Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Spring 2023

Homework 5 – A modified monopoly board game with object sharing

Due: 16/05/2023, Tuesday, 21:00

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!

Introduction

In this homework, you are asked to implement a simple monopoly board game using the *object sharing* concept of C++. The main program, which includes the game implementation using classes, is given to you. You are expected to write two classes: *Board* class and *Player* class. We are going to explain these classes in more detail in the following sections. Before that, we start with a general description of the game.

The Game

The game is a two-player turn-based game, which is played on a board-like data structure that you will create. The board consists of slots, and each slot represents a property that can be owned by any of the players. Both players start with an initial capital that is given in the beginning of the program as input. Both players start the game on the first slot of the board and move circularly on the board according to the dice result.

The players play in turns. First, player 1 (given the name "Ali") starts the game. Then player 2 (given the name "Liz") plays. This continues in turns until the game finishes (criteria for ending the game is given later in the document). Before playing in each turn, the player first rolls a dice to determine the amount of cells to move; the program uses a random number generator that returns an integer between (and incl.) one and six to simulate a dice (given in the main program; you will not implement this). When a player lands on any slot, they might have the option to buy it, if it is not owned by the other player. If, on the other hand, a player lands on other player's owned slot, he/she will have to pay a rent fee to that owning player. If a player lands on his own slot, no action is needed.

During the game, when a player lands on or passes through the starting slot, some amount of money will be deposited to his/her balance.

Program Flow and Inputs

The program starts by asking the user to enter 6 inputs that will be used in the game.

- *Maximum number of rounds*: indicates the maximum number of rounds for the game. In one round, first Ali plays, and then Liz plays. This input has to be a positive integer number; otherwise program stops.
- *Initial capital*: indicates the initial balance to be deposited to both players at the beginning of the game. This input has to be a positive integer number as well; otherwise program stops.
- *Passing starting slot deposit*: indicates the amount of money to be deposited when a user passes through (or lands on) the starting slot during the game. This input has to be a positive integer number; otherwise program stops.
- *Slot price*: the price of ownership for a slot. All slots on the board have the same price. This input has to be a positive integer number; otherwise program stops.
- *Number of slots*: indicates the number of slots of the board. This input has to be an integer number greater than or equal to 6; otherwise program stops.
- *The Seed*: is the seed value for the random number generator This is used in order to differentiate the dice outcomes in every game and round/player. This input has to be a positive integer number; otherwise program stops.

After reading these 6 inputs, the rounds will start. In each round, Ali (player 1) starts by rolling a dice which will result in a value in the range [1-6]. This is the number of slots that Ali will move circularly. After this move, Ali will land on a slot. He either:

a) lands on an unclaimed slot. In that case, a prompt is shown on the screen to give him the option to buy it. If he wants to buy, then the program checks if Ali has enough balance for this purchase. If so, the slot price will be deducted from his balance and he becomes the owner of this slot. If not, a message is displayed.
b) lands on his own slot. In that case, a message is displayed on the screen to indicate that he landed on his own slot. No extra action in this case.
c) lands on other player's (i.e. Liz's) slot. In that case, a message is shown on the screen to notify him that he landed on one of the Liz's slots. Here, no extra input is taken, but Ali pays the rent of that slot out of his own balance to Liz. The rent amount is always half of the *slot price*.

The same steps mentioned above are followed for Liz (Player 2) as well; and then, one round is considered to be finished. At the end of each round, the positions and balances of the players, and the board content (slot ownerships) will be displayed.

The game will continue until one of the players goes bankrupt (i.e. being balance < 0) at the end of a round and/or all of the game rounds are completed. The winner is determined differently in both cases; please refer to the provided main program to understand the winner selection logic.

The `Board` Class and the Corresponding Data Structure

The `Board` class will be used to create and manipulate a board on which the game is played. A board is represented by a circular linked list with a node structure shown below. Each node represents a slot of the board.

```
struct Node {
    string owner;
    Node* next;
    Node ();
    Node (string o, Node * p);
};
```

Note that constructors of `Node` have not been implemented here. You will implement them as you need it. When initializing, use the string "`None`" as the owner of a slot.

We do not use the default Node constructor in main program. If you will not use it as well, you do not need to implement it.

Now let us detail the `Board` class. It has only one private data member (let's call it `first`). This is basically the pointer to the *first* node that we add to the list.

The Board class must have a parametric *constructor* that takes the number of slots as parameter and creates a circular linked list that contains that amount of `Node`s. You must also write a *destructor* for the Board class.

Writing *copy constructor* and *assignment operator* are optional; however, if you use them, you have to write as deep copy ones. Please be aware of implicit usages of copy constructors (such as value parameter passing and value return). We did not use copy constructor and assignment operator in the main program (explicitly or implicitly).

