Solution: Fix the Code

This lesson provides solution to a challenge given in the previous lesson.

WE'LL COVER THE FOLLOWING

- Solution
- Solution explanation

Solution

Here is the fixed code, that will produce the desired output upon entering 06:09 15:2 as input.

```
import std.stdio;
import std.string;
import std.exception;

/* Reads the time as hour and minute after printing a
 * message. */
void readTime(string message,
              out int hour,
              out int minute) {
    write(message, "? (HH:MM) ");

    readf(" %s:%s", &hour, &minute);

    enforce((hour >= 0) && (hour <= 23) &&
            (minute >= 0) && (minute <= 59),
            "Invalid time!");
}

/* Returns the time in string format. */
string timeToString(int hour, int minute) {
    assert((hour >= 0) && (hour <= 23));
    assert((minute >= 0) && (minute <= 59));

    return format("%02s:%02s", hour, minute);
}

/* Adds duration to start time and returns the result as the
 * third pair of parameters. */
void addDuration(int startHour, int startMinute,
                 int durationHour, int durationMinute,
                 out int resultHour, out int resultMinute) {
```

```
                 out int resultHour, out int resultMinute) {
    resultHour = startHour + durationHour;
    resultMinute = startMinute + durationMinute;

    if (resultMinute > 59) {
        ++resultHour;
    }
}

void main() {
    int startHour;
    int startMinute;
    readTime("Start time", startHour, startMinute);

    int durationHour;
    int durationMinute;
    readTime("Duration", durationHour, durationMinute);

    int endHour;
    int endMinute;
    addDuration(startHour, startMinute,
                durationHour, durationMinute,
                endHour, endMinute);

    writefln("%s hours and %s minutes after %s is %s.",
             durationHour, durationMinute,
             timeToString(startHour, startMinute),
             timeToString(endHour, endMinute));
}
```

Program to calculate end time

Solution explanation

The `assert` failure after entering 06:09 and 15:2 takes us to the following line:

```
string timeToString(int hour, int minute) {
    assert((hour >= 0) && (hour <= 23));
    // ...
}
```

For this `assert` check to fail, this function must have been called with an invalid hour value.

The only two calls to `timeToString()` in the program do not appear to have any problems:

```
writefln("%s hours and %s minutes after %s is %s.",
         durationHour, durationMinute,
         timeToString(startHour, startMinute),
         timeToString(endHour, endMinute));
         timeToString(endHour, endMinute));
```

A little more investigation should reveal the actual cause of the bug: the hour and minute variables are swapped when reading the start time:

```
readTime("Start time", startMinute, startHour);
```

That programming error causes the time to be interpreted as 09:06 and incrementing it by duration 15:2 causes an invalid hour value. An obvious correction is to pass the hour and minute variables in the right order:

```
readTime("Start time", startHour, startMinute);
```

In the next lesson, you will find another challenge related to the code given above.