

Proving the Interface

When designing an interface, it's often useful to write a client program, just to try it out. This is called **proving your interface**. You may find that you need additional member functions. Or, you might find that the prototypes for the functions are not exactly what you need to complete your task, and you can change them at this stage.

Here's the `run()` function which will act as the client for your new `Time` type. This is a revision of `H01`, using member functions.

```
int run()
{
    printHeading(); // already written

    Time startTime;
    Time duration;

    // Prompt and read the input
    cout << "    Time: ";
    if (! startTime.read(cin)) { return die(); }
    cout << "    Duration: ";
    if (! duration.read(cin)) { return die(); }

    // Processing section
    Time after = startTime.sum(duration);
    Time before = startTime.difference(duration);

    // Output section
    duration.write(cout) << " hours after, and before, ";
    startTime.write(cout) << " is [";
    after.write(cout) << ", ";
    before.write(cout) << "]" << endl;

    return 0;
}
```

The interface looks OK, so let's go ahead and **implement** the member functions.



This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.