Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Fall 2016-2017

Homework 5–object sharing

Due: 14/11/2016, 23:00
(Late submission penalty: -20%)

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 HAVE TO write down the code on your own.
You CANNOT HELP any friend while coding.
Plagiarism will not be tolerated!

Introduction: This homework aims to make you practice on object sharing.You are asked to write **two** classes `Player` and `Board` that two `Player` objects share one `Board` via reference variables. The main function of the program is already given to you. We will explain details about the homework in the following sections.

Description: You are going to implement a game where there are 2 players and a board. Each player can roll a dice (which is a random number between 1 and 6). Each time she rolls a dice, her point increases by this random number and the number of her moves increases by one. The game continues as long as the total point amount is less than 50 or the total number of moves of the players is less than 15.

Using Object Sharing Principles and Object Oriented Design: As mentioned above, you have a single `Board` and two `Player` objects in your program. The `Board` object must be shared by the `Player` objects. For this object sharing, you have to employ reference variables.

From the above paragraph it should be clear that you will write two classes for `Board` (`which must be implemented in Board.h and Board.cpp`) **and** `Player` (`which must be implemented in Player.h and Player.cpp`)**. 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 member functions of each class correctly. We will evaluate your object oriented design, e.g., private-public members, reference variables, as well. Moreover, you are not allowed to use friend class or friend functions in your design.**

Program Flow: At the beginning of the main function, which is provided with the homework specification, initial values of the `Board` object are `totalPoint` set to 0 and `totalMove` set to 0.

The specifications of the `Player` class are as follows:

- `Player (Board& board, string name)`: The constructor of `Player` class has two parameters. It should assign the `Board` object such that the `Player` object shares the board. In addition, the player name is set with the given parameter. It should also set the `point` and `noMoves` of the player to 0; both are integer member variables. Note that you should define appropriate data members in the `Player` class.

- `bool rollDice()`: A player rolls a dice and create a random number between (1-6); the value of a dice must printed on screen. The function increases the player's points by the random number created. Also, the function should increases the number of move of the player by one. After rolling a dice, the `totalPoint` and `totalMove` of the Board object must also be increased. The max value for `totalPoint` is 50 and for `totalMove` is 15 meaning that when one of these values exceeds the maximum, the game is finished and the function returns `false`. Otherwise, it will return `true`.

- `void display()`: The function should display the current information of a player including his/her name, points, number of moves.

The specifications of member functions of `Board` class are as follows:

- `void display()`: The function should display the total points and total number of moves of the board so far as shown in the sample runs.

Please note that you can assume all inputs are entered correctly, so you do not need any input check in your program. There can be other member functions of `Player` and `Board` classes. You should be aware of such functions and write them appropriately.

The console input of the code will be shared with you. You will only implement some part of it and the Board and Player classes.

Once the game starts, the players will take turns one after the other (by pressing enter key). When the game ends, the winner is the one who has more points on average per move (the double value of point/noMove). After each player took her turn the information about her will be displayed on console screen; the totalPoint and totalMove will be displayed after each round. During the game you might encounter a situation where the last move of the player causes totalPoint to be more than 50; hence the game is ended by considering the last move of the player valid.

Sample runs

Below, we provide some sample runs of the program that you will develop. The bold phrases are inputs taken from the user.

Sample Run 1

Please enter the name of the first player: **leyli**
Please enter the name of the second player: **Burcu**
Sdg
User must press enter to continue the game
Press any key to continue . . .

Sample run 2

Please enter the name of the first player: leyli
Please enter the name of the second player: Burcu
(Enter key is pressed)
dice: 4
Player name: leyli
point 4
Number of Moves: 1

Dfh
User must press enter to continue the game
Press any key to continue . . .

