# Department of Computing

**CS 212: Object Oriented Programming**

**Class: BESE-11 AB**

**Lab03: Basic Programming in Java**

**Date: 17th March, 2021**

**Time: 9:00pm- 12:00pm & 2:00-5:00 pm**

**Instructor:**

Ms. Hania Aslam

## Learning Objectives

After completing this lab you will be able to do variable declarations and initializations with some printing and math in Java.

We'll try a few experiments using it to see what happens for some basic errors. As you carry out the steps in this lab, it is important to think about what you are seeing and to understand what's going on. The more attentive you are here the fewer headaches you'll have later!

## Basic Lab Instructions!

* Talk to your classmates for help.
* You may want to bring your textbook to future labs to look up syntax and examples.
* Stuck? Confused? Have a question? Ask a TA/Lab Engineer for help, or look at the book or past lecture slides.
* Complete as many problems as you can within the allotted time. You don't need to keep working on these exercises after you leave the lab.
* Before you leave today, make sure to check in with one of the Lab Engineers/TAs in the lab to get credit for your work.

## Activity #1.

Your first task will be to create a "Hello World" program using Java. Create a new directory CS-212, and navigate into it. Using notepad or the editor of your choice (preferably NetBeans), create a file called ActivityOne.java and enter the following code:

// This program outputs the message "Hello World" public class ActivityOne

{

public static void main (String args[])

{

System.out.println ("Hello World");

}

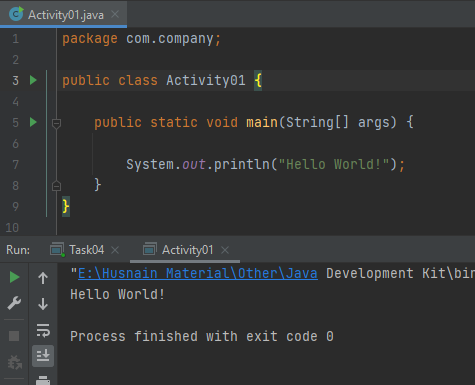
}

## In case you are using NetBeans, just Build and Execute the project.

Alternatively, if you are working on a Linux like environment then save the file, and open the



command window to use it for compiling and running your program. In the new window, navigate into your CS-212 directory and issue the following command to compile the program.



## javac ActivityOne.java

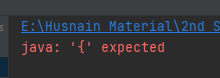
If you get any compiler messages, then the learning process has already started! Make sure the name of the file and the name of the class are identical, and that you're in the right place in the directory structure, and the code is exactly as shown above.

Once the program compiles, enter the following command to run it on the command prompt.

## java ActivityOne

Once your program compiles successfully, you will see the output in the console window.

After getting the program to work, let's see what happens if we deliberately break it. We'll try several variations that violate java syntax to learn what happens. Watch carefully to see what message the compiler generates for each of these small errors—they are mistakes every programmer makes on a regular basis.

1. Start by changing the name of the class in the first line, saving the file, and then attempt to re-compile it with the javac/NetBeans Build command. Observe what happens.
2. 
3. Now restore the name in the class header and try removing one of the opening braces, then save and compile again. Put the brace back and remove the closing brace that matched it, and save and compile again. Make a note of the messages the compiler generates for each change.
4. 
5. Try a version with a lowercase "s" at the beginning of the output statement instead of the uppercase "S" (for System class).
6. 
7. Finally, try a version where the only change is the removal of the semicolon at the end of the print statement. This is a mistake every programmer makes more than once—it's good to see how the compiler responds.
8. 

