# LAB 1: CREATING, COMPILING AND RUNNING JAVA PROGRAMS

Four (4) activities for this lab:

- Installing JDK
- Installing Apache NetBeans (or your preferred editor)
- Creating, compiling and running the example
- Do and submit the given exercises.

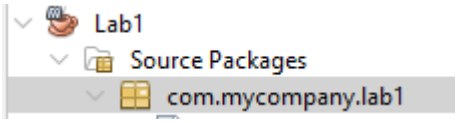
Highly recommended to get these done before the lab session

**1) Installing JDK**
**2) Installing Apache NetBeans**

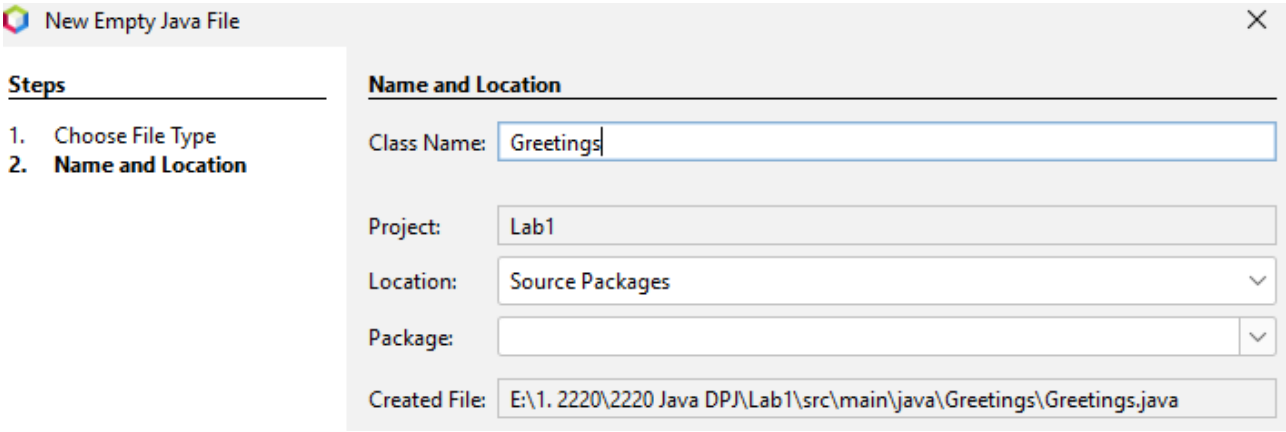
- Please refer to the provided instructor manual.
- You may also create a Java project without Maven in Apache NetBeans.
  - Java with Ant
  - https://www.geeksforgeeks.org/creating-java-project-without-maven-in-apache-netbeans-11-and-higher/

**3) To create, compile and running the example (Example 1-Greetings.java)**

- Create a new project, named it as **Lab1.**
- Click the default package (*com.mycompany.lab1*) and right click.

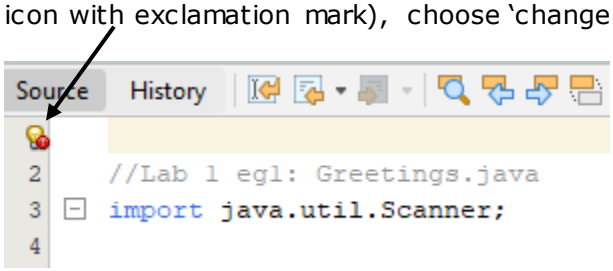



  - New > Other > Empty Java file (this option is to create an empty java file)
  - You should see a page as below. Name the class name accordingly and click 'Finish'.




  - Type the program. Once done save the file.

- Change/declare the package accordingly by right click the error (the lightbulb icon with exclamation mark), choose 'change package declaration to..."

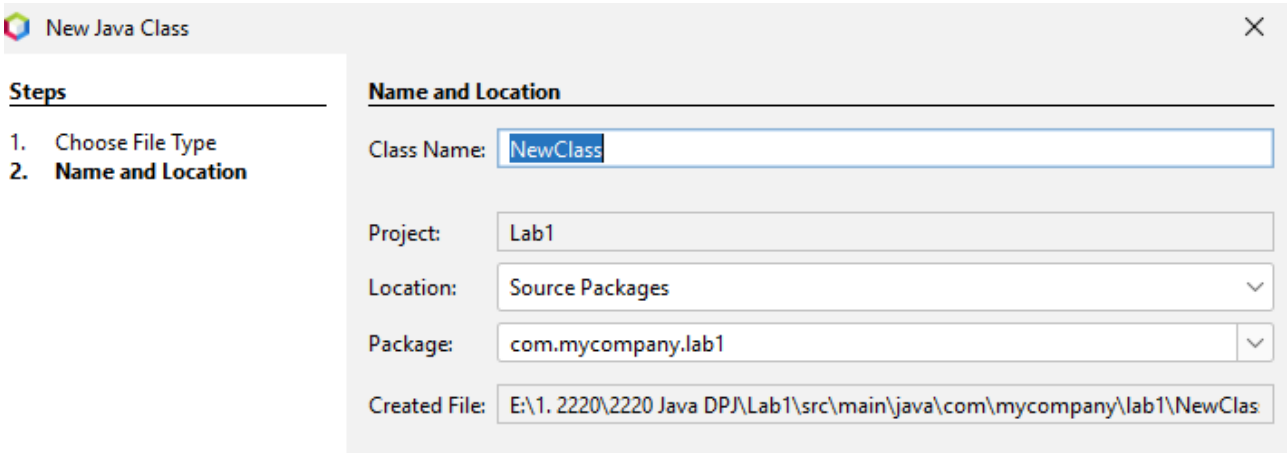



```
1  package com.mycompany.lab1;
2
3  //Lab 1 eg1: Greetings.java
4  import java.util.Scanner;
```

