

Assignment 05 – Ten Methods

You must work in **alone** on this assignment. Do not use any Java language features we have not cover so far in this course.

Assignment Objectives

After completing this assignment the student should be able to:

- Write methods that have no parameters no return value
- Write methods that have parameters
- Write methods that have a return value

Assignment Requirements

For this assignment you are given the following files:

Assignment05.java (you must complete this file)

Problem Description and Given Info

Within the the **Assignment05.java** file, you must define the following static methods. In the **main** method, you may write any code that wish to test the methods you have been asked to define.

1) Write (define) a static method named **displayGreeting**, that takes no arguments and returns no value. When this function is called, it should print the text **"Hello, and welcome!"**.

Example:

displayGreeting() will print **Hello, and welcome!**

2) Write (define) a static method named **displayText**, that takes a single **String** argument and returns no value. When this function is called, it should print the value of the argument that was passed to it.

Examples:

displayText("Hello")	will print Hello
displayText("123")	will print 123
displayText("abc" + "123")	will print abc123

3) Write (define) a static method named **printTotal**, that takes three **int** arguments. When this function is called, it should print the sum of the three arguments passed to it. This function should return no value.

Examples:

printTotal(0, 0, 0)	will print 0
printTotal(0, 1, 3)	will print 4
printTotal(100, 23, 2)	will print 125

4) Write (define) a static method named **getTotal**, that takes three **int** arguments. When this function is called, it should return the sum of the three arguments passed to it as an **int**.

Examples:

getTotal(0, 0, 0)	will return 0
getTotal(0, 1, 3)	will return 4
getTotal(100, 23, 2)	will return 125

5) Write (define) a static method named **getAverage**, that takes three **int** arguments. When this function is called, it should return the average of the three arguments passed to it as a **double**.

Examples:

getAverage(0, 0, 0)	will return 0.0
getAverage(0, 1, 3)	will return 1.33333...
getAverage(100, 13, 7)	will return 40.0

6) Write (define) a static method named **averageLength**, that takes three **String** arguments. When this function is called, it should return the average length (number of characters) of the **String** arguments passed to it as a **double**.

Examples:

averageLength("a", "abc", "ab")	will return 2.0
averageLength("hello", "goodbye", "monday")	will return 6.0
averageLength("wednesday", "tuesday", "monday")	will return 7.33

7) Write (define) a static method named **lengthOfShortest**, that takes two **String** arguments. When this function is called, it should return the length (number of characters) of the shortest **String** argument passed to it as an **int**.

Examples:

lengthOfShortest("abc", "ab")	will return 2
lengthOfShortest("hello", "goodbye")	will return 5
lengthOfShortest("thursday", "friday")	will return 6

8) Write (define) a static method named **stringOfStars**, that takes one **String** argument. When this function is called, it should return a **String** of asterisks (*) that is the same length as the string argument passed to it.

Examples:

stringOfStars("abc")	will return "****"
stringOfStars("Hello, world!")	will return "*****"
stringOfStars("0123456789")	will return "*****"

9) Write (define) a static method named **maxStringOfStars**, that takes two **String** arguments. When this function is called, it should return a **String** of asterisks (*) that is the same length as the longest string argument passed to it.

Examples:

maxStringOfStars("a", "abc")	will return "****"
maxStringOfStars("hello", "goodbye")	will return "*****"
maxStringOfStars("thursday", "friday")	will return "*****"

10) Write (define) a static method named **midStringOfStars**, that takes three **String** arguments. When this function is called, it should return a **String** of asterisks (*) that is the same length as the string argument with the length that would be in the middle if the lengths of the arguments were arranged in ascending order.

Examples:

midStringOfStars("a", "abc", "ab")	will return "***"
midStringOfStars("hello", "goodbye", "yes")	will return "*****"
midStringOfStars("123456", "12", "1234")	will return "*****"

Method Template

Here is a template that you may use or refer to when defining your methods. All of your ten methods will follow this template; you must provide the components designated by the angle brackets (< >)

```
static <returnType> <methodName>(<parameters>) {  
    <methodBody>  
}
```

What to turn in

For this assignment you must upload the following files by the due date.

Assignment05.java

Any assignment submitted **less than 24 hours after the posted due date** will have **10 points deducted**.

Any assignment submitted **more than 24 hour after the posted due date** will receive **a zero in the grade book**.

Grading Rubric

Criteria	Points
<i>All required files are submitted</i>	10
Each file includes a comment header with the following information: <ul style="list-style-type: none"> • CSE 110 : <Class #> / <meeting days and times> • Assignment : <assignment #> • Author : <name> & <studentID> • Description : <of the file contents> • Partial credit can be awarded 	
<i>Code is neat and well organized</i>	10
<ul style="list-style-type: none"> • Good naming conventions for all identifiers • Good use of whitespace • Descriptive comments • Methods are small and flat • Partial credit can be awarded 	
<i>Code compiles with no syntax errors</i>	20
<ul style="list-style-type: none"> • No Partial credit can be awarded • No credit will be awarded for structure or logic if your code does not compile 	
<i>Code passes structure tests</i>	30
<ul style="list-style-type: none"> • Partial credit can be awarded based on percentage of tests passed • Link to Structure Tests - https://repl.it/@PM_CSE_ASU/CSE110TenMethodsStructureTests 	
<i>Code passes logic tests</i>	30
<ul style="list-style-type: none"> • No credit will be awarded for logic if your code does not pass all structure tests • Partial credit can be awarded based on percentage of tests passed • Link to Logic Tests - https://repl.it/@PM_CSE_ASU/CSE110TenMethodsLogicTests 	
TOTAL	100