## Activity #2.

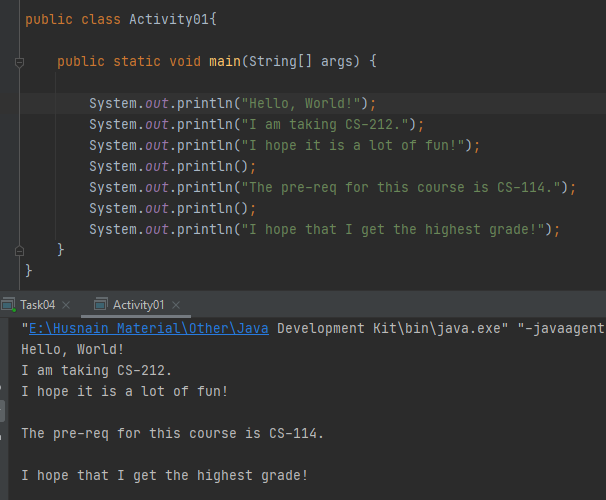
Modify your ActivityOne.java file to produce the following console output. Note the blank lines; you should include those in your output as well.

Hello, world!

I am taking CS 212.

I hope it is a lot of fun!

The pre-req for this course is CS-114. I hope I get the highest grade!



**Activity #3.**

Programs should be indented properly to make them easier to read.

{ brace -> increase indent of following lines by one tab

} brace -> decrease indent of that line and following lines by one tab The following program has poor indentation. Fix it.

public class ActivityThree {

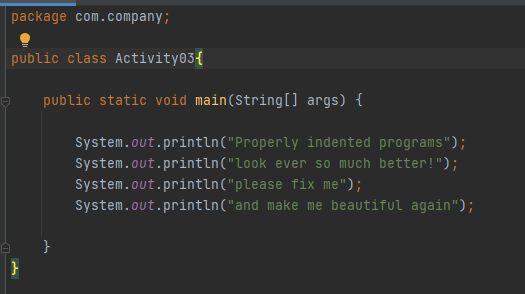
public static void main(String[] args) { System.out.println("Properly indented programs");

System.out.println("look ever so much better!"); System.out.println("please fix me");

System.out.println("and make me beautiful again");

}

}



## Activity #4.

How many lines of output are produced (including blank lines)? public class ActivityFour {

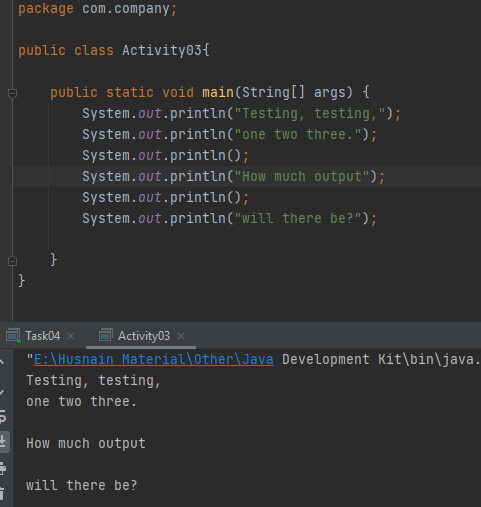
public static void main(String[] args) {

System.out.println("Testing, testing,"); System.out.println("one two three."); System.out.println();

System.out.println("How much output"); System.out.println(); System.out.println("will there be?");

}

}



Hence there are 6 lines of output produced.

**Activity #5.**

The following program contains 11 errors! What are they? public class ActivityFive

public static main(String args) {

System.out.println(Hello world); system.out.Pritnln("Do you like this program"?); System.out.println()

System.println("I wrote it myself.";

{

}

Once you think you've found the errors, create/compile/run a corrected version of this program.

1-) No curly brackets after class name in line 1.

2-) No keyword ‘void’ in line 2.

3-) No square brackets in the parameter of the main method in line 2.

4-) No quotation marks in the argument of the ‘println’ function in line 3.

5-) The ‘S’ of the System class is small in line 4.

6-) The ‘p’ of the println function is capital in line 4.

7-) The print function is spelled incorrectly in line 4.

8-) The question mark is out of the quotation marks in line 4.

9-) No semicolon at the end of line in line 5.

10-) The print statement is written incorrectly in line 7. It should be ‘System.out.println()’.