The member functions of the `Board` class used in main are the followings. You must implement them as instructed.

display: This function will display the content of the board. Basically we display the owners of slots (`Ali`, `Liz,` or `None` if not owned). Please refer to the sample runs for output formatting, which is quite tricky since there are some extra characters to display. We expect you to apply pattern recognition capabilities from IF 100 to understand the rules of this display format.

who_owns: This function takes a `Node*` parameter and returns a string that corresponds to the owner of that node.

If needed, you may add extra functions to the `Board` class. It is allowed to have a function that returns `first`.

You will write the Node struct and class definition in Board.h file and the member function implementations in Board.cpp file.

The `Player` Class

The `Player` class will be used to manage the players of the game. There will be two player objects playing on the same board in a game. Thus, player objects must share a Board object using *object sharing* concept and principles of C++ as we have seen in the lectures. We have seen two different methods for object sharing in the course; due to our main function implementation, which is provided to you, you must use the reference variable method. Other than the shared `Board` object, the player class will have:

- a `string` private data member to store the player's name.
- an `int` private data member to store the player's balance.
- a `Node*` to indicate the current position of the player on the board.

The Player class needs a *constructor* that takes three parameters: (i) the shared Board object, (ii) the name of the player as a string, and (iii) the initial capital (as int). The constructor makes necessary initializations.

Other functions of the Player class are as follows.

move: This function takes an integer parameter, say `steps`. The function will move the player for `steps` slots on the board circularly. To do so, player's position will be updated. This function should return 1 if the player has passed by or stopped on the initial starting slot; returns 0 otherwise.

where_am_I: this function returns the player's current position on the board as a `Node*`

pay_to_player: this function takes one Player, say `otherPlayer`, and one integer, say `money`, as parameters. The function withdraws `money` from the balance of the player (i.e. Player object on which the function is called) and add it to the balance of `otherPlayer`.

get_balance: this function returns the player's balance**.**

deposit_money: this function takes an integer parameter and add it to the balance of the player.

buy_slot: this function takes an integer parameter for the price of a slot. It assigns the name of the player as the owner of the slot that the player is current on. It also withdraws the price of the slot from the player's balance.

is_bankrupt: this function returns true if the player's balance is negative; returns false otherwise.

display: this function displays the name of the player and its balance. You have to leave some spaces before the name depending on the current position of the player. Please refer to sample runs to figure out a rule to determine the amount of spaces.

You will write the class definition in Player.h file and the member function implementations in Player.cpp file.

Important Rule about using `friend`

In this homework, you are **not** allowed to use **`friend`** keyword, which means the use of friend functions and friend classes are prohibited. Our aim by this restriction is not to make your life miserable, but to enforce you to proper object oriented design and implementation.

Using Object Sharing Principles and Object-Oriented Design

In your program, the *Board* object must be shared by the *Player* objects. For this object sharing, you have to employ the method that uses **reference variables**.

It should be clear that you will write two classes for `Board` and `Player`. You need to analyze the requirements carefully and make a good object-oriented design for these classes. In this context, you have to determine the data members and design/implement member functions of each class correctly. We will evaluate your object-oriented design as well.

Provided main.cpp file

Together with this homework package, we provide the main program in **main.cpp**. This file has the `main` function, which contains the game implementation by using related class functions. In this homework, our aim is to reinforce object oriented design capabilities; thus, we did not want you to deal with the class usage, but focus on their design and implementation. Please examine these functions to understand how the classes are used. We will test your codes with this main.cpp with different inputs. You are not allowed to make any modifications in this file; you have to use other files for class definitions and implementations. Moreover, you will not submit main.cpp.

Files to be Submitted

You will upload total of four (4) files at CodeRunner. The files that you will upload are **`Board.h`**, **`Board.cpp`**, **`Player.h`** and **`Player.cpp`** files. You must use exactly these file names. This time, you will **not** copy-paste anything to the answer part while testing/submitting.

One of the header files is for the Player class (`Player.h`) and the other is for the Board class (`Board.h`; define `Node` struct here). Similarly, one of the cpp files is for the Player class (`Player.cpp`) and the other is for the board class (`Board.cpp`). In general, you will have class definitions in the header files and member function implementations in the cpp files.

You will **not** submit/upload `main.cpp`.

Please see the previous homework specifications for the other important rules and the generic submission guidelines

Sample Runs

Some sample runs are given below, but these are not comprehensive, therefore you have to consider **all possible cases** to get full mark. Inputs are shown in **bold** and *italic*.

Please do not try to understand the requirements by checking out the sample runs only. They may give you just an idea, but you have to read the requirements from the document and the main.cpp to fully understand what to do.

