

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

Due Sunday, September 30th, by 6 pm

+5% bonus if submitted by Friday September 28th 11:55 pm,

+2% bonus if submitted by Saturday September 29th, 11:55 pm

This assignment is due **September 30rd, by 6 pm**

- ***All 3 components*** (Cloud9 workspace, Moodle CodeRunner quiz attempts, and zip file contains your cpp file(your solution and algorithm in pseudocode), your outputs) **must be completed and submitted by Sunday, September 30th, 6:00 pm for your homework to receive points.**
- Complete all the components by **Friday September 28th 11:55 pm** to receive a 5% bonus, and complete submissions before **Saturday September 29th 11:55 pm** to receive a 2% bonus.

Interview Grading:

- This assignment requires a grading interview. If you do not schedule (on moodle) and complete an interview grading session (in person) with your TA, you will receive a 0 for the whole assignment, regardless of the score from the autograder.
- If you need to reschedule an interview grading appointment and do not inform your TA at least 24 hours in advance, you will also receive a 0.
- The interview grading will start on October 2nd. (after Practicum 1)

Objectives:

- Understanding the working of **loops** - while statements and also work with nested loops
- Understand and work with **if-else** conditionals and **switch** statements.
- Writing and testing C++ functions
 - Understand problem description
 - Design your function:
 - come up with a step by step algorithm,
 - convert the algorithm to pseudocode
 - imagine many possible scenarios and corresponding sample runs or outputs
 - Convert the pseudocode into a program written in the C++ programming language
 - Test it in the Cloud9 IDE and submit it for grading on Moodle

Jeopardy Dice

Note: This is part of Projects section and constitutes to 5% of your grade

This project's objective is to create a Jeopardy Dice game in which one player (the user) competes against the computer to reach 100 points. You will turn in one file called `jeopardy_dice.cpp`. Read this document about the standards for homework submission before you turn your work in.

Game Mechanics

- There are two players: the user and the computer, who alternate turns until one of them reaches 100 points or higher.
- The user is always the first player and starts the game.
- If any player reaches 100 points or more at the end of their turn, the game ends immediately and the other player does not get another turn.
- During one turn the player (either the user or computer) accumulates `turnTotal` over a series of dice rolls.
 - If they roll a 1 or 6, their turn ends and the `turnTotal` becomes 0.
 - If they roll a 3, their turn ends and `turnTotal` becomes 15 (irrespective of what they collected up to that roll)
 - If they roll a 2, 4, 5, the value on the dice accumulates to `turnTotal`
 - The value on the dice is selected randomly during every roll.
- At the end of the turn, `turnTotal` is accumulated to `playerTotal` (either user or computer based on the turn).
- During a turn, **before** each roll, if the player is the user, they are given two options:
 - Continue rolling
 - Hold; if the user chooses to Hold the user's turn ends and the turn is passed to the computer.
- If the player is the computer, they will always continue rolling until their turn total reaches the value 10 or higher. If the `turnTotal` for the computer is 10 or higher, the computer's turn ends and the turn is passed to the user.

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

Specifications: You are given the following template (in the file `hmwk4.cpp`, posted on Moodle):

```
// CSCI1300 Fall 2018
// Author: <Your name>
// Recitation: <123> - <Your TA name>
// Cloud9 Workspace Editor Link: <https://ide.c9.io/ .....>
// Project1
//Put your pseudocode for the entire game here (see template in hmwk4.cpp)

#include <iostream>
#include <cmath>
#include <cstdlib>
#include <unistd.h>
#include <stdio.h>
using namespace std;

/**
 * rollDice
 * returns a random integer between 1 and 6, works as rolling a dice.
 * return value, int number (1-6)
 */

int rollDice()
{
    return random() % 6 + 1;
}

// your 3 + functions go in here ...

/**
 * game ()
 * driver function to play the game
 * the function does not return any value
 */
void game()
{
    // your solution goes here
}

int main()
{
    // start the game!
    game();
    return 0;
}
```

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

- Your solution must have an algorithm in pseudocode explaining how you are approaching the problem, step by step.
- Your `main()` function will have just one statement: a call to the function `game()`. Do not modify `main()`.
- `game()` function is the driver function of your program and we will be testing your `game()` function on Moodle (CodeRunner).
 - The function does not take any input parameters.
 - The function does not return any value.
- Your solution should have at least 3 functions apart from `game()` and `rollDice()` functions. Since the functions you will create will be called from inside the `game()` function, their definition should be above the definition of `game()`.
- You are given the freedom to design the functions of your choice. There are no restrictions on how many functions you create.
- Creating functions that does nothing is not permitted. Each function should have a purpose and should be well documented.
- Out of the three functions (or more) you will create,
 - One of the functions must have at least 1 input parameter
 - One of the functions must return a value
 - One of the functions should not return anything, but it should print/display something.

