

Lab 01: Java Fundamentals I .

20 pts

Distribute on September 2, 2025

Due before September 14, 2025(Sunday) at 11:59 P.M. (one minute before 12:00 midnight)

Learning Outcomes (CLO vs SO Mapping)

- Recognize the software and hardware components of a computer system (1)vs(6)
- Utilize Java syntax in fundamental programming algorithms (3)vs(1)
- Recognize and apply the various input and output devices in programming (4)vs(2)

Requirements

For this lab, you will create a small Java program to practice

- declaring, initializing, and assigning values to variables
- knowing the range for various data types
- properties of arithmetic and type-cast operators
- evaluating arithmetic expressions
- displaying output to the console window
- Use Scanner class reading input from the keyboard
- Write and call methods

Preliminaries

1. Create a Java project.
The name of the project must be **lab01_<your FirstNameLastName>**.
For example, my project would be named lab01_PeterNg.
2. Add a Java class named **NumericTypes** and **declare it as your main class** when the project is set up
3. Add the following comment block to the beginning of your Java class:

```
/*
 * <your name>
 * CS 16000-01 – 02/03, Fall Semester 2025
 * (Note: Write either 02 or 03, depending on which section your section is.)
 * Lab 01
 *
 */
```

Exercises

Suggested reading: textbook Chapter 2: 2.4 Primitive Data Types, Chapter 2: 2.8 Conversion between Primitive Data Types, and Chapter 2: 2.15 Read Keyboard Input.

For this assignment, describe the experiences and explanations as comments or comment blocks beside the program code.

1. Variables Declaration: Declare the following variables with their respective data types and comments on assigning values: byteNumber, shortNumber, intNumber, longNumber, floatNumber, and doubleNumber. Select the types as suggested by the identifiers. In addition, declare pbmTitle as a reference variable of a String class.

For each of the following problems 2 through 4, do part (A) and part (B):

2. 2(A). Assign all the variables (given in problem 1) their values according to the instructions in problems 2 through 4.

Example: byteNumber = 127; //a literal is a value that is written into
shortNumber = 127; //the code of a program

...
doubleNumber = 127; //a variable is a named storage in memory

pbmTitle = "For problem 2(A): "; //string literal

Then, display them on the console, applying the println() method, which has two parameters: pbmTitle and output. (These two statements can be reused throughout.)

Example: Declare a variable, output, as a reference variable of the String class, and then place this statement before the println() statement.

```
String output = "\n\tbyteNumber = " + byteNumber +  
                "; \n\tshortNumber = " + shortNumber +  
                "; \n\tintNumber = " + intNumber +  
                "; \n\tlongNumber = " + longNumber +  
                "; \n\tfloatNumber = " + floatNumber +  
                "; \n\tdoubleNumber = " + doubleNumber + ".";
```

```
System.out.println(pbmTitle + output);
```

```
// System.out.println("For problem 2(A): " + output); //This statement is the same as above
```

The println() method will display on the console the following output:

For problem 2(A):

```
byteNumber = 127  
shortNumber = 127  
intNumber = 127  
longNumber = 127  
floatNumber = 127.0  
doubleNumber = 127.0
```

Then, write a method, displayConsole(), which is as follows:

```
displayConsole(String pbm, byte bN, short sN, int iN,
               float fN, double dN) { ... }
```

placed in between “} //end of main”, and “} //end of class NumericTypes”. To be exact, the method header (referred to as (I)) is as follows:

```
public static void displayConsole(String pbm, byte bN, short sN,
                                int iN, float fN, double dN) { ... }
```

This method consists of seven formal parameters: pbm, bN, sN, iN, fN, and dN. They all designated a String class and various data types. Furthermore, they are associated one-to-one with the actual parameters of the call statement displayConsole(...) in the main() method.

The body of this method contains the following statements:

```
String output = "\nFrom displayConsole()\n" + pbm
               + "\n\tbyteNumber = " + bN
               + ";\n\tshortNumber = " + sN
               + ";\n\tintNumber = " + iN
               + ";\n\tlongNumber = " + lN
               + ";\n\tfloatNumber = " + fN
               + ";\n\tdoubleNumber = " + dN + ".";
System.out.println(output);
```

In the main() method, write a call statement

```
displayConsole(pbmTitle, byteNumber, shortNumber, intNumber,
               longNumber, floatNumber, doubleNumber);
```

The statement has seven actual arguments, which pass the values to the calling statement in (I).

