# Lab 1 – Classes

Object Oriented Programming using C++

Scenario: You work for a software firm that is creating flight scheduling software. You are tasked to write code for flight schedules that include flight source, destination, and departure time information. A teammate of yours has created an expected interface for your code. You are asked to implement the class definitions to support those operations.

## Task 1. The `Time` class:

Create a class called `time` to store time in 24-hour format with fields `hour` and `minute`. Your code should support the following set of statements:

```
1 Time t1; //t1 initialized to 0 hour 0 minute, i.e., 12:00
2 Time t2 (3); //t2 initialized to 3 hour 0 minute, i.e., 03:00
3 Time t3 (5,10); //t3 initialized to 5 hour 10 minute, i.e., 05:10
4 int minute = 10;
5 cout << t1;
6 cout << t2 << t3;
7 cout << t2+t3;
8 t1 = t3+minute;
```

In t2, you only set `hour` field from the parameters. Note that the `hour` field can only support numbers in the range 0 to 23 and the `minute` field can only support numbers in the range 0 to 59. Be sure to include getters and setters.

### Grading information:

[LP] Support lines 1 – 4.

[HP] Support all 8 lines.

## Task 2. The `Flight` class:

Create another class called `Flight` with the fields `src` (string type), `dest` (string type), and `departureTime` (`Time` type). Note that the field `departureTime` is of type `Time`. Your code should support the following set of statements (Hint: Remember the "has-a" relationship):

```
1 string src = "SFO";
2 string dest = "LAX";
3 Flight f1(src, dest); //Note that there should be no default value of src and dest
4 f1.setDepartureTime(13, 45); //Sets the departureTime field to 13:45
5 cout << f1; //Outputs flight information in a suitable format
6 Time t(6, 30);
7 t1.setDepartureTime(h); //Sets the departureTime field to values from t
8 cout << f1;
```

### Grading information:

[LP] Support lines 1 – 4.

[HP] Support all 8 lines.

### Task 3. Thoughts and Documentation:

Discuss the following questions for Task 1:

1. [LP] What constructors did you need to use? Is one constructor enough?
2. [HP] What is the difference between lines 5 and 6? Do you need special implementation to support line 6?
3. [HP] Do you need to overload the assignment operator to support any the above operations?
4. [HP] What did you need to do to support line 8? What would happen if you tried `cout << minute + t3` instead?

Discuss the following questions for Task 2:

1. [LP] What constructors did you need to use? Is one constructor enough?
2. [HP] Do you need to overload the assignment operator for the `Time` class to support line 7? Why or why not?
3. [HP] Did you need to do any input validation to support lines 3 and 4? What about line 7?

Include your discussions in a file called `Notes`.

**VERY IMPORTANT:** Also include ALL the references you used for this lab in the `Notes` file. Failure to cite your sources counts as an act of academic dishonesty and will be taken seriously without zero tolerance.

## Specifications

All tasks have components labeled [LP] and [HP]. If you complete ALL the LP components satisfactorily, you will receive a grade of "low pass" on the lab. If you complete ALL the LP components and the following HP components satisfactorily, you will receive a grade of "high pass":

- Task 1 HP
- Task 2 HP
- Task 3 answer 4/5 discussion questions satisfactorily

If you do not meet the criteria for a "low pass", the submission will be marked as "revision needed".

### What to submit:

Your final submission will need to have the files as follows:

- `Time.h`
- `Time.cpp`
- `Flight.h`
- `Flight.cpp`
- `lab1-cmpe126.cpp`
- `Notes`

The statements provided in the tasks should be in `lab1-cmpe126.cpp`. Feel free to add more lines to test your implementation. Beautify the output as you like it.

**NOTE: You can look for help on the Internet but refrain from referencing too much. Please cite all your sources in your `Notes` file.**

## When to submit:

Submit your lab before **Thursday, February 8th, 11:59pm**. You are strongly advised to submit before Friday, February 2nd, 11:59pm.

When you submit your assignment, you automatically agree to the following statement. If you do not agree, it is your responsibility to provide the reason.

"*I affirm that I have neither given nor received unauthorized help in completing this homework. I am not aware of others receiving such help. I have cited all the sources in the solution file.*"