Input and Output

You may assume that the user will only enter valid input. Notice that the program only prompts the user if they want to roll again if it is the user's turn, and that the prompt is case insensitive (the user may type either "y" or "Y", "n" or "N").

As the game progresses, we will use output statements to check the progress of the user (or the computer) during a turn, and at the end of each turn. Below are two possible examples of program output. User input is shown **in bold**. Your program should **reproduce** the output shown below (excluding differences due to random values).

Example Output 1	Example Output 2
Welcome to Jeopardy Dice! It is now human's turn Do you want to roll a dice (Y/N)?: y You rolled a 2	Welcome to Jeopardy Dice! It is now human's turn Do you want to roll a dice (Y/N)?: n computer: 0

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

<p>Your turn total is 2 Do you want to roll again (Y/N)?: y You rolled a 5 Your turn total is 7 Do you want to roll again (Y/N)?: y You rolled a 4 Your turn total is 11 Do you want to roll again (Y/N)?: y You rolled a 2 Your turn total is 13 Do you want to roll again (Y/N)?: n computer: 0 human: 13</p> <p>It is now computer's turn</p> <p>Computer rolled a 6 Computer turn total is 0 computer: 0 human: 13</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: Y You rolled a 2 Your turn total is 2 Do you want to roll again (Y/N)?: y You rolled a 5 Your turn total is 7 Do you want to roll again (Y/N)?: N computer: 0 human: 20</p> <p>It is now computer's turn</p> <p>Computer rolled a 1 Computer turn total is 0</p>	<p>human: 0</p> <p>It is now computer's turn</p> <p>Computer rolled a 2 Computer turn total is 2 Computer rolled a 5 Computer turn total is 7 Computer rolled a 4 Computer turn total is 11 computer: 11 human: 0</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: n computer: 11 human: 0</p> <p>It is now computer's turn</p> <p>Computer rolled a 2 Computer turn total is 2 Computer rolled a 6 Computer turn total is 0 computer: 11 human: 0</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: n computer: 11 human: 0</p> <p>It is now computer's turn</p> <p>Computer rolled a 2 Computer turn total is 2 Computer rolled a 5 Computer turn total is 7 Computer rolled a 1 Computer turn total is 0</p>
--	--

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

<p>computer: 0 human: 20</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: y You rolled a 4 Your turn total is 4 Do you want to roll again (Y/N)?: y You rolled a 2 Your turn total is 6 Do you want to roll again (Y/N)?: y You rolled a 3 Your turn total is 15 computer: 0 human: 35</p> <p>[omitted]</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: y You rolled a 1 Your turn total is 0 computer: 55 human: 71</p> <p>It is now computer's turn</p> <p>Computer rolled a 6 Computer turn total is 0 computer: 55 human: 71</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: y You rolled a 4 Your turn total is 4</p>	<p>computer: 11 human: 0</p> <p>[omitted]</p> <p>It is now computer's turn</p> <p>Computer rolled a 4 Computer turn total is 4 Computer rolled a 4 Computer turn total is 8 Computer rolled a 3 Computer turn total is 15 computer: 81 human: 4</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: n computer: 81 human: 4</p> <p>It is now computer's turn</p> <p>Computer rolled a 3 Computer turn total is 15 computer: 96 human: 4</p> <p>It is now human's turn</p> <p>Do you want to roll a dice (Y/N)?: n computer: 96 human: 4</p> <p>It is now computer's turn</p> <p>Computer rolled a 3 Computer turn total is 15 computer: 111 human: 4</p>
--	--

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

<p>Do you want to roll again (Y/N)?: y You rolled a 2 Your turn total is 6 Do you want to roll again (Y/N)?: y You rolled a 5 Your turn total is 11 Do you want to roll again (Y/N)?: y You rolled a 2 Your turn total is 13 Do you want to roll again (Y/N)?: y You rolled a 5 Your turn total is 18 Do you want to roll again (Y/N)?: y You rolled a 4 Your turn total is 22 Do you want to roll again (Y/N)?: y You rolled a 4 Your turn total is 26 Do you want to roll again (Y/N)?: y You rolled a 4 Your turn total is 30 Do you want to roll again (Y/N)?: n computer: 55 human: 101 Congratulations! human won this round of jeopardy dice!</p>	<p>Congratulations! computer won this round of jeopardy dice!</p>
---	---

Testing

It is your responsibility to test your program. The random element of this program makes it so that you should not expect to exactly reproduce the example outputs. However, aside from the specific numbers, your output (the `cout` messages) should exactly match these examples.