11-) The closing parenthesis is missing in line 7.

12-) Curly bracket should be closed (It is open) in line 8.

## 

## Activity #6.

Discover what error messages the compiler produces when you make each of the following mistakes. How many unique error messages are you able to cause the compiler to produce?

1. Naming your file incorrectly, then compiling.



1. Forgetting a keyword such as void or class



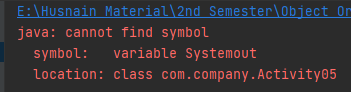
1. Forgetting a quotation mark "



1. Forgetting a parenthesis ( or )



1. Forgetting a dot .



1. Using too many or too few braces { or }



Notice that the error messages don't always make it obvious what is wrong. But they usually tell you the right line number to fix.

## Activity #7.

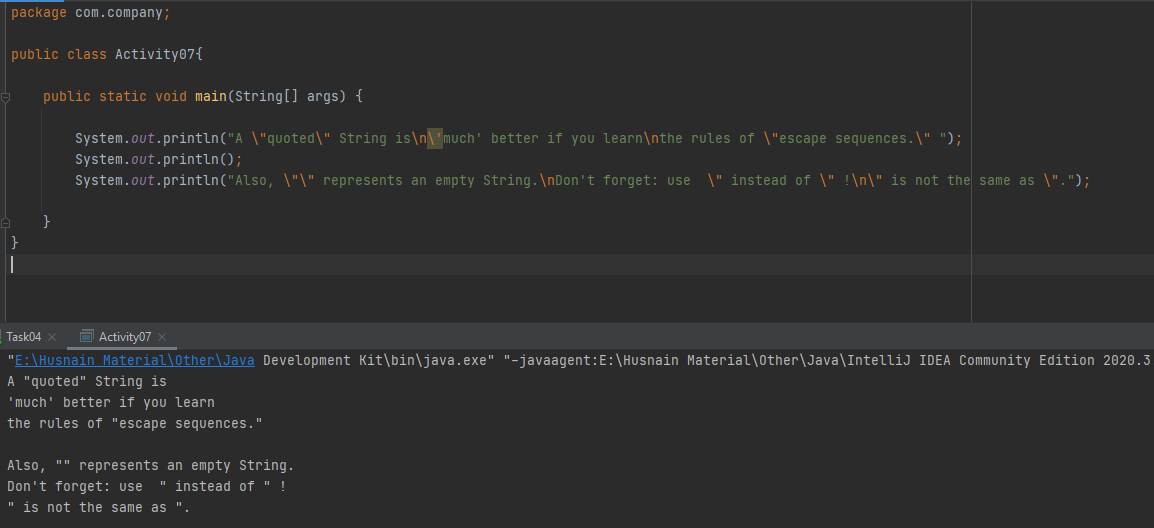
Write a complete Java program that produces the following output (note the blank line):

A "quoted" String is 'much' better if you learn

the rules of "escape sequences."

Also, "" represents an empty String. Don't forget: use \" instead of " !

'' is not the same as ".



**Activity #8.**

Write a complete program that produces the following output:

\/

\\//

\\\///

///\\\

//\\

/\



## Activity #9.

Change the following program to use compound assignments: class ActivityNine {

public static void main (String[] args){

int result = 1 + 2; // result is now 3 System.out.println(result);

result = result - 1; // result is now 2 System.out.println(result);

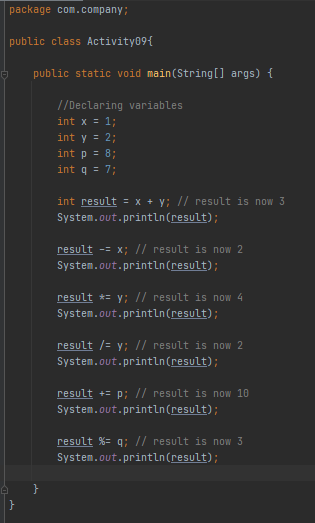
result = result \* 2; // result is now 4 System.out.println(result);

result = result / 2; // result is now 2 System.out.println(result);

result = result + 8; // result is now 10 result = result % 7; // result is now 3 System.out.println(result);

