If you are working in a group, you must enter their names and EUIDs as comments in the code file. Only one submission per group is needed.

11. Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., cell01, cell02, ..., cell06), so you should make sure that your program compiles and runs on a CSE machine.

### **DESIGN (ALGORITHM):**

On a piece of paper (or word processor), write down the algorithm, or sequence of steps, that you will use to solve the problem. You may think of this as a “recipe” for someone else to follow. Continue to refine your “recipe” until it is clear and deterministically solves the problem. Be sure to include the steps for prompting for input, performing calculations, and displaying output.

You should attempt to solve the problem by hand first (using a calculator as needed) to work out what the answer should be for a few sets of inputs.

Type these steps and calculations into a document (i.e., Word, text, or PDF) that will be submitted along with your source code. Note that if you do any work by hand, images (such as pictures) may be used, but they must be clear and easily readable. This document shall contain both the algorithm and any supporting hand-calculations you used in verifying your results.

### **TESTING:**

Test your program to check that it operates as desired with a variety of inputs. Then, compare the answers your code gives with the ones you get from hand calculations.

### **SUBMISSION:**

- Your program will be graded based largely upon whether it works correctly on the CSE machines, so you should make sure your program compiles and runs on the CSE machines.
- Your program will also be graded based upon your program style. This means that you should use comments (as directed), meaningful variable names, and a consistent indentation style as recommended in the textbook and in class.
- We will be using an electronic homework submission on Canvas to make sure that all students hand their programming projects on time. You will submit both (1) the program source code file and (2) the algorithm design document to the **Project 1** dropbox on Canvas by the due date and time.
- Projects are meant to be problem-solving exercises and are designed to help you practice your coding on larger projects with various pieces of functionality.
- If you choose to work alone, the coding should be primarily your sole work.
- If you choose to work in a group, you are allowed to get assistance from group members when working on these assignments. However, each student is required to report the name(s) of the students they worked with on the assignment. The maximum size of the group is 4 students, you can of course form a smaller group.

- Cheating for these assignments is defined as copying from a fellow student outside your working group, or copying from the web, or any other sources outside the class. You should not copy someone else's code or let a classmate examine your code if you have not identified as working in a group for your homework.
- As a safety precaution, do not edit your program (using vim or nano) after you have submitted your program where you might accidentally re-save the program, causing the timestamp on your file to be later than the due date. If you want to look (or work on it) after submitting, make a copy of your submission and work on that copy. Should there be any issues with your submission, this timestamp on your code on the CSE machines will be used to validate when the program was completed.

#### **SAMPLE OUTPUT 1 (This shows how Exit works)**

```
$ ./a.out
+-----+
|           Computer Science and Engineering           |
|           CSCE 1030 - Computer Science I             |
|           Student Name EUID euid@my.unt.edu           |
+-----+
Enter your name:john123 smith
Your name can only have alphabets or spaces. Enter again.
Enter your name:john smith
-1      -1
1. Display Right Number
2. Display Left Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?5
Bye John Smith !!!
Your final points balance=100
```

#### **SAMPLE OUTPUT 2 (This shows how Changing Random Numbers work)**

```
$ ./a.out
+-----+
|           Computer Science and Engineering           |
|           CSCE 1030 - Computer Science I             |
|           Student Name EUID euid@my.unt.edu           |
+-----+
Enter your name:john c smith
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
```

```

5. Exit
What do you want to do?4
Your remaining points=99
Generating new boundaries.
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?5
Bye John C Smith !!!
Your final points balance=99

```

### **SAMPLE OUTPUT 3 (This shows how Displaying Left Bound works)**

```

$ ./a.out
+-----+
|          Computer Science and Engineering          |
|          CSCE 1030 - Computer Science I            |
|          Student Name EUID euid@my.unt.edu         |
+-----+
Enter your name:sheila hArris
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?1
You will only get 1 point for guessing correctly and lose 10 points
for guessing incorrectly, now.
152     -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?2
You have already displayed the left boundary, you cannot display
both.
152     -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?5

```

Bye Sheila Harris !!!  
Your final points balance=100

**SAMPLE OUTPUT 4 (This shows how Displaying Right Bound works)**

```
$ ./a.out
+-----+
|           Computer Science and Engineering           |
|           CSCE 1030 - Computer Science I             |
|           Student Name EUID euid@my.unt.edu          |
+-----+
Enter your name:JOHN SMITH
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?2
You will only get 1 point for guessing correctly and lose 10 points
for guessing incorrectly, now.
-1      156
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?1
You have already displayed the right boundary, you cannot display
both.
-1      156
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?5
Bye John Smith !!!
Your final points balance=100
```

**SAMPLE OUTPUT 5 (This shows how Guessing a number works and how points increase/decrease based on the guesses made)**

```
$ ./a.out
+-----+
|           Computer Science and Engineering           |
|           CSCE 1030 - Computer Science I             |
|           Student Name EUID euid@my.unt.edu          |
+-----+
Enter your name:John SMITH
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:154
You guessed correctly. You get 5 points.
Your remaining points=105
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 5 points.
Your remaining points=100
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 5 points.
Your remaining points=95
-1      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
```

What do you want to do?3

Enter your guess:2000

You guessed incorrectly. You lose 5 points.

Your remaining points=90

-1        -1

1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit

What do you want to do?1

You will only get 1 point for guessing correctly and lose 10 points for guessing incorrectly, now.

152       -1

1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit

What do you want to do?3

Enter your guess:2000

You guessed incorrectly. You lose 10 points.

Your remaining points=80

152       -1

1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit

What do you want to do?3

Enter your guess:2000

You guessed incorrectly. You lose 10 points.

Your remaining points=70

152       -1

1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit

What do you want to do?3

Enter your guess:2000

You guessed incorrectly. You lose 10 points.

Your remaining points=60

152       -1

1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers



```
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=50
152      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=40
152      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=30
152      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=20
152      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=10
152      -1
1. Display Left Number
2. Display Right Number
```

```
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=0
152      -1
1. Display Left Number
2. Display Right Number
3. Guess a number in between
4. Change numbers
5. Exit
What do you want to do?3
Enter your guess:2000
You guessed incorrectly. You lose 10 points.
Your remaining points=-10
You are out of points.
Bye John Smith !!!
```