Lab 22
One More Time

Objective:

Write a class converts 24-hour time to 12-hour time
- First download the driver and put it in your project
  - DO NOT ALTER THE DRIVER!
- Write a class file called **TimeFormatException** that inherits from Exception
- Create the following constructors
  - Default – calls the parent's constructor and says something like "This is an incorrect time"
  - One that has a string parameter that passes that message to the parent's constructor
- Write a class file called **Clock24** that DOES NOT HAVE a main method
- Some of the attributes of Clock24 are
  - Hours
  - Minutes
  - isAM – whether or not it is AM or PM
- Create the following Constructors
  - Default – sets everything to default values
    - Hours = 0 – we assume that the 24th hour is 0
    - Minutes = 0
    - isAM = true
  - One that has the parameters
    - Hours
    - Minutes
- Accessors and Mutators for each variable
  - MAKE SURE THE MUTATORS CHECK FOR VALID VALUES!
- Create the following Methods
  - setTime – this method will take in an hours value and a minutes value and if those values are valid then they are converted to 12-hour time (Don't forget to change from AM to PM and visa versa), and finally are set to the classes instance variables hours and minutes. If they are not valid such as if they were negative, they throw a TimeFormatException.
  - setTime – this overloaded has a single string parameter formatted as "hours:minutes", so for instance "17:45". This string value is to be broken down into separate hours and minutes and then those values are checked just like the other set time
    - Hint: For the hours part, use a loop to look at each character in the string adding those characters to another string until it reaches the ":". Once it does reach ":" parse it. Same thing for minutes but after the ":", and it ends when it reaches the end of the string.
    - In the parsing process you need to handle the general Exception that comes when the user enters in something that is not a number. Use a try catch for this.
  - printTime – prints out the 12-hour time formatted as hours:minutes time of day, so "1:45 PM"

Example Dialog:

```
Enter a time in the 24-hour notation or enter "Exit" to quit
2:45
2:45 AM
Enter a time in the 24-hour notation or enter "Exit" to quit
14:45
2:45 PM
Enter a time in the 24-hour notation or enter "Exit" to quit
-1:56
EXCEPTION!: The hour is incorrect
2:45 PM
Enter a time in the 24-hour notation or enter "Exit" to quit
asdf;lasdf
EXCEPTION!: Wrong format
EXCEPTION!: The hour is incorrect
2:45 PM
Enter a time in the 24-hour notation or enter "Exit" to quit
Exit
Goodbye!
```

Lab Report Questions:

1. Is it possible to catch multiple exceptions?
2. What is the difference between "throw" and "throws"?

Finally:
Upload TimeFormatException and Clock24 file to the dropbox