}

}



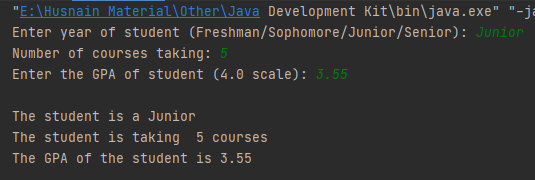
## Task #1:

Writing a program that stores a student’s year (Freshman, Sophomore, Junior, or Senior), the number of courses the student is taking, and his or her GPA on a 4.0 scale. Declare variables with the appropriate names and types to hold this information. Print the stored information at the end.

**Code:**

package com.company;  
import java.util.Scanner;  
  
public class Task01 {  
 public static void main(String[] args) {  
  
 //Declaring variables  
 String year;  
 int courses;  
 float gpa;  
  
 Scanner sc = new Scanner(System.*in*);  
  
 //Taking Inputs  
 System.*out*.print("Enter year of student (Freshman/Sophomore/Junior/Senior): ");  
 year = sc.nextLine();  
  
 System.*out*.print("Number of courses taking: ");  
 courses = sc.nextInt();  
  
 System.*out*.print("Enter the GPA of student (4.0 scale): ");  
 gpa = sc.nextFloat();  
 System.*out*.println();  
  
 //Printing Outputs  
 System.*out*.println("The student is a " + year);  
 System.*out*.println("The student is taking " + courses + " courses");  
 System.*out*.println("The GPA of the student is " + gpa);  
 sc.close();  
 }  
}

**Output:**



## Task #2:

Write an application that inputs one number consisting of five digits from the user, separates the number into its individual digits and prints the digits separated from one another by three spaces each. For example, if the user types in the number 42339, the program should print:

4 2 3 3 9

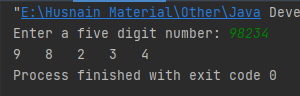
Assume that the user enters the correct number of digits. What happens when you execute the program and type a number with more than five digits? What happens when you execute the program and type a number with fewer than five digits?

**Hint:** You may want to use repeated division to separate out the digits.

**Code:**

package com.company;  
import java.util.Scanner;  
  
public class Task02 {  
  
 public static void main(String[] args) {  
  
 //Declaring required variables for the program  
 int num;  
 int first\_digit;  
 int second\_digit;  
 int third\_digit;  
 int forth\_digit;  
 int fifth\_digit;  
  
 Scanner sc = new Scanner(System.*in*);  
  
 //Taking Input  
 System.*out*.print("Enter a five digit number: ");  
 num = sc.nextInt();  
  
 //Calculating every digit  
 first\_digit = (num / 10000);  
 second\_digit = ((num % 10000) / 1000);  
 third\_digit = (((num % 10000) % 1000) / 100);  
 forth\_digit = ((((num % 10000) % 1000) % 100) / 10);  
 fifth\_digit = ((((num % 10000) % 1000) % 100) % 10);  
  
 //Printing Output  
 System.*out*.printf("%d %d %d %d %d", first\_digit, second\_digit, third\_digit, forth\_digit, fifth\_digit);  
 sc.close();  
  
 }  
}

## Output:



## When the input is greater than 5 digits:

## 

## When the input is smaller than 5 digits:

## 

## Task #3:

Suppose you have a real number variable x. Write a Java expression that computes the following value y while using the \* operator only four times:

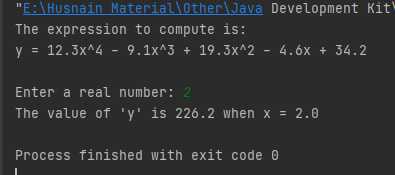
## y = 12.3x4 − 9.1x3 + 19.3x2 − 4.6x + 34.2