Sample Run 1 (reached max. number of rounds and tie)

```
Please enter the maximum number of rounds(>0):
3
Please enter the starting capital(>0):
1000
Please enter the passing starting slot deposit(>0):
500
Please enter the slot price(>0):
20
Please enter the number of slots(>=6):
6
Please enter the seed for the random number generation (>0):
1

******************************** Ali's turn **********************
[Ali] You are going to move 4 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

******************************** Liz's turn **********************
[Liz] You are going to move 6 move(s).
[Liz] You passed by (or landed on) the starting slot. 500 will be deposited to your balance.
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.

                        Ali 980
Liz 1500
None->None->None->None->Ali ->None
^                               |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 4 move(s).
[Ali] You passed by (or landed on) the starting slot. 500 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

******************************** Liz's turn **********************
[Liz] You are going to move 4 move(s).
[Liz] You stepped on one of Ali's slots, you will pay 10.
```

```
            Ali 1470
                          Liz 1490
None->None->Ali ->None->Ali ->None
^                                 |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 5 move(s).
[Ali] You passed by (or landed on) the starting slot. 500 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 2 move(s).
[Liz] You passed by (or landed on) the starting slot. 500 will be deposited to your balance.
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Liz] The slot has been sold to you.

      Ali 1970
Liz 1970
Liz ->None->Ali ->None->Ali ->None
^                                 |
|----<-----<-----<-----<-----<---v

The game came to an end.
Ali's balance is: 1970
Liz's balance is: 1970
Tie.
```

Sample Run 2 (reached max. number of rounds and Liz won)

```
Please enter the maximum number of rounds(>0):
4
Please enter the starting capital(>0):
1000
Please enter the passing starting slot deposit(>0):
300
Please enter the slot price(>0):
200
Please enter the number of slots(>=6):
7
Please enter the seed for the random number generation (>0):
2

******************************** Ali's turn **********************
[Ali] You are going to move 6 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

******************************** Liz's turn **********************
[Liz] You are going to move 4 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.
```

```
                                    Ali 800
                        Liz 1000
None->None->None->None->None->None->Ali
^                                       |
|----<-----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 4 move(s).
[Ali] You passed by (or landed on) the starting slot. 300 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

******************************** Liz's turn **********************
[Liz] You are going to move 5 move(s).
[Liz] You passed by (or landed on) the starting slot. 300 will be deposited to your balance.
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.

                  Ali 900
            Liz 1300
None->None->None->Ali ->None->None->Ali
^                                       |
|----<-----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 2 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 5 move(s).
[Liz] You passed by (or landed on) the starting slot. 300 will be deposited to your balance.
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.

                               Ali 900
Liz 1600
None->None->None->Ali ->None->None->Ali
^                                       |
|----<-----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 6 move(s).
[Ali] You passed by (or landed on) the starting slot. 300 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

******************************** Liz's turn **********************
[Liz] You are going to move 6 move(s).
[Liz] You stepped on one of Ali's slots, you will pay 100.

                        Ali 1100
                                    Liz 1500
None->None->None->Ali ->Ali ->None->Ali
^                                       |
```

```
|----<-----<-----<-----<-----<-----<---v

The game came to an end.
Ali's balance is: 1100
Liz's balance is: 1500
Liz won.
```

Sample Run 3 (reached max. number of rounds and Ali won)

```
Please enter the maximum number of rounds(>0):
5
Please enter the starting capital(>0):
1000
Please enter the passing starting slot deposit(>0):
200
Please enter the slot price(>0):
50
Please enter the number of slots(>=6):
6
Please enter the seed for the random number generation (>0):
9

****************************** Ali's turn **********************
[Ali] You are going to move 3 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

****************************** Liz's turn **********************
[Liz] You are going to move 2 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Liz] The slot has been sold to you.

                 Ali 1000
            Liz 950
None->None->Liz ->None->None->None
^                               |
|----<-----<-----<-----<-----<---v


****************************** Ali's turn **********************
[Ali] You are going to move 3 move(s).
[Ali] You passed by (or landed on) the starting slot. 200 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

****************************** Liz's turn **********************
[Liz] You are going to move 4 move(s).
[Liz] You passed by (or landed on) the starting slot. 200 will be deposited to your balance.
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Liz] The slot has been sold to you.

Ali 1200
Liz 1100
Liz ->None->Liz ->None->None->None
^                               |
|----<-----<-----<-----<-----<---v
```

```
******************************** Ali's turn **********************
[Ali] You are going to move 5 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 3 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Liz] The slot has been sold to you.

                              Ali 1200
                  Liz 1050
Liz ->None->Liz ->Liz ->None->None
^                                  |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 5 move(s).
[Ali] You passed by (or landed on) the starting slot. 200 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 1 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Liz] The slot has been sold to you.

                        Ali 1400
                        Liz 1000
Liz ->None->Liz ->Liz ->Liz ->None
^                                  |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 2 move(s).
[Ali] You passed by (or landed on) the starting slot. 200 will be deposited to your balance.
[Ali] You stepped on one of Liz's slots, you will pay 25.

******************************** Liz's turn **********************
[Liz] You are going to move 6 move(s).
[Liz] You passed by (or landed on) the starting slot. 200 will be deposited to your balance.
[Liz] You stepped on your own slot.

Ali 1575
                        Liz 1225
Liz ->None->Liz ->Liz ->Liz ->None
^                                  |
|----<-----<-----<-----<-----<---v

The game came to an end.
Ali's balance is: 1575
Liz's balance is: 1225
Ali won.
```