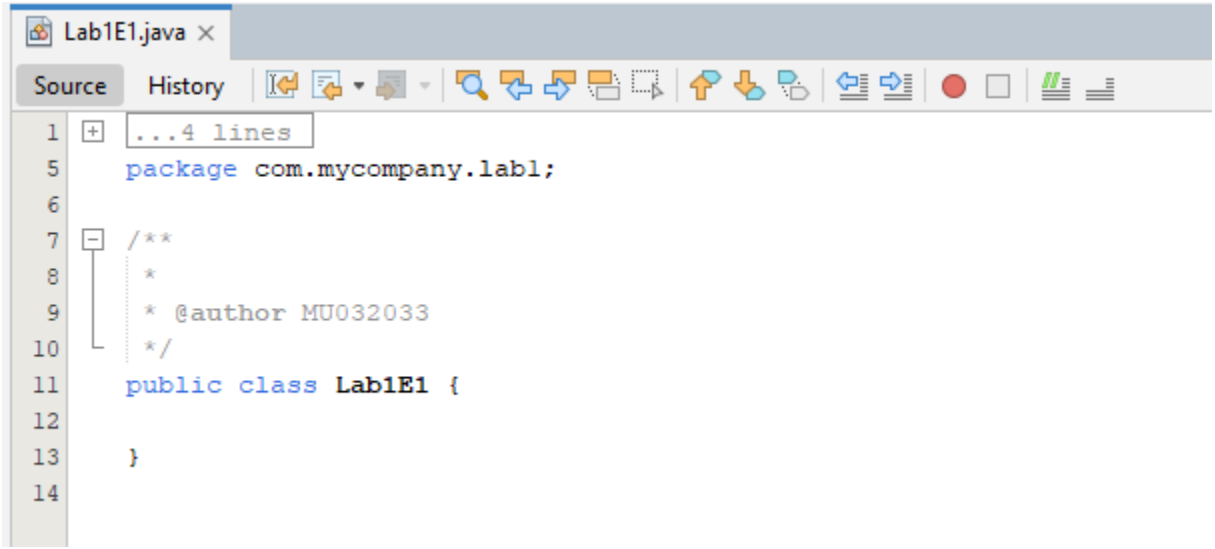
- Save the program
- To run: ***Run > Run File (Shift F6).***

**4) Refer to Page 6 and Page 7 for the questions.**

- Select/click the default package (*com.mycompany.lab1*) and right click.
  - New > Java Class (this option will create the class for you).
  - Name the class accordingly. Suggestion: Lab1Question1, Lab1E1, Lab1Question2 etc.
  - Next click 'Finish'.

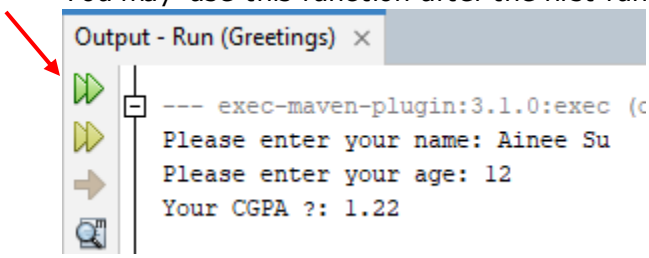



- Next, you should see a workspace for you to start coding.



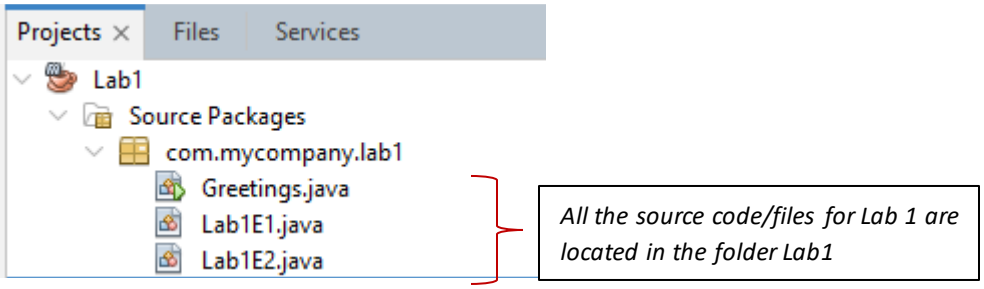

- Provide your info as instructed (in the comment).
- Read the question properly and create/write the program.
- To run: ***Run > Run File (Shift F6)***

  - You may use this function after the first run, to re-run.




