**CIS 246 – Spring 2020**

|  |  |
| --- | --- |
| **Program:** | **8** |
| **Points:** | **20** |
| **Chapter(s):** | **10** |
| **File(s) to Submit:** | **Party.h, Party.cpp, Player.h, Player.cpp, Driver.cpp (as one zip file)** |

**Summary**

Write a C++ application that creates a party of player objects that could be used in a role-playing game. Print the players in a neatly formatted list.

This assignment builds on Program 7. Finish Program 7 first, and this one will be easier.

In this assignment, you will practice:

* Overloading the compound assignment operator +=

**Description**

Program 8 has exactly the same purpose as Program 7 – it will randomly create Player objects, and then display them. The difference is how this functionality will be implemented.

In this version of the program, the driver creates one Party object. The array or vector that stores the Player objects will be located in the Party class.

The driver randomly creates five objects of a Player class, adds them to the Party object via an overloaded operator, and then calls a print function of the Party class to display the Player information.

**Requirements for the Player Class**

Use the exact same Player class from Program 7, separated into a header and implementation file.

**Requirements for the Party Class**

This class must be separated into a header and implementation file.

**private data members:**

1. Array or vector of Player objects (If using a built-in array, set the size to 5.)

**public functions:**

1. **Print** function – Move the print loop from Program 7’s driver here.
2. **Overloaded += operator function** – This function takes a reference to a Player object as an argument, and then adds it to the array or vector.

**Note** – add the following as the last line of this function:

**return \*this;**

**Extra Credit (Max 5 points)**

In the Party class, overload the **prefix increment operator** to add a Player object with both **default arguments**. Use this operator in the driver when adding a default Player object:

Example: ~~party++;~~ ++party;

**Requirements for the Driver Program**

1. **main**

The main function creates a Party object. Do not create an array or vector here!

Within a counter-controlled loop that executes five times:

* Generate a random number from a set of 3. (Same as program 7)
* Use a **selection statement** to check the random number and create a Player object in one of the three ways listed above. (Same as program 7)
* When the player object is created, add it to the Party object using the **+= compound assignment operator**.

Example code:

Party party;

Player player1; 🡨 created using one of the 3 techniques from Program 7.

party+=player1; 🡨 overloaded compound assignment operator +=

* Display a message to the screen indicating how the Player object was created. (Same as program 7)

Once the loop completes, call the print function of the Party class, which will display the contents of the array or vector, as shown in the sample output.

**General Requirements**

For complete credit, you must:

1. **MEET ALL REQUIREMENTS ACCORDING TO THE INSTRUCTIONS** – Follow the instructions as written for completing this project, even if you [think you] know a “better” way to do something.
2. **INCLUDE COMMENTS** – Include comments in your code. There must be a comment at the top of each source code or header file that includes your name, the assignment number, and a description of the code in that file. There must be comments at each important step in your algorithm that describes that step.
3. **FOLLOW BEST PRACTICES** – Follow best practices in C++ programming, including, but not limited to, appropriate use of private/public, appropriate use of classes and/or header files, sets & gets, white space, alignment, meaningful variable names, naming conventions, using statements, etc. Points will be deducted for sloppy code that is hard to read, even if it works, so pay attention to these details.
4. **SUBMIT ALL FILES BEFORE THE DUE DATE** – Submit a .zip of ONLY source code files to the dropbox for this assignment on Canvas before the due date. Do not submit anything except .cpp and/or .h, within a zip. Do not submit .exe files. Do not submit a folder structure. Do not submit project files from an IDE.

**Sample Run**

Note that this output is the result of random number generation. Your output will vary, but it should include output that explains how each Player object was created, and a neatly formatted list at the end.

Note also that there should be no difference in the output for programs 7 and 8.

A screenshot of a cell phone

Description automatically generated