

CS 120 Project 2

Due on Blackboard by Friday, February 9

For this project, you will design, implement, test, and use a C++ class with overloaded operators.

Design

Think about what your class will represent. What data does it hold and what data types? Does the data have restrictions on its value(s)? What functionalities can it perform? How will you use the functionality? What is the goal of the program? Think about what overloaded operators the user will benefit from most. What will the overloaded operator do? Will it need to be a friend of the class? Could it use helper functions?

Some examples (do not use these):

- A Point class that holds x and y coordinates. Has a method that finds the distance between it and another Point. Overloads the + and - operators.
The program sets a Point to random coordinates (within a reasonable range) and a second Point to the origin (0, 0). The user knows the distance between the two points and has to use four keys (w, s, a, d) to move the Point from the origin to find the other Point. Every time the user moves, the updated distance prints to the console.
- A Baseball_Player class that holds name, number of hits, number of walks, and number of at-bats. Has a method that calculates the batting average. Overloads the << operator to print the statistics and the < and > operators to compare the batting average of two players.
The program simulates an at-bat. Asks user for name of player, looks in file for existing player with matching name (and if not found, creates a default player). Gives pitches and gets user input to swing, stay, or bunt. It prints the current pitch count before each pitch. At the end of the at-bat, it updates the object data and saves it to the file. Compares the player to all the players saved in the file and prints the results to the console.

Requirements

Your class must have at least two private member variables and at least four methods.

At least one method must not be a constructor, destructor, getter, or setter.

Your class must have at least two overloaded operators.

Your class should be declared in a header file (with RMEs) and defined in a .cpp file, both of the same name.

You must test all methods and operators of your class. All testing should be in functions in a file called testing.cpp.

You must use objects of the class (including overloaded operators) to create a fully functioning program in a file called main.cpp.

You will need to submit the header and .cpp files of your class, testing.cpp, main.cpp, and any other files you use in your program to Blackboard.

Grading

The project is out of 70 points.

Design and Style

- 5 pts Does the code compile? Are all files submitted and does each file have the correct parts of the code?
- 5 pts Are there sufficient RME comments to explain what each method accomplishes and what each field represents?
- 5 pts Do the files follow the style guidelines from class? Are they readable? Do the names make sense?
- 5 pts Is there evidence of a well-thought-out design? Does each method have a clear purpose? Is this the best way to implement the class given the functionality goals?

Implementation

- 4 pts Are there at least two fields?
- 8 pts Are there at least four methods?
- 4 pts Is there a method that is not a constructor, destructor, getter, or setter?
- 8 pts Are there at least two overloaded operators?

Testing

- 8 pts Is every method tested (directly or indirectly)?
- 8 pts Does testing cover all possible cases?

Program

- 10 pts Does the program use objects of the class? Is the program fully functional? Does the functionality make sense to use with the class?