- Repeat the steps for Question 2.
  - Select/click the default package (*com.mycompany.lab1*) and right click.
  - New > Other > File types > Java Main Class (this option will create a class with a main method).

- Your left panel (Window Group) may look like this:



**EXAMPLE 1:** An example of program that requires simple input routines. This program is using Scanner class.

**Problem**: Write a program that prompts a user to enter name, age and CGPA, and then greets the user and shows the year that the user was born.

**Solution**:

```
//Lab 1 eg1: Greetings.java

import java.util.Scanner;

public class Greetings
{
    public static void main (String [] args)
    {
        Scanner input = new Scanner(System.in);
        final int YEAR = 2023;

        System.out.print("Please enter your name: ");
        String name = input.nextLine();

        System.out.print("Please enter your age: ");
        int age = input.nextInt();

        int year = YEAR - age;

        System.out.print("Your CGPA ?: ");
        double cgpa = input.nextDouble();

        System.out.println("\nHello "+ name);
        System.out.println("You were born in "+ year);
        System.out.println("CGPA: "+ cgpa);
        System.out.printf("CGPA: %.3f\n", cgpa);
    }
}
```

What is this?

final: To declare constant

Prompt user to enter name

Prompt user to enter age

Calculate the year user was born

Prompt user to enter CGPA

Using method **printf,** and displaying result in 3 decimal places

Review: Save (compile), run the program and fix any error(s)

**For discussion:**

a) Method **System.out.print** vs. **System.out.println** vs **System.out.printf**
b) **Scanner** class
c) Method **next()** vs. **nextLine()**
d) **String** class
e) The symbol +.

## Note

**Scanner Class**

- Example no 1 uses Java's predefined **Scanner** class from package **java.util.**
- A scanner enables a program to read data (e.g. numbers) for use in the program. The data can come from many resources (file or keyboard).
- Before using Scanner, the program must create it and specify the source of the data.
- **System.in** is to input data – enables Java applications to read information typed by the user.
- **nextDouble()–** to obtain a double from a user. **FIND** other methods that can be used.
- For more details >> https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Scanner.html

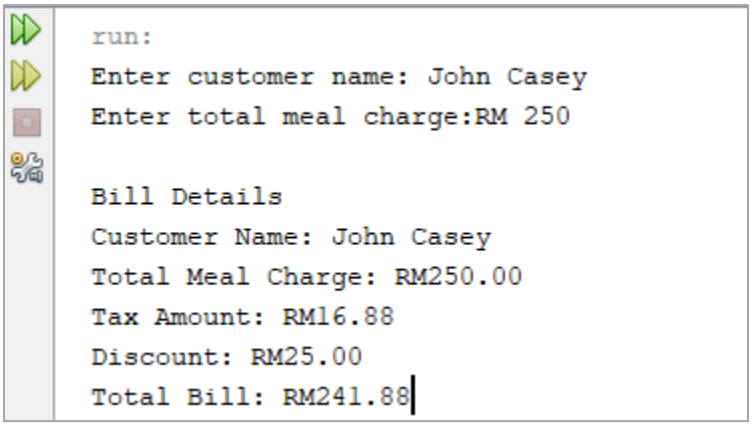
# LAB EXERCISE

***Reminder:***
*Write the following info in your java file:*
*//Name and ID*
*//Group*
*//Lab x /Q x*

## QUESTION 1:

Write a program that computes the service tax and discount on a restaurant bill.

- The program should ask the user to enter customer name and charge for the meal.
- The tax is **6.75%** and the discount for that day is **10%.** Declare the two values as **constants**.
- Then, calculate the total bill for the customer.
- Display the customer name, the meal charge, tax amount, discount amount and total bill. You may refer to the output screen.



## QUESTION 2

Write a program to compute the salesperson's commission amount and monthly salary. At the end of each month, all salespersons receive a guaranteed base salary amount ( **RM 2500**) and also ***earn an amount of commission based on the amount of sales*** they make.

- Declare a ***constant*** to represent the base salary amount.
- The program should prompt the user to enter the sales amount.
- The commission amount is calculated according to the table below. The program will display the ***commission amount*** and the ***monthly salary***.

| Sales Amount (RM) | Commission Rate (%) |
|---|---|
| Less than 10000.00 | 5% |
| 10000.00 – 14999.99 | 10% |
| 15000.00 – 19999.99 | 12% |
| 20000.00 and above | 15% |

- The program will display the ***base salary, commission amount*** and the ***monthly salary*** as shown in the sample output.

**Sample Output:**

```
Enter sales amount: RM 19999.99
Base Salary: RM 2500.00
Commission Amount: RM 2400.00
Monthly Salary: RM 4900.00

------------------------------------
BUILD SUCCESS
```