

Assignment 3

NumDays Class	50 pts.
Separate Compilation for NumDays Class	20 pts.
PFAArrayD Modification	30 pts.
TOTAL	100 pts.

Part 1

NumDays Class

Implement a class called NumDays. The class's purpose is to store a value that represents a number of work hours and convert it to a number of days. For example, 8 hours would be converted to 1 day, 12 hours would be converted to 1.5 days, and 18 hours would be converted to 2.25 days. The class should have a constructor that accepts a number of hours, default constructor, member functions for storing and retrieving the hours and days. The class should also have the following overloaded operators:

1. **+ Addition operator.** When two numDays objects are added together, the overloaded + operator should return the sum of the two object's hours members.
2. **- Subtraction operator.** When one NumDays Object is subtracted from another, the overloaded - operator should return the difference of the two object's hours members
3. **++ Prefix and postfix increment operators.** These operators should increment the number of hours stored in the object. When incremented, the number of days should be automatically re-calculated (if you store number of days in member variable).
4. **-- Prefix and postfix decrement operators.** These operators should decrement the number of hours stored in the object. When decremented, the number of days should be automatically re-calculated (if you store number of days in member variable).
5. **<< cout's stream insertion operator.** This operator should cause the days and hours to be displayed in the form: *1.5 day(s) -- 18 hour(s)* .
Overload << as friend function.
6. **< Less operator.** Relational operator returns true if the number of days and hours of the right operand is less than the number of days and hours of the left operand.
7. **> More operator.**
8. **== Equal Operator.**

Demonstrate (test) all the member functions and overloaded operators in main().

Part 2

Separate Compilation for NumDays Class

Modify the program for NumDays Class to use separate compilation. Put class definition in a .h file. Place the implementations in a separate .cpp file. Place main() in the third file. Compile and run your program.

PFArrayD Modification

Start with the source code of the PFArrayD class. Modify the program to use namespaces and separate compilation. Put class definition and other function declarations in one file. Place the implementations in a separate file. Place the demonstration program in a third file.

Create a namespace for the PFArrayD class. Distribute the namespace definition across two files: header and implementation.

To provide access to names in namespaces, you may use local “using” declarations such as:

using std::cout;

or qualify names using the names of namespaces, such as:

std::cout.

You may not use global namespace directives (such as *using namespace std;*) which are not in block and apply to the entire file.