

Advanced software development assignment 1

by Ethan Walton-Deaves

Submission date: 10-Dec-2021 04:39PM (UTC+0000)

Submission ID: 165807028

File name: 24855_Ethan_Walton-

Deaves_Advanced_software_development_assignment_1_382345_1018371693.doc (670.5K)

Word count: 1541

Character count: 6598

Advanced software development

Assignment 1

Explain how I used .ccp and .h files in my works.

In this assignment I used the head and ccps files for naming and working with numbers made from a random number generator. In my header files I have only 1 heading that I used for most of my classes. I inputted most of my code that makes the numbers work in my ccp file while I called them to the source file using its functions name and pointers. I believe that using the files this way helps as it stops repeatable code from appearing and cuts back on the amount of code needed in the source file. For example, in my game there are two players, and both are handled by the same head and ccps files. For example, when getting their numbers for the game to start they the code is set up in such a way that once the person inputs their names, they get their numbers. Then the system asked for the second name. Once the name is given the exact same code will act and will give player 2 their numbers.

Class for rules was set up so that I did not need to clog up the source file with lots of words where the main code should be. The header file called text is used here to help the programmer with large poetically repeatable text such as tutorials and rules. In the class two functions are stored. The first is the rules and the start of the game where the system asks whose going first and welcoming the player to the game. The other part is congratulating he player who won or saying it was unfortunately a draw.

Another class was used in asking what the person's name was. It sued a "cin" command where the user would input their own name. I used a pointer to point to the ccp file "person" to get this code and return their own name after they had inputted it.

Understanding that without head and ccp files the main code would have large amounts of repeatable code that we don't want and so, once set up, using pointers when needing to do a complex code that may take time to complete will not only look more professional but much easier just to code in a key phrase to access it immediately. Also once set up there is no need for the complex code to be repeated in the main file again in the future.

Making the dice game

Concept of the game

Creating this game, I took inspiration firstly on blackjack which is getting as close to 21 as possible without going over. So I tool that and instead of using cards I used dice. 6 dice is thrown, and you are to get as close to the number 18 as possible the program itself tells you the total number of those 6 random numbers. The program then asks you to input whose closer player one or player 2 and then the system congratulates you. Meanwhile behind the code some of the information such as the players name and the total they got get stored away in text file that is in the folder with the program on it called score.txt. You access it from there.

Text

The text in this game is simple and obvious to follow. The text in this program also wants a response from the user to input there name and who is closer to the number 18 after the numbers have been given. Text is also used when asking about the files and if you wish to delete the files data. Most of the rules told comes from its own header file which I point to when I need that information to be told. In this situation the information will be shown at the start using a pointer so that the text will appear immediately.

```

Dice Roller!
do you want to delete the recorded files data?
if yes then type 1 if no type 0

0
Hello there welcome to my program:
here you will randomly get give numbers and get the closet to the number 18!
Firstly start by figuring out who will go first!

Your first?    OK!

0
what is your name?
Ethan

```

```

text* text1 = new text();
cout << text1->getRules() << endl;

```

```

int text::getRules()
{
    cout << "Dice Roller!" << endl;

    int remove;
    cout << "do you want to delete the recorded files data?" << endl << "if yes then type 1 if no type 0" << endl << endl;
    cin >> remove;

    if (remove == 1);
    {
        ofstream myfile;
        myfile.open("score.txt", ios::out);
        myfile.close();
    }

    cout << "Hello there welcome to my program: " << endl << "here you will randomly get give numbers and get the closet to the number 18!" << endl;
    cout << "Firstly start by figuring out who will go first!" << endl << endl << "Your first?    OK!" << endl << endl;
    return 0;
}

```

Features

I used the class system to focus on the different players at the same time as the players will get the same amount of numbers and greetings no matter the case. This means that I get the code both players at the same time from my ".ccp files". Within my .ccp files I have a question which I ask the user. What is your name where they can answer and give whatever name they want to give which then provides to give the user their numbers using a random number generator. Then I ask the second player and the same question applies. Next, I ask who is the closet to the number 18? And ask for them to input ether the number 1 or number 2. if it is a draw, they can input draw and the code will then proceed to ether congratulate the player who won or say that it was too bad it was a draw.

Random number generator.

Firstly I can call a random number from the source file from the .ccp file. Anytime I input the code "cout << number1->GetNumber() << endl;" after announcing where to find it using "number* number1 = new number();". The random number generator is coded to give a random number between 0 – 6 however since a dice only has 1-6 numbers I have to give the random number

generator +1 so that it does not give me any 0's.

Itself the random number generator uses "1+ RAND %6" which means between 1 and 6.

The next class to talk about is person class which when inputted the code "person* person1 = new person();" which points to the ccp file with the code that allows the user to input their name.

