Program Purpose

Using Visual Studio 2017 create a CLR console application. Create a program plan and then convert it into C++ statements. Practice debugging, declaring variables, type casting, formatting output, if statements and input /output to / from the console window and files, input validation, loops, arrays and functions.

Always bring to class

1. Gaddis' book, How-to handouts from the Blackboard and your class notes.
2. This assignment sheet & the grade sheet for this lab already printed out.
3. USB Flash drive(s) or other storage media.

LATE PROGRAMS will be not be accepted. Please be sure to upload compressed solution and Word document grade sheet to Canvas by due date/time.

Mandatory Instructions

Your next assignment will be to write a program to play the children's game Chutes and Ladders. The game is played on a board which contains 81 squares, numbered 1 through 81. Each player starts at position 1 and throws one die to determine the number of squares to move on that turn. The first player to reach (or exceed) square 81 wins the game. "Chutes" and "Ladders" exist at various points on the board to enhance play. If a player's token lands on a chute, they "slide" down it and move backward on the board, but if they land on a ladder, they "climb" it and move forward on the board. The beginning and ending locations of each are:

Chutes - move back		**Ladders - move forward**	
top	**bottom**	**bottom**	**top**
11	4	8	16
15	7	12	27
30	21	28	39
44	31	33	46
58	43	48	55
64	54	59	75

Write the program to play the game with two players, Judy and Jane, all starting at position 1, and terminating when one player has reached square 81 or beyond. The output should be a table indicating the player, the move number, the position before the move, the value of the die, and the position after the move. A header for the table should be printed before the game begins. In addition, a line of dashes should separate each set of turns. The winning player should be printed at the end.

Method:

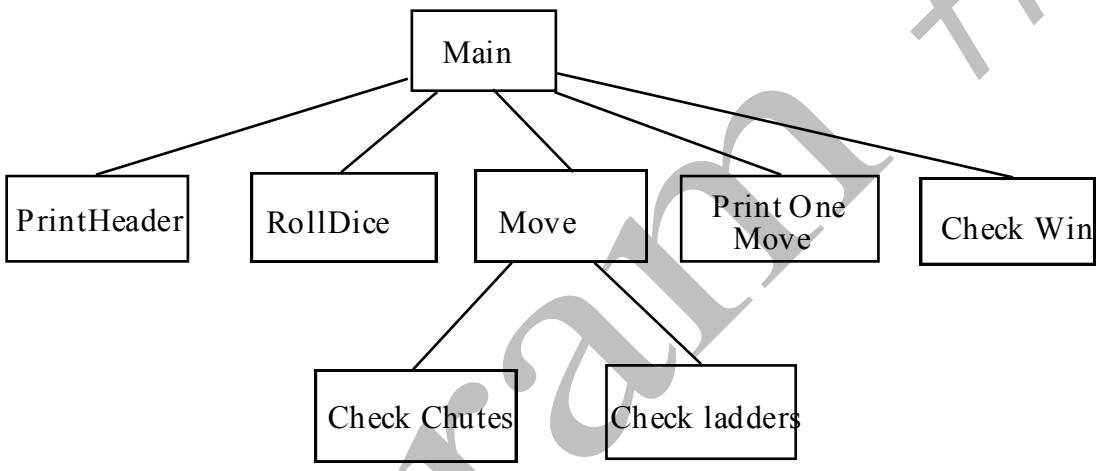
You should use functions which return a single value through the return statement or return no value (void) when you write this program. Functions should use value parameters as needed, but use reference parameters only if a function has more than one value to return. Use 2 variables to represent the players' position on the board. Both variables will be initialized to 1 since they start at position 1.

The main program should contain the following steps:

- determine your first and second player
- a loop to control the game
  - call a function, **CheckWin**, that returns a value of (true or false) to check if either player has reached or passed position 81. Place the call to this function where the Boolean expression normally goes in the while loop. Use the C++ **bool** data type.
  - in the loop you should have the following for each player:
    - a call to a function, **RollDice**, to determine the dice roll and return the value rolled
    - a call to a function, **Move**, to move a player and set its new position on the board
      - a call to a function, **CheckChutes**, to determine new position if landed on a chute
      - a call to a function, **CheckLadders**, to determine new position if landed on a ladder
    - a call to a function, **PrintOneMove** to print the status of the current player
- print the winner after the loop

Structure chart:

Note that the functions Check Chutes and Check Ladders should be called by the Move function to help with its operations.



Stepwise Refinement:

You have been given a ***suggested design*** in the structure chart which shows the functions that should be part of the program. Create first a "stub" program for your solution. A stub program should contain function prototypes, function calls and function headers – all with any needed parameters (arguments). Function bodies should be an empty with correct return value and braces. The main function should declare variables needed, and should also contain function calls. All of this should compile.

Additional coding items:

Use the following function to roll the die. You need to add the line **#include <cstdlib>** in order to use the **rand** function. The **rand** function, when called repeatedly, will return a series of pseudo-random numbers. You will get the same series of numbers each time. This is good for testing purposes.

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

Note: **(Do not** do this until your program is debugged and running correctly.) Add the following line as the **first line** in your main function.

```
srand(time(NULL));
```

The call to the **srand** function "seeds" the **rand** function with the system time thus assuring us of true random numbers. When you add **srand**, you will also need to use **#include <ctime>** at the beginning of your program.

Input: There is no input from the user other than the names for the two players.

Sample output:

```
Player   Move Number   Current Position   Die         New Position
*********************************************************************
Judy        1              1               4              5
Jane        1              1               3              4
---------------------------------------------------------------------
Judy        2              5               3             16
Jane        2              4               2              6
---------------------------------------------------------------------
Judy        3             16               2             18
Jane        3              6               1              7
```

(And so on until one player reaches or exceeds position 81.)

Function Prototypes

```
void  PrintHeader(void);
void  PrintTurnSeparator(void);    // Optional
int   RollDice();
int   Move(int, int);
void  PrintOneMove(string, int, int, int, int);
bool  Win(int, int);
int   CheckChutes(int);
int   CheckLadders(int);
```

Optional Instructions

Add a function called GetPlayerNames() that will prompt the user for each player name. Use do while loops to validate entered names to make sure they are of non zero length and that player 1 name is not equal to player 2 name.

```
void  GetPlayerNames(string &, string &);
```

Program Documentation & Style:

1. Declare all variables and constants that your program uses at the beginning of your program.
2. Your program should include two types of comments:
   a. Header Comments at the top including lines with:
      - Your name, course name, and class time
      - Program assignment number, program file name and due date

- A sentence or two explaining the purpose of the program
- A description of the input data needed by the program when you run it
- A description of the processing (calculations) done by the program
- A description of the results (output) produced by the program

b. In-line comments: There should be an in-line comment for each main step in your program. In general, this means a comment with each group of C++ statements that handle the declarations, input, processing, and the output steps of your program

3. Use meaningful identifier names
4. Include clear prompts for the user about entering the data
5. Include clear descriptions of the results when you display them

What to turn in?

1. Log into Canvas, locate this assignment, and upload the compressed project folders.
2. Upload the grade sheet (Word document) after you have edited it to provide requested information.
3. You're done.