2(B). Assign values to the corresponding variables through reading input from the keyboard using the Scanner class and its methods, nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), and nextDouble() (Chapter 2: 2.15 Reading Keyboard Input).

Note that, to use “Scanner keyboardInput = new Scanner(System.in);” you have to write the following import statement:

```
import java.util.Scanner; //Needed for Scanner class;
```

Placed this import statement before the public class NumericTypes{...}. The Eclipse editor has the following:

```
import java.util.Scanner; //Needed for Scanner class
```

```
/*
 * <your name>
 * CS 16000-01 - 02/03, Fall Semester 2024
 * (Note: Write either 02 or 03, depending on which section your section is.)
 * Lab 01
 *
 */

public class NumericTypes {

    public static void main(String[] args) {
        ...
        Scanner keyboardInput = new Scanner(System.in);
        ... //see the following Example

    } //end of main

    public static void displayConsole(String pbm, byte bN,
                                     short sN, int iN, float fN, double dN) {
        ... }

} //end of class
```

Example:

```
Scanner keyboardInput = new Scanner(System.in);
System.out.println("Enter six times the same integer number 123: ");
byteNumber = keyboardInput.nextByte();
shortNumber = keyboardInput.nextShort();
...
doubleNumber = keyboardInput.nextDouble();
```

Then, print the input data which are read from the keyboard.

```
pbmTitle = "For problem 2(B): ";

String output = "\n\tbyteNumber = " + byteNumber +
               "\n\tshortNumber = " + shortNumber +
               "\n\tintNumber = " + intNumber +
               "\n\tlongNumber = " + longNumber +
               "\n\tfloatNumber = " + floatNumber +
               "\n\tdoubleNumber = " + doubleNumber;

System.out.println(pbmTitle + output);
```

Likewise, to call the `displayConsole()` method, which is outlined in part 2(A) (using this method referred to as I), write a call statement in the `main()`: The call statement is as follows:

```
displayConsole(pbmTitle, byteNumber, shortNumber, intNumber,
               longNumber, floatNumber, doubleNumber);
```

For problem 2(B), do you get the same output as problem 2(A)? If not, correct it. The correction can be done by inserting the cross-out statement “`output = ...;`” before

```
System.out.println(pbmTitle + output);
```

Without any correction, both `println()` and `displayConsole()` methods must generate the same output.

~~For problems 3 and 4, do you get any error messages? If a statement has an error, correct it.~~

~~— Otherwise, insert `//` before the statement or use the proper cast operator. For example,~~

```
//intByte = 127; //?128
```

~~— There is a correct statement. Use the proper cast operator. For example,~~

```
intByte = (byte)127; //This works. //?128
```

~~Report your explanation in a comment after the assignments. Observe their differences (A) and (B), if any.~~

~~Report your explanation in a comment..~~

3. Use the program codes that were written for problems 2(A) and 2(B) to perform problem 3(A). and 3(B) with 123. And

```
pbmTitle = "For problem 3(A):  ";
```

3(A) Assign all the variables (given in problem 1) with a literal 123 as their values.

3(B) Assign the value 123 to the corresponding variables through reading input from the keyboard using the Scanner class and its methods, `nextByte()`, `nextShort()`, `nextInt()`, `nextLong()`, `nextFloat()`, and `nextDouble()`.

As in 2(A) and 2(B), use `println()` and `displayConsole()` methods to display the values for all these variables for (A) and (B).

Report your explanation in a comment. Do you get any error messages? **Are there any differences in output from (A) and (B)?**

4. (A) assign `intNumber = 127`. Then, re-assign `byteNumber`, `shortNumber`, `longNumber`, `floatNumber`, and `doubleNumber` with the variable `intNumber`. For example, since `intNumber` is an integer datatype, it is four bytes (32 bits) long. The variable `byteNumber` is one byte (8 bits) long. Therefore, use the cast operator `(byte)` or `(short)` to be a prefix of the variable `intNumber`.

```
byteNumber = (byte)intNumber;  
...  
longNumber = intNumber;  
floatNumber = intNumber;  
doubleNumber = intNumber;
```

As in 2(A) and 2(B), use `println()` and `displayConsole()` methods to display the values for all these variables for (A) and (B).