Sample Run 4 (Liz's bankruptcy)

```
Please enter the maximum number of rounds(>0):
9
Please enter the starting capital(>0):
40
Please enter the passing starting slot deposit(>0):
5
Please enter the slot price(>0):
40
Please enter the number of slots(>=6):
6
Please enter the seed for the random number generation (>0):
5

******************************** Ali's turn **********************
[Ali] You are going to move 3 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

******************************** Liz's turn **********************
[Liz] You are going to move 2 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.

                  Ali 0
            Liz 40
None->None->None->Ali ->None->None
^                               |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 2 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] Sorry, you don't have enough balance to buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 1 move(s).
[Liz] You stepped on one of Ali's slots, you will pay 20.

                              Ali 20
                  Liz 20
None->None->None->Ali ->None->None
^                               |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 2 move(s).
[Ali] You passed by (or landed on) the starting slot. 5 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] Sorry, you don't have enough balance to buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 6 move(s).
[Liz] You passed by (or landed on) the starting slot. 5 will be deposited to your balance.
[Liz] You stepped on one of Ali's slots, you will pay 20.
```

```
      Ali 45
                  Liz 5
None->None->None->Ali ->None->None
^                                 |
|----<-----<-----<-----<-----<---v


****************************** Ali's turn **********************
[Ali] You are going to move 4 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Ali] You didn't buy this slot.

****************************** Liz's turn **********************
[Liz] You are going to move 6 move(s).
[Liz] You passed by (or landed on) the starting slot. 5 will be deposited to your balance.
[Liz] You stepped on one of Ali's slots, you will pay 20.

                               Ali 65
                  Liz -10
None->None->None->Ali ->None->None
^                                 |
|----<-----<-----<-----<-----<---v

The game came to an end.
Ali's balance is: 65
Liz's balance is: -10
Ali won.
```

Sample Run 5 (Ali's bankruptcy)

```
Please enter the maximum number of rounds(>0):
18
Please enter the starting capital(>0):
80
Please enter the passing starting slot deposit(>0):
1
Please enter the slot price(>0):
80
Please enter the number of slots(>=6):
6
Please enter the seed for the random number generation (>0):
16

****************************** Ali's turn **********************
[Ali] You are going to move 5 move(s).
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] The slot has been sold to you.

****************************** Liz's turn **********************
[Liz] You are going to move 2 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Liz] The slot has been sold to you.

                               Ali 0
            Liz 0
None->None->Liz ->None->None->Ali
^                                 |
|----<-----<-----<-----<-----<---v


****************************** Ali's turn **********************
```

```
[Ali] You are going to move 5 move(s).
[Ali] You passed by (or landed on) the starting slot. 1 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] Sorry, you don't have enough balance to buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 6 move(s).
[Liz] You passed by (or landed on) the starting slot. 1 will be deposited to your balance.
[Liz] You stepped on your own slot.

                         Ali 1
            Liz 1
None->None->Liz ->None->None->Ali
^                               |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 6 move(s).
[Ali] You passed by (or landed on) the starting slot. 1 will be deposited to your balance.
[Ali] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
Y
[Ali] Sorry, you don't have enough balance to buy this slot.

******************************** Liz's turn **********************
[Liz] You are going to move 2 move(s).
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.

                         Ali 2
                         Liz 1
None->None->Liz ->None->None->Ali
^                               |
|----<-----<-----<-----<-----<---v


******************************** Ali's turn **********************
[Ali] You are going to move 4 move(s).
[Ali] You passed by (or landed on) the starting slot. 1 will be deposited to your balance.
[Ali] You stepped on one of Liz's slots, you will pay 40.

******************************** Liz's turn **********************
[Liz] You are going to move 2 move(s).
[Liz] You passed by (or landed on) the starting slot. 1 will be deposited to your balance.
[Liz] You stepped on an unclaimed slot, do you want to buy it? (Y/N)
N
[Liz] You didn't buy this slot.

            Ali -37
Liz 42
None->None->Liz ->None->None->Ali
^                               |
|----<-----<-----<-----<-----<---v

The game came to an end.
Ali's balance is: -37
Liz's balance is: 42
Liz won.
```