Homework

29

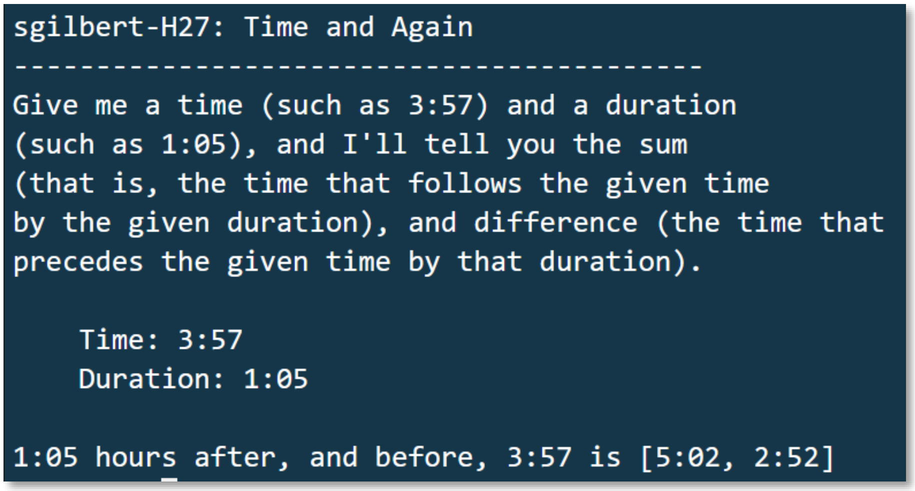
Time and Again

Do you remember H01 where you wrote a program that adds and subtracts time? It asked the user for two input values—a time (like **`3:57`**) and duration (such as **`1:05`**), then, it printed the **sum** (here **`5:02`**) and **difference** (**`2:52`**)[1].

Unlike **`H01`**, for this version you're going to use structures with member functions.

Interactions

The interactions (when using **`make run`**) look almost exactly the same as those for **`H01`**.



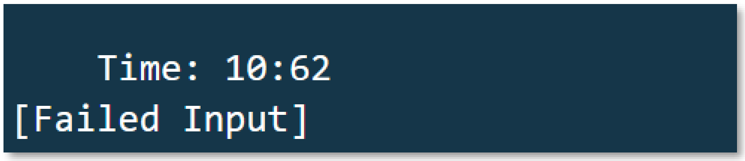
```
sgilbert-H27: Time and Again
------------------------------------------
Give me a time (such as 3:57) and a duration
(such as 1:05), and I'll tell you the sum
(that is, the time that follows the given time
by the given duration), and difference (the time that
precedes the given time by that duration).

    Time: 3:57
    Duration: 1:05

1:05 hours after, and before, 3:57 is [5:02, 2:52]
```

However, if you enter in an **invalid value** for either **`Time`**, the input will fail and print an error message instead. This is possible because the program **does not** directly access the data members of the structure.

[1] Problem from Doug Cooper's Oh! Pascal, 3rd Edition, Chapter 2.



```
   Time: 10:62
[Failed Input]
```

The run() Function

The **`run()`** function is already written for you; you should not change it at all.

Instead, you will:

- Fill in the **file comment** for both the header and implementation files.
- Add your name to the **`STUDENT`** variable in the **`.cpp`** file.
- Add your **`Time`** structure definition to the header file along with the prototypes for the member functions. Don't forget the **`#include`** for **`<iostream>`**
- Implement the member functions in the **`.cpp`** file.

You do not need to change the **`die()`** or **`printHeading()`** functions. They are complete as well. Here are some hints for the member functions.

Input & Output

Both the **`read()`** and **`print()`** functions take a **reference to a stream object** as their only parameter, and return the same parameter. That allows you to write code like this:

```
t.print(cout) << endl;  // returns cout; can insert endl
```

Remember that must **fully qualify** the names **`istream`** and **`ostream`** in the header file.

For the **`read()`** member function follow this plan:

```
Create two local variables, h and m
Use the stream to read into those variables (discard ':')
If the values are out of range then
   Put the stream into a failed state
Otherwise
   Assign h and m to the data members
Return the input stream
```

You can put an input stream (in) into a failed state, like this:

```
in.setstate(ios::failbit);
```

For the **`print()`** member function, refer back to **`H01`** to see the correct formatting.

Sum and Difference

Follow the instructions for **`H01`** to correctly calculate the **sum** and **difference** (after and before). You'll need some local variables, to hold the total number of minutes and the calculated hours and minutes. Once you have them, you return a **`Time`** object like this:

```
return Time{hoursAfter, minutesAfter};
```

Check your work with **`make test`**. If you want to run the program interactively, use **`make run`**. Use **`make submit`** to turn in your assignment, and check on the Discussion Board, or at my office hours. if you are having problems. Make sure you start early enough as well.