(B) 127 must be input by reading from the keyboard using the `Scanner` class first. Then,
`intNumber = keyboardInput.nextInt();`
Then, re-assign the variable `intNumber` to the variables `byteNumber`, `shortNumber`,
`longNumber`, `floatNumber`, and `doubleNumber`

As in 2(A) and 2(B), use `println()` and `displayConsole()` methods to display the values for all these variables for (A) and (B).

Report your explanation in a comment. Do you get any error messages? Correct it if there is.
Are there any differences in output from (A) and (B)? Compare your experience to the previous work in problem 2.

5. Re-assign all the variables (given in problem 1) to the following values in order. For each subproblem from a through f, do (A) assign the given integer to all the variables with `intNumber` and then (B) read input from the keyboard using the `Scanner` class.

As in 2(A) and 2(B), use the `displayConsole()` method to display the values for all the variables for (A) and (B) parts.

- a.* 128
- b.* 129
- c.* -128
- d.* -129
- e.* 32767
- f.* 32768
- g.* -32768
- h.* -32769
- i.* 2147483647
- j.* 2147483648

Do you get any error messages? Correct it if there is. If so, explain each and write your explanation in a comment. Is there any difference between part (A) and part (B)?

For problems 6 and 7, assign values to the variable `floatNumber` only. You do not need to read input from the keyboard.

6. Assign the **integer** literal 69 as a value of the variable `floatNumber`. Do you get any error messages? Correct the error if there is one. Print the value of the variable `floatNumber` to the console.

Re-assign `floatNumber` with the value 69.0 and then 69.68. Do you get any error messages? Correct the errors if there are. Print the assigned values to the variable `floatNumber` to the console.

Explain your experience in a comment for each of the cases.

7. Assign `intNumber` and `doubleNumber` the values 825 and 751, respectively. Declare **ratio** as a variable of type `double` and assign **ratio** the value `intNumber` divided by 751. Print the result to the console. Then, print the result with four (4) digits after the decimal point to the console. Re-assign the **ratio** to the value `intNumber` divided by `doubleNumber` and print the ratio to the console. Then, print the result to the console with three (3) digits after the decimal point. Explain the printed results in a comment.

8. Print the following expressions to the console

- `'a' + 'b'`
- `" " + 'a' + 'b'`
- `" " + ('a' + 'b')`
- `(" " + 'a' + 'b')`
- `(" " + 'a') + 'b'`
- `' ' + 'a' + 'b'`
- `' ' + ('a' + 'b')`
- `(' ' + 'a' + 'b')`
- `(' ' + 'a') + 'b'`

Explain your printing experience in a comment.

9. Use `println()` to print the following expressions to the console

- `"These are numbers: " + intNumber + " and " + doubleNumber`
- `"These are numbers: " + intNumber + doubleNumber`
- `"These are numbers: " + (intNumber + doubleNumber)`

Explain your printing experience in a comment. Do you get the following output:

These are numbers: 825 and 751.0

These are numbers: 825751.0

These are numbers: 1576.0

10. Write a single `"System.out.println();"` to print (display) the following expressions to the console

- `"Formula1: " + (825 + 751/intNumber*doubleNumber + 1)`
- `"Formula2: " + 825 + 751/intNumber*doubleNumber + 1`

- "Formula3: " + (825 + 751)/intNumber*(doubleNumber + 1)
- "Formula4: " + ((825 + 751)/intNumber*(doubleNumber + 1))
- "Formula5: " + ((825 + 751)/intNumber*doubleNumber + 1)
- "Formula6: " + (825 + 751)/intNumber*doubleNumber + 1

Explain your printing experience in a comment.

The output is:

```
Formula1: 826.0
Formula2: 8250.01
Formula3: 752.0
Formula4: 752.0
Formula5: 752.0
Formula6: 751.01
```

Submit: Zip your project folder and upload the zip file to Brightspace at the designated place.

//Lab01 JFundScanner Q 09022025