
Tutorial 3 of 10 - Due Saturday Feb. 4th, 11:59 pm

- No late tutorials will be accepted.
- If there are specific instructions for making a function, please follow them exactly. That means that
 - function names
 - function return types
 - parameter types and order

should all be **EXACTLY** as described. If the script can't read it, you will receive 0 for that part.

- Your **Tutorial 3** code will be marked by a Python script running on the Gradescope server. You are being given a similar script so that you may make sure your code runs correctly.
-

1 Download and Submission Instructions

Download [tutorial3.zip](#) and unzip it into your workspace. You will find the test script [t3test.py](#) and a [tutorial3](#) folder. In the [tutorial3](#) folder you will find [test1.cc](#), [test2.cc](#) and [defs.h](#).

The script [t3test.py](#) is provided for convenience, so that you can have immediate feedback on your code. However, your actual mark will be given by the Gradescope server.

You should add to the [tutorial3](#) folder the header and source files for three classes from Assignment 1, [Date](#), [Reservation](#), and [Room](#) as well as a [Makefile](#). The [Makefile](#) should have (at minimum) commands to [make all](#), which makes [test1](#) and [test2](#) executables, and [make clean](#), which removes all executables and object files. Once you have written the classes and completed your tests, submit this tutorial to Gradescope. The Gradescope server is flexible in how you submit. You may zip the folder, zip the files, or submit the folder itself or individual files. What follows are one set of instructions that will work on Gradescope.

Zip your “tutorial3” directory into a file “tutorial3.zip”. If you are using the VM you must do this from the command line. Open a terminal in the folder that contains “tutorial3”. Use the command [zip -r tutorial3.zip tutorial3](#). **Note the -r. It will not work without -r.** This will zip your file and update it if you change the contents. Submit “tutorial3.zip” to Gradescope by the deadline. You will receive your mark immediately, and you may submit as many times as you like. You may also submit late tutorials, but there will be a penalty (somewhere between 10% and 50%). Presently the server is set to receive tutorials up to a week late.

2 Testing Your Tutorial With [t3test.py](#)

[t3test.py](#) is a test script that is very similar to the script that is being run on Gradescope (there might be slightly different input or different even tests). So the mark you see here should be close to the mark you will receive on Gradescope.

To run [t3test.py](#), open a terminal in the directory that contains your “tutorial3” folder. You may have to make the script executable, so type [chmod +x t3test.py](#). To have the script test your code, run [./t3test.py](#). Your output and mark should be printed to the console.

Tutorial 3 of 10 - Due Saturday Feb. 4th, 11:59 pm

3 Learning Outcomes

This tutorial will test the application logic in two of the classes from Assignment 1 (note there are three classes but `Date` is not tested explicitly).

4 Instructions

4.1 Overview

In this tutorial you will include the `Date` (Section 5.3), the `Reservation` (Section 5.4), and the `Room` (Section 5.5) classes from Assignment 1. There are two test files provided, `test1.cc` and `test2.cc` that will test the `overlaps` and `isMatch` functions from those classes (there is some overlap with the marks earned in Assignment 1). You must provide a `Makefile` that compiles `Date`, `Reservation`, and `Room` into object files and compiles `test1.cc` and `test2.cc` into executables linking to the appropriate object file. As usual the test script, `t3test.py` is provided.

Note that this *only* tests the `isMatch` and `overlaps` functions. That means you do not have to include the `lessThan` or `equals` functions or any other unrelated functions to get full marks (though you can include them if you wish).

4.2 Date Class

Complete Section 5.3 in Assignment 1. Link the object file `Date.o` to any appropriate executable.

4.3 Reservation Class

Complete Section 5.4 in Assignment 1. Link the object file `Reservation.o` to any appropriate executable.

4.4 Room Class

Complete Section 5.5 in Assignment 1. Link the object file `Room.o` to any appropriate executable.

4.5 Makefile

Your Makefile should compile three object files, `Date.o`, `Reservation.o`, and `Room.o`. It should then link them to the appropriate executable. You should compile `test1.cc` and `test2.cc` into the executables `test1` and `test2` respectively.

In addition your Makefile should contain an `all` command that creates the `test1` and `test2` executables and a `clean` command that removes all executables and object files.

4.6 t3test.py

Run this python script to test the functions described above from the `Reservation` and `Room` classes.