

CS 120 Project 2

Due on Blackboard by Friday, February 10

For this project, you will design, implement, and test 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? 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? Write either a diagram or bullet points outlining the answers to these questions.

Some examples:

- a Point class that holds x and y coordinates. Has a method that finds the distance between it and another Point. Can overload the + and - operators.
- an Electric class that holds the price per kilowatt-hour and amount of energy consumed. Has a method that prints the electric bill. Overloads the + and - operators.
- a Movie class that holds the title, director, length, and year of release. Has a method that compares two movies. Overloads the < and > operators to compare release date (or length).
- a Baseball_Player class that holds number of hits, number of walks, and number of at-bats. Has a method that calculates the batting average. Overloads the < and > operators to compare batting average.
- a Pokemon class that holds name, HP, and CP. Has a method to power up. Overloads the > and > operators to compare CP.

Implement

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.

Test

To make sure your class is working the way you designed, include a main.cpp file that has the main function and tests the class methods thoroughly, including overloaded operators.

Grading

The project is out of 60 points.

Design and Style

- 5 pts Are there three files and does each file have the correct code? Does the code compile?
- 5 pts Are there sufficient comments and/or writings 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?
- 6 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?
- 6 pts Are there at least four methods?
- 5 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?