Sample run 3

Please enter the name of the first player: **leyli**
Please enter the name of the second player: **cengiz**
(Enter key is pressed)
dice: 2
Player name: leyli
point 2
Number of Moves: 1

(Enter key is pressed)
dice: 5
Player name: cengiz
point 5
Number of Moves: 1

Total points: 7
Total moves: 2

(Enter key is pressed)
dice: 4
Player name: leyli
point 6

Number of Moves: 2

(Enter key is pressed)
dice: 2
Player name: cengiz
point 7
Number of Moves: 2

Total points: 13
Total moves: 4

(Enter key is pressed)
dice: 1
Player name: leyli
point 7
Number of Moves: 3

(Enter key is pressed)
dice: 1
Player name: cengiz
point 8
Number of Moves: 3

Total points: 15
Total moves: 6

(Enter key is pressed)
dice: 2
Player name: leyli
point 9
Number of Moves: 4

(Enter key is pressed)
dice: 2
Player name: cengiz
point 10
Number of Moves: 4

Total points: 19
Total moves: 8

(Enter key is pressed)
dice: 5
Player name: leyli
point 14
Number of Moves: 5

(Enter key is pressed)
dice: 5

Player name: cengiz
point 15
Number of Moves: 5

Total points: 29
Total moves: 10

(Enter key is pressed)
dice: 3
Player name: leyli
point 17
Number of Moves: 6

(Enter key is pressed)
dice: 3
Player name: cengiz
point 18
Number of Moves: 6

Total points: 35
Total moves: 12

(Enter key is pressed)
dice: 4
Player name: leyli
point 21
Number of Moves: 7

(Enter key is pressed)
dice: 5
Player name: cengiz
point 23
Number of Moves: 7

Total points: 44
Total moves: 14

(Enter key is pressed)
dice: 3
Player name: leyli
point 24
Number of Moves: 8

Total points: 47
Total moves: 15
The game is finished and scores of players are as follow:
leyli 3
cengiz 3.28571
The winner is: cengiz
Press any key to continue . . .

Sample run 4

Please enter the name of the first player: **leyli**
Please enter the name of the second player: **cengiz**
(Enter key is pressed)
dice: 5
Player name: leyli
point 5
Number of Moves: 1

(Enter key is pressed)
dice: 5
Player name: cengiz
point 5
Number of Moves: 1

Total points: 10
Total moves: 2

(Enter key is pressed)
dice: 5
Player name: leyli
point 10
Number of Moves: 2

(Enter key is pressed)
dice: 5
Player name: cengiz
point 10
Number of Moves: 2

Total points: 20
Total moves: 4

(Enter key is pressed)
dice: 2
Player name: leyli
point 12
Number of Moves: 3

(Enter key is pressed)
dice: 2
Player name: cengiz
point 12
Number of Moves: 3

Total points: 24
Total moves: 6

(Enter key is pressed)
dice: 2
Player name: leyli
point 14
Number of Moves: 4

(Enter key is pressed)
dice: 6
Player name: cengiz
point 18
Number of Moves: 4

Total points: 32
Total moves: 8

(Enter key is pressed)
dice: 6
Player name: leyli
point 20
Number of Moves: 5

(Enter key is pressed)
dice: 6
Player name: cengiz
point 24
Number of Moves: 5

Total points: 44
Total moves: 10

(Enter key is pressed)
dice: 3
Player name: leyli
point 23
Number of Moves: 6

(Enter key is pressed)
dice: 4
Player name: cengiz
point 28
Number of Moves: 6

Total points: 51
Total moves: 12
The game is finished and scores of players are as follow:
leyli 3.83333
cengiz 4.66667
The winner is: cengiz
Press any key to continue . . .

Some Important Rules:
In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate.

When we grade your homeworks we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we may run your programs in *Release* mode and **we may test your programs with very large test cases**.

What and where to submit (PLEASE READ, IMPORTANT): You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. It'd be a good idea to write your name and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.
Name your cpp file that contains your program as follows:

"SUCourseUserName_YourLastname_YourName_HWnumber.cpp"

Your SUCourse user name is actually your SUNet username that is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw2.cpp

Do not add any other character or phrase to the file name. Make sure that this file is the latest version of your homework program. Compress this cpp file using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains your cpp file.

You will receive no credits if your compressed zip file does not expand or it does not contain the correct file. The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows:

SUCourseUserName_YourLastname_YourName_HWnumber.zip

For example zubzipler_Zipleroglu_Zubeyir_hw1.zip is a valid name, but

hw1_hoz_HasanOz.zip, HasanOzHoz.zip

are**NOT** valid names.

Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!
CS204 Team (Leyli Javid Khayati, Kamer Kaya)