EXERCISE 3

The Time Class

The objects of class Time have exactly two attributes, minutes and hours. Both of these attributes are integers. However, there is an important restriction on minutes: it must be between 0 and 59. If you go over 59 minutes, you are supposed to increase the hours attribute instead.

Open up the Python interactive shell and type in the following two statements:

```
>>>from ctime import Time
>>>t = Time(2,30)
```

The from syntax is an alternate way to import a module. It allows you to avoid writing the module name in front of everything.

You should fill out the tables below, just as you did in the second lab on assignment statements. In the first table you are to determine the value of the expression or a command. If it is a command, you should just write either none or error (if the command causes an error). In the second table, you should guess the variable or literal that makes the second column true.

Fill in the tables as directed.

Expression	Calculated
t.minutes	
t.hours	
t.days	
s = t	
s.minutes	
t.minutes = 20	
s.minutes	
s.minutes = 60	

Expression	Calculated	Missing
x = Time(2, ☐)	x.minutes == 35	
y = Time(☐,10)	y.hours == 5	
y.☐ = 25	y.hours == 5	
z = ☐	id(z) == id(x)	
z.minutes = ☐	x.minutes == 15	
z = ☐	id(z) == id(y)	
z.hours = ☐	y.hours == 3	
w = str(☐)	w == '2:15'	

The Function add_time(time1,time2)

In the file lab03.py, you will see a stub for a function called add_time. Note that this function creates a new Time object, so this function will need to call the constructor for Time.

Implement the function specified below in lab03.py. Because this function creates a new Time object, it will need to call the constructor for Time.

```
def add_time(time1, time2):
    """
    Returns: The sum of time1 and time2 as a new Time object

    Example: Sum of 1hr 59min and 1hr 2min is 3hr 1min
    DO NOT ALTER time1 or time2, even though they are mutable

    Parameter time1: the starting time
    Precondition: time1 is a Time object

    Parameter time2: the time to add
    Precondition: time2 is a Time object
    """
```

This time we do not need you to provide any test cases. However, it might be a good idea to test the function before attempting to get credit for it.