## Hint: You may want to look up for a built-in library function to achieve this.

**Code:**

package com.company;  
import java.util.Scanner;  
  
public class Task03 {  
  
 public static void main(String[] args) {  
  
 //Declaring variables  
 float x;  
 float part\_1;  
 float part\_2;  
 float part\_3;  
 float part\_4;  
 float part\_5;  
  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("The expression to compute is: ");  
 System.*out*.println("y = 12.3x^4 - 9.1x^3 + 19.3x^2 - 4.6x + 34.2\n");  
  
 //Taking Input  
 System.*out*.print("Enter a real number: ");  
 x = sc.nextFloat();  
  
 //Breaking the expression into parts and calculating the value of each  
 part\_1 = (float) ((12.3) \* (Math.*pow*(x, 4)));  
 part\_2 = (float) ((9.1) \* (Math.*pow*(x, 3)));  
 part\_3 = (float) ((19.3) \* (Math.*pow*(x, 2)));  
 part\_4 = (float) (4.6 \* x);  
 part\_5 = 34.2F;  
  
 //Printing Output  
 System.*out*.println("The value of 'y' is " + (part\_1 - part\_2 + part\_3 - part\_4 + part\_5) + " when x = " + x);  
 }  
}

**Output:**



**Task #4:**

In physics, a common useful equation for finding the position s of a body in linear motion at a given time t, based on its initial position s0, initial velocity v0, and rate of acceleration a, is the following:



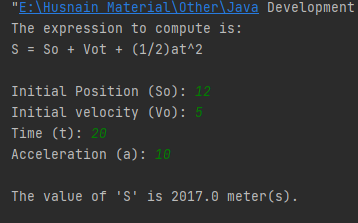
Write code to declare variables for s0, v0, **a, and t**, and then write the code to compute **s** on the basis of these values.

**Hint:** Java has a built-in function to compute power in Math class.

**Code:**

package com.company;  
import java.util.Scanner;  
  
public class Task04 {  
 public static void main(String[] args) {  
  
 //Declaring variables  
 float S;  
 float So;  
 float Vo;  
 float a;  
 float t;  
 float part\_1;  
 float part\_2;  
 float part\_3;  
  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("The expression to compute is: ");  
 System.*out*.println("S = So + Vot + (1/2)at^2\n");  
  
 //Taking Input  
 System.*out*.print("Initial Position (So): ");  
 So = sc.nextFloat();  
 System.*out*.print("Initial velocity (Vo): ");  
 Vo = sc.nextFloat();  
 System.*out*.print("Time (t): ");  
 t = sc.nextFloat();  
 System.*out*.print("Acceleration (a): ");  
 a = sc.nextFloat();  
 System.*out*.println();  
  
 //Breaking the expression into parts and calculating the value of each  
 part\_1 = So;  
 part\_2 = Vo;  
 part\_3 = (float) ((0.5) \* a \* (Math.*pow*(t, 2)));  
  
 S = (part\_1 + part\_2 + part\_3);  
  
 //Printing Output  
 System.*out*.println("The value of 'S' is " + S + " meter(s).");  
 sc.close();  
 }  
}

## Output:



## Hand in

Hand in the source code from this lab at the appropriate location on the LMS system. You should hand in a single compressed/archived file named Lab\_1\_<Your CMS\_ID. Your\_NAME >.zip (without angle brackets) that contains ONLY the following files.

1. All completed java source files representing the work accomplished for this lab: ActivityOne.java; ActivityTwo.java; ActivityThree.java; ActivityFour.java; ActivityFive.java; ActivitySix.java; ActivitySeven.java; ActivityEight.java; ActivityNine.java; Task1.java, Task2.java, Task3.java, Task4.java. The files should contain author in the comments at the top.

## To Receive Credit

1. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.
2. The lab time is not intended as free time for working on your programming/other assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the programming/other assignments