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

Submitting the assignment:

1. Complete the CodeRunner question.
2. Zip the following 3 (4) files into one file, name it *<firstName>_<lastName>_HW4.zip*
 - a. *<firstName>_<lastName>_hwk4.cpp*
 - b. *<firstName>_<lastName>_output1.txt*
 - c. *<firstName>_<lastName>_output2.txt*
 - d. *<firstName>_<lastName>_hwk4_ex.cpp* (if you have done extra credit)

`output1.txt` file and `output2.txt` file contain two of your run results. You run your code twice and paste the outputs in txt file. Submit the zip file under **Hmwk 4** submission link before **Sunday, September 30th, by 6 pm.**

Grading Rubric:

Criteria	pts
Coderunner	40
Algorithm	10
Comments and Coding style	10
Two run outputs	5
Interview grading	35
Total	100
Extra credit question	+20
Early submission bonus	5% or 2%

Marble Run (20pts Extra Credit).

Two players alternately take marbles from a pile. In each move, a player chooses how many marbles to take. The player must take at least one but at most half of the marbles. Then the other player takes a turn. The player who takes the last marble loses.

You will write a function `marbleRun` in which the computer plays against a human opponent. Your function takes three inputs:

1. The initial size of the marble pile (integer between 10 and 100)
2. The mode of the computer (boolean)
3. Who takes the first turn (boolean)

The mode of the computer (0 or 1) indicates whether the computer plays *smart* or *stupid*. In stupid mode the computer simply takes a random legal value (between 1 and $n/2$, where n is the size of the pile) from the pile whenever it has a turn. In smart mode the computer

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

takes off enough marbles to make the size of the pile a power of two minus 1—that is, 3, 7, 15, 31, or 63. That is always a legal move, except when the size of the pile is currently one less than a power of two. In that case, the computer makes a random legal move.

You will note that the computer cannot be beaten in smart mode when it has the first move, unless the pile size happens to be 15, 31, or 63. Of course, a human player who has the first turn and knows the winning strategy can win against the computer.

Examples:

Example 1 : Computer gets to start and is playing in smart mode.

Result : Computer Wins

Let's start the game!

Initial size is: 100

First player to play : Computer

Is computer in smart mode? Yes

Computer is playing..

Computer took 37

Remaining marbles: 63

Your move. Marbles remaining : 63

55

Wrong Move. You can only choose between 1 to 31

Your move. Marbles remaining : 63

30

Remaining marbles: 33

Computer is playing..

Computer took 2

Remaining marbles: 31

Your move. Marbles remaining : 31

12

Remaining marbles: 19

Computer is playing..

Computer took 4

Remaining marbles: 15

Your move. Marbles remaining : 15

7

Remaining marbles: 8

Computer is playing..

Computer took 1

Remaining marbles: 7

Your move. Marbles remaining : 7

4

Wrong Move. You can only choose between 1 to 3

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

Your move. Marbles remaining : 7

3

Remaining marbles: 4

Computer is playing..

Computer took 1

Remaining marbles: 3

Your move. Marbles remaining: 3

1

Remaining marbles: 2

Computer is playing..

Computer took 1

Remaining marbles: 1

You lose. Computer won

Example 2 : Human gets to start but doesn't apply winning strategy and computer is playing in smart mode.

Result : Computer Wins

Let's start the game!

Initial size is: 100

First player to play : Human

Is computer in smart mode? Yes

Your move. Marbles remaining: 100

51

Wrong Move. You can only choose between 1 to 50

Your move. Marbles remaining: 100

31

Remaining marbles: 69

Computer is playing..

Computer took 6

Remaining marbles: 63

Your move. Marbles remaining: 63

10

Remaining marbles: 53

Computer is playing..

Computer took 22

Remaining marbles: 31

Your move. Marbles remaining: 31

15

Remaining marbles: 16

Computer is playing..

Computer took 1

Remaining marbles: 15

Your move. Marbles remaining: 15

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

7

Remaining marbles: 8

Computer is playing..

Computer took 1

Remaining marbles: 7

Your move. Marbles remaining: 7

3

Remaining marbles: 4

Computer is playing..

Computer took 1

Remaining marbles: 3

Your move. Marbles remaining: 3

1

Remaining marbles: 2

Computer is playing..

Computer took 1

Remaining marbles: 1

You lose. Computer won

Example 3 : Human starts and applies winning strategy and computer is in stupid mode.

Result : Human wins.

Let's start the game!

Initial size is: 100

First player to play : Human

Is computer in smart mode? No

Your move. Marbles remaining: 100

37

Remaining marbles: 63

Computer is playing..

Computer took 26

CSCI 1300 - Starting Computing

Instructor: Fleming

Project 1

Remaining marbles: 37
Your move. Marbles remaining: 37
6
Remaining marbles: 31
Computer is playing..
Computer took 12
Remaining marbles: 19
Your move. Marbles remaining: 19
4
Remaining marbles: 15
Computer is playing..
Computer took 6
Remaining marbles: 9
Your move. Marbles remaining: 9
2
Remaining marbles: 7
Computer is playing..
Computer took 3
Remaining marbles: 4
Your move. Marbles remaining: 4
1
Remaining marbles: 3
Computer is playing..
Computer took 1
Remaining marbles: 2
Your move. Marbles remaining: 2
1
Remaining marbles: 1
You Win. Computer loses