

Homework assignment 07

Complete the following programs for this week's homework.

Programming assignments

Creating a test class (10 points)

Chapter 10 discusses class encapsulation and abstraction, meaning that you don't need to know how a class works internally in order to use it. For this problem, you'll write a test class using following **MyTimer** class:

Class Name
- startTime: java.util.Date
+ MyTimer() + start(): void + stop(): void + reset(): void + getElapsedTimeMillis(): int + getElapsedTimeSeconds(): double + getElapsedTimeMillis(java.util.Date startTime, java.util.Date endTime): int <<static>> + getElapsedTimeSeconds(java.util.Date startTime, java.util.Date endTime): int <<static>>

- Write a test class
 - That creates a timer instance
 - Starts the timer
 - Adds up all the numbers from 0-99, printing the elapsed time in milliseconds on each iteration
 - Stops the timer

Note:

- *Do not write the MyTimer class!* This is an example showing you don't need to know everything about the class to use it
- **It will not compile** (because there is no MyTimer class!) Don't worry about the red squiggles. You don't have

a MyTimer class, so you won't actually be able to run this program.

- This is good practice for the exam, since you won't be able to run your code. Use this to practice.

Static Sanitization (20 points)

Whenever you write a public-facing website or service that allows users to enter data, that data needs to be checked to make sure weird stuff doesn't happen to your code or database. That's called *sanitizing* your data.

Write a class named **Sanitizer** that includes a *static* method **sanitize()** that takes a String as input, removes any of the following characters and returns the sanitized String.

- Characters to remove: "#!*\$%&" (not the quotes. or this sentence)

Then write a test class that will call the static sanitize method with an "unsanitized" string and then print out the returned "sanitized" string.

Note:

- The Sanitizer class should be properly created so one cannot make an instance of it (remember the modifiers to use on the constructor to prevent this. And look at chapter 9)
- Look at the section on the String class in Chapter 10. You should be able to do this without writing a single loop

Comma Separated Scores (30 points)

Comma Separated Value, or CSV, is a common format for table data stored as a file or transmitted. The contents of a csv file at its simplest form could something like this:

```
78.0,92.5,88.8,96.4
```

Write a program that asks the user to enter a list of comma separated scores of *double* type and returns the average of the given scores. Example input and output:

```
please enter the string of scores, separated by commas:
78.0,92.5,88.8,96.4
Average score is 88.92500000000001
```

Note: - The line of scores entered should not have any spaces

Remember that comments and good naming conventions factor into your grade! Don't forget.

