

Lab Eight

Objectives:

- To create a module-based program
- To create and call methods that have parameters.
- To use overloading.
- To generate random characters
- To create a summary

Note: This lab will cover concepts from Chapter Six, Sections 6.1-6.11,

Assignment:

This assignment has two parts.

Part I

You will create a new program called **LastNameFirstInitialLab8**.

Add the necessary import statement(s), and the comments for the integrity statement, lab number, and programmer name and course name in the correct places. As always, it must also print out the lab #, programmer's name, course name and section #, and the program information.

The program that will have 4 methods. These are:

Method 1: Name: getWholeNumber().

Purpose: Will ask a user to enter a whole number and return that number

Attributes: Value-returning, no parameters

Method 2: Name: getRealNumber().

Purpose: Will ask a user to enter a real number and return that number

Attributes: Value-returning, no parameters

Method 3: Name: printCharacter()

Purpose: Will accept the whole number returned from the getWholeNumber() method as its parameter which will be assigned to a character type variable called letter. The two variables will be used in a printline as shown in the sample.

Attribute: Void, one parameter, the whole number from the getWholeNumber() method

Method 4: Name: printCharacter()

Purpose: Will accept the real number returned from the getRealNumber() method as its parameter which will be assigned to a character type variable called letter. The two variables will be used in a printline as shown in the sample.

Attribute: Void, one parameter, the real number from the getRealNumber() method

Hint: You will need to use the round() method from the Math library to round the number before typecasting it to an integer.

Notes:

- The use of the printCharacter() methods are examples of overloading (Section 6.8).
- The use of the same variable called letter in two different methods demonstrates scope rules (Section 6.9)
- You can see an ASCII chart in your textbook, Appendix B.
- Your program must look like the sample using the same verbiage. I suggest that you look at the samples before you begin so you will know what this program will look like.
- Do not go above the objectives of the assignment.
- The grading rubric is at the end of the assignment.

Bonus (5 additional points):

Add a void method called **myMath()** that will demonstrate the use of the math class methods: **decrementExact**, **min**, **log**, and **floor**.

This method will accept the whole number returned from the getWholeNumber() method and the real number returned from the getRealNumber() method as its parameters.

- For the [decrementExact\(\) method](#), use the whole number.
- Before using the min method, call the getWholeNumber() method assigning it to a variable called secondNumber. You will then compare the two whole numbers, the parameter and secondNumber.
- For the log() and floor() methods, use the real number.

The method will print the results as shown in the sample. The Math class methods can be called within the println.

Part II

You will write a summary of the project. This is a short analysis of the problem *written in Word*. It will include the:

- Analysis of the Problem
- Problem Definitions
- Input
- Output
- Formulas
- Problems
- Successes
- Time Required to Complete Project
- Unusual happenings
- Which team member worked on which part(s) of the program

Call the summary **Summary**

Hint: Go to the **Course Modules > Help Files > How to Write a Summary**

To Submit this lab:

In Canvas, upload the Word file called **Summary.docx** and the Java file called **LastNameFirstInitialLab8.java** to the assignment. (Do not create and upload a zip file! I will be looking for the 2 files).

SAMPLE RUN SCREEN #1

Values shown in **bolded blue** are variables.

Lab Eight
Christine Kikuchi
CSC 130, Sec #

This program had 4 methods.

The 1st method is called getWholeNumber().

It gets an integer from a user and returns that value.

The 2nd method is called getRealNumber().

It gets a real number from a user and returns that value.

The program has an example of overloading.

The 3rd method is called printCharacter().

It uses the value from the getWholeNumber() method and displays the number and its character equivalent.

The 4th method is also called printCharacter().

It uses the value from the getRealNumber() method and displays the number and its character equivalent.

In order to be able to print from the ASCII table, the program limits the values that the user can enter.

*** You will enter values in the range of 33-126 for whole numbers and 33.0-126.0 for real numbers ***

Enter a whole number, one that does not have a decimal point: 46

Enter a real number, one that has a decimal point: 123.465

The number is **46** and the character for this is a(n) .

The number is **123.465** and the character for this is a(n) {

SAMPLE RUN SCREEN #2

Values shown in **bolded blue** are variables.

Lab Eight
Christine Kikuchi
CSC 130, Sec #

This program had 4 methods.

The 1st method is called getWholeNumber().

It gets an integer from a user and returns that value.

The 2nd method is called getRealNumber().

It gets a real number from a user and returns that value.

The program has an example of overloading.

The 3rd method is called printCharacter().

It uses the value from the getWholeNumber() method and displays the number and its character equivalent.

The 4th method is also called printCharacter().

It uses the value from the getRealNumber() method and displays the number and its character equivalent.

In order to be able to print from the ASCII table, the program limits the values that the user can enter.

*** You will enter values in the range of 33-126 for whole numbers and 33.0-126.0 for real numbers ***

Enter a whole number, one that does not have a decimal point: 97

Enter a real number, one that has a decimal point: 77.01

The number is **97** and the character for this is a(n) **a**

The number is **77.01** and the character for this is a(n) **M**

BONUS SAMPLE RUN SCREEN #1

Values shown in **bolded blue** are variables.

Lab Eight
Christine Kikuchi
CSC 130, Sec #

This program had 5 methods.

The 1st method is called `getWholeNumber()`.

It gets an integer from a user and returns that value.

The 2nd method is called `getRealNumber()`.

It gets a real number from a user and returns that value.

The program has an example of overloading.

The 3rd method is called `printCharacter()`.

It uses the value from the `getWholeNumber()` method and displays the number and its character equivalent.

The 4th method is also called `printCharacter()`.

It uses the value from the `getRealNumber()` method and displays the number and its character equivalent.

An additional method called the `myMath()` will display working with some `Math` class methods.

In order to be able to print from the ASCII table, the program limits the values that the user can enter.

*** You will enter values in the range of 33-126 for whole numbers and 33.0-126.0 for real numbers ***

Enter a whole number, one that does not have a decimal point: 37

Enter a real number, one that has a decimal point: 96.23

The number is **37** and the character for this is a(n) **%**

The number is **96.23** and the character for this is a(n) **`**

The argument of **37** decremented by one is **36**

Enter a whole number, one that does not have a decimal point: 12

The maximum of the two numbers **37** and **12** is **37**

The logarithm of **96.23** is **4.567**

The floor of **96.23** is **96.000**

BONUS SAMPLE RUN SCREEN #2

Values shown in bolded blue are variables.

Lab Eight
Christine Kikuchi
CSC 130, Sec #

This program had 5 methods.

The 1st method is called getWholeNumber().

It gets an integer from a user and returns that value.

The 2nd method is called getRealNumber().

It gets a real number from a user and returns that value.

The program has an example of overloading.

The 3rd method is called printCharacter().

It uses the value from the getWholeNumber() method and displays the number and its character equivalent.

The 4th method is also called printCharacter().

It uses the value from the getRealNumber() method and displays the number and its character equivalent.

An additional method called the myMath() will display working with some Math class methods.

In order to be able to print from the ASCII table, the program limits the values that the user can enter.

*** You will enter values in the range of 33-126 for whole numbers and 33.0-126.0 for real numbers ***

Enter a whole number, one that does not have a decimal point: 65

Enter a real number, one that has a decimal point: 96.5

The number is **65** and the character for this is a(n) **A**

The number is **96.5** and the character for this is a(n) **a**

The argument of **65** decremented by one is **64**

Enter a whole number, one that does not have a decimal point: 54

The maximum of the two numbers **65** and **54** is **65**

The logarithm of **96.50** is **4.570**

The floor of **96.50** is **96.000**