

CS100J Lab 05. Writing functions Spring 2005

Name _____ NetId _____

Section time _____ Section instructor _____

The purpose of this lab is to give you practice with developing the bodies of methods. At the same time, this lab will give you practice with Strings. This lab will help you prepare for the prelim.

The methods will, among other things, change a time in a String into a different format. The time comes in four formats:

24-hour-string:	"hours>:<minutes>" <hours> is in range 0:23 and <minutes> is in range 0..59 Examples: "4:20" "13:00" "23:59"
AM-PM-string:	"<hours>:<minutes>AM" or "<hours>:<minutes>PM" <hours> is in range 0:11 and <minutes> is in range 0..59 Examples: "4:20AM" "3:00PM" "11:59PM" "0:0"
24-hour-verbose	Example: "4 hours and 20 minutes" Example: "23 hours and 5 minutes" Note: exactly one blank between each of the pieces.
24-hour-correct	Exactly like the 24-hour-verbose format except that it is gramatically correct. So, instead of "1 hours and 20 minutes" it reads "1 hour and 20 minutes" and instead of "0 hours and 1 minutes" it reads "0 hours and 1 minute".

Note also that if a String *s* contains only digits (not even blanks), then the function call

`Integer.parseInt(s)`

yields the integer represented by *s*. For example, `Integer.parseInt("345")` is 345.

What to do

When you are finished writing and testing the functions given below, show them to your lab instructor. We suggest that you save both .java files that you created on a USB storage key or else mail them to yourself. If you do not have time to finish this lab assignment in the allotted time, then show the completed lab to your instructor the next week.

First, create in DrJava a new class `Methods` and saving it appropriately. To simplify your work, you can download file [Methods.java](#) from here or from the course website; it has all the methods in it already. The function bodies have a return statement just so that the class will compile.

Second, create a JUNit test class, as usual.

Third, for each function given below, in turn, do the following:

1. Write the body of the method. BEFORE YOU WRITE THE BODY, IF THE METHOD DOES NOT HAVE A SUITABLE SPECIFICATION IN A JAVADOC COMMENT, WRITE ONE.
2. Think about what test cases would be necessary for you to know that the method is correct. Create a `testX` method in class `MethodTester`, and insert those test cases
3. Test the method.

Advice: If you need help, ask the TA or a consultant. Don't waste time! Some pondering is necessary, but don't overdo it.

Guidelines: You should check the javadoc comments (click button javadoc) to make sure they are right. Always keep your program indented properly, and don't let lines get too long. If scrolling is necessary, then fix it so that it is not necessary. Everything should be readable.

```
/** s is a time in the 24-hour-string format; Return the same time in 24-hour-verbose format */  
public static String toVerbose(String s) {  
    return "";  
}
```

```
/** s is a time in the 24-hour-string format; Return the same time in AM-PM format. */  
public static String ToAMPM(String s) {  
    return "";  
}
```

```
/** s is a time in 24-hour-string format; Return the same time in 24-hour-correct time */  
public static String timeToCorrect(String s) {  
    return "";  
}
```

```
/** s is a time in AM-PM-string format; Return the same time in 24-hour-string format.*/  
public static String eliminateAMPM(String s) {  
    return "";  
}
```

```
/** s is a time in EITHER the 24-hour-string format OR the AM-PM-string format;  
Return the same time in the 24-hour-string format*/  
public static String removeAMPM(String s) {  
    return "";  
}
```

```
/** s is a time in the 24-hour-string format;  
Return the time as the number of minutes. E.g. "14:35" is 14*60 + 35.*/  
public static int timeInMinutes(String s) {  
    return 0;  
}
```

```
/** s is a time in the AM-PM-string format;  
Return the time as the number of minutes. E.g. "14:35" is 14*60 + 35.  
See if you can write the body as a single return statement. */  
public static int AMPMtimeInMinutes(String s) {  
    return 0;  
}
```

```
/** = time s1 < time s2;  
Precondition: s1 and s2 are in either 24-hour-string format or AM-PM-string format  
See if you can write the body as a single return statement. */  
public static boolean isLess(String s1, String s2)  
    return false;  
}
```