The total is gained by using an if statement which is connected to the get number part of the ccps files code. This is because in the source file get number is used 6 times so in this situation I inputted a system where every time a number gets sent it adds 1 to a background counter. Once this hits 6 the next number that is sent is also announced with the total of the last 6 numbers. Then any amount in the "_total" variable is set back to 0 and the background counter I also set back to 0. this means both players can use the code "get number" which also means that writing in the background file will be easier to achieve.

However there is a problem. If the user replayed the game they would get the same numbers every time. So to fix this I added another if loop in the ccp file which states that if another background counter is = or smaller than 1 then start the loop. This loop then asks for a number between 1 – 100. This number then gets inputted into the system. Now every number that is picked will give the user different results in the number generator making the game more random. An additional system can be put in that takes two numbers that both player inputted before hand and adds them together to make a number that will change the random generator increasing its randomness. Or add the #include <ctime> to add a timer so you always get a different number.



Storing to a file

Storing to file is used in storing the names of the players. This is used in the persons ccp file where the system asks for the players' names at the start of the program the system then stores there names in a file that can be accessed in the file with the program in it. next the file stores the total amount of the 6 numbers that each player had. This could but does not include the 6 numbers as the scores of the program are the more important aspect of the game.

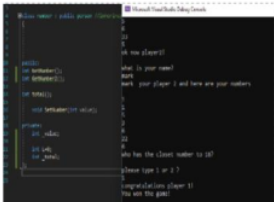
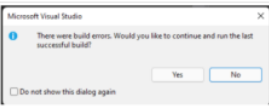
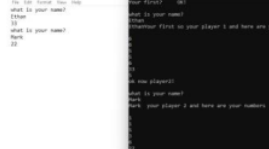
Video of one play though with a screen shot of file

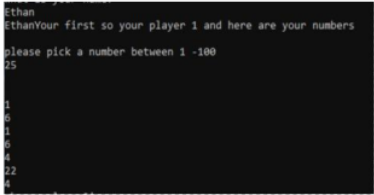
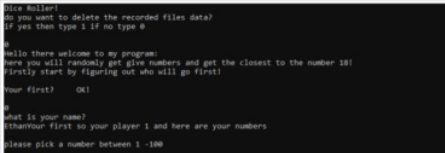
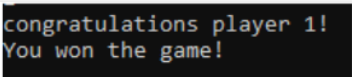
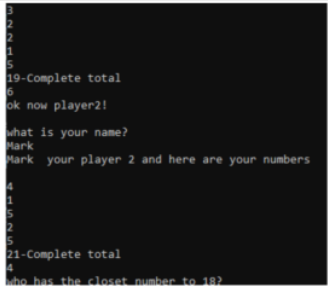
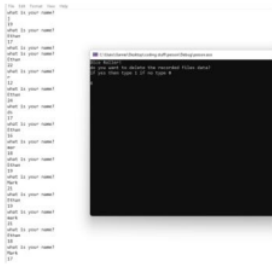
This link takes you to the video that shows my program working.

<https://share.descript.com/view/OVeCz3Tn2vJ>

Screenshot of working code being transferred to the text document

```
score - Notepad
File Edit Format View Help
what is your name?
ethan
17
what is your name?
er
16
```

Test	Proof	Resolved?
Any code that is not needed / wasted code		Multiple lines of code have been deleted when the program does not use these lines. Code deleted from: Number.h - getnumber2 Number.h - total Number.ccp - number::total Source.ccp person2=new person Person.h - _name2
Working code test		Yes. If loop on number.ccp was not closed
File data		File is removing data Yes incident was resolved when changing “out” to “app” on a file transfer line of code in the source file

Test to see if the upgraded random generator is working		Working correctly
Test to see if class text works		Working correctly
Test to see if new header in text.h “winner” works		Working correctly
Test to see if the code is working and correct values are being shown		Correctly showing the total and 6 other numbers that add to that total
Test to see if removing data from the file is a success		System did not delete the code Found problem Code did not have “out” when opening the file s the file did not delete but added more data to the file. File now works as intended

Referencing

[Buckys C++ Programming Tutorials - 27 - Random Number Generator](#)

Advanced software development assignment 1

GRADEMARK REPORT

FINAL GRADE

51 / 100

GENERAL COMMENTS

Instructor

Classes (25): 18
Polymorphism (10): 0

Game (20): 10

Report (35): 18

Testing (5): 2.5

Demo (5): 2.5

Feedback:

=====

The report needed to provide more evaluation of the benefits of OOPs and discuss the OOP concepts.

Code is very confused. There is inheritance between the person and number class but it is not useful. No polymorphism. Game is fairly limited.

You needed to demo the code. No audio.

PAGE 1

PAGE 2

PAGE 3



Comment 1

Sounds like you forgot to seed the random number generator

PAGE 4

PAGE 5

