Department of Computing
Xi'an Jiaotong-Liverpool University

Lab 2 – Introduction to Java Programming in NetBeans

Aim

The aim of this lab is to learn the very basics of Java and NetBeans. At the end of the lab, you should be able to open NetBeans, create a project, and modify simple Java programs to display information.

Resources

You will find your lecture notes useful

Tips:

1. If you are not sure why you are doing something, ask a TA. This is what they are here for.

2. These labs are expected take more than the 2 allocated hours. You should complete them in your own time before the next classes. Practice makes perfect!

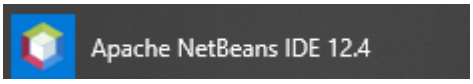
3. The lab is an online lab, but a video is provided, and forums are available for questions

Folder Management

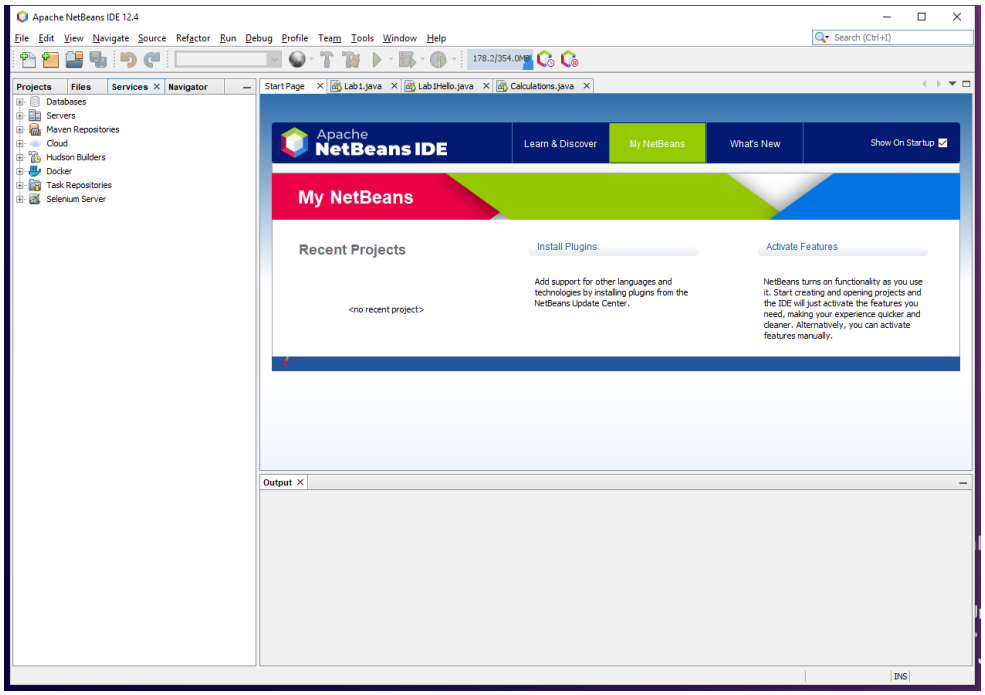
- It is important to manage your folders correctly. You should create a CPT111 folder, and then a folder for this week's work. All your work in this lab should be saved within this folder.
- Each week, you should create a new folder for your work

Programming with NetBeans

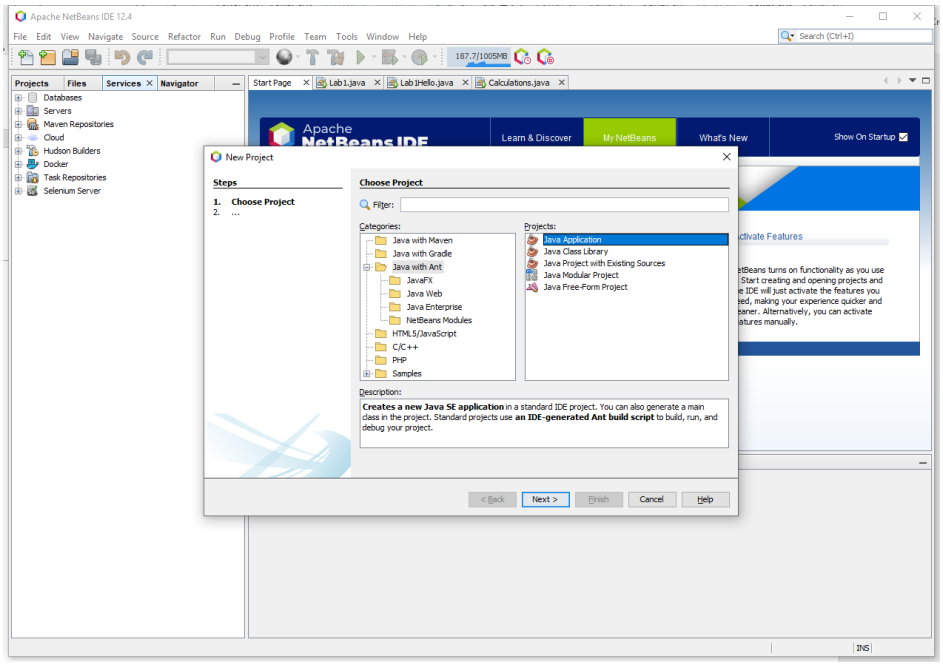
- Open NetBeans, which should be in your start menu. Look for this icon:



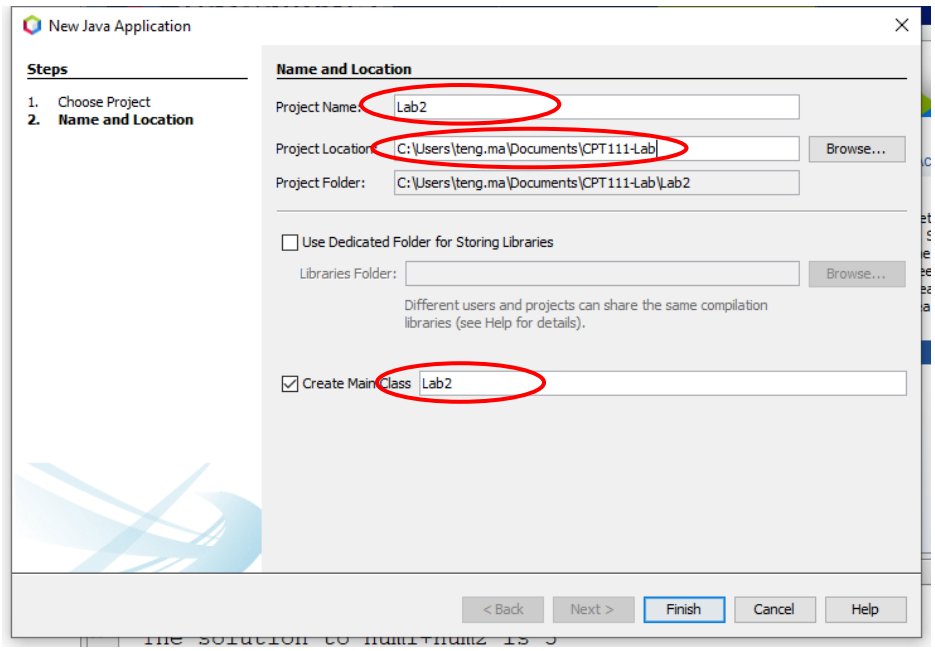
- The main GUI is shown as follows:



- Click "File" -> "New Project"; Select "Java with Ant" in Categories list and "Java Application" in Projects List; Click "Next";

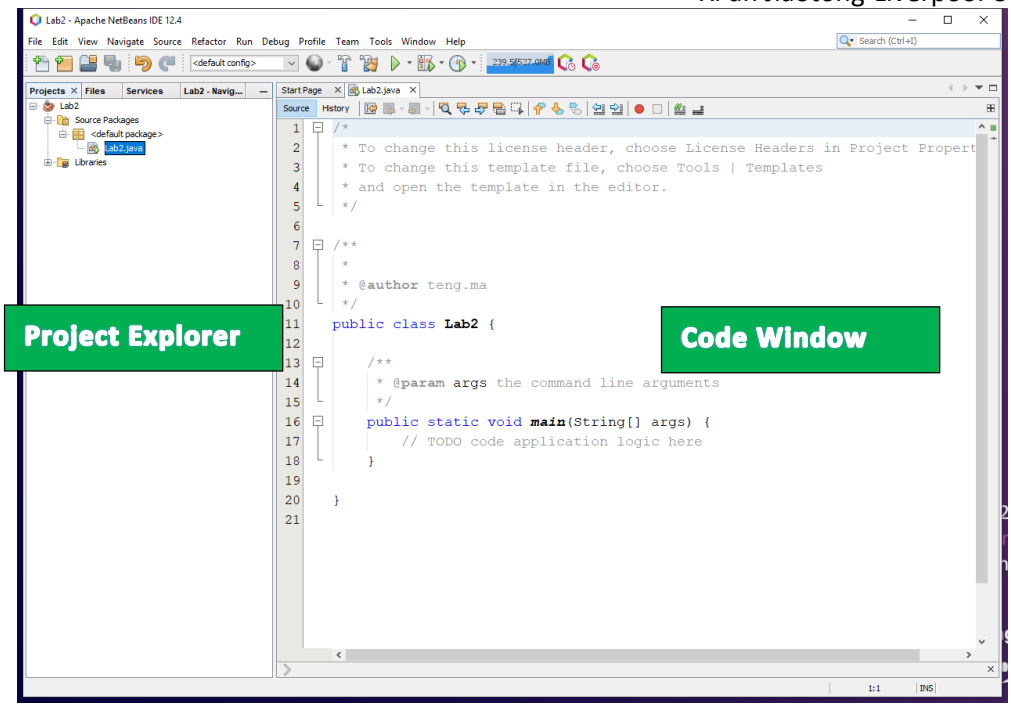


- you should see a "New Java Application" Dialog:



- Set the project name and project location; Click "Finish"; You should see several panels:

Department of Computing
Xi'an Jiaotong-Liverpool University



- The left panel is the **project explorer**. This is where the java files and projects you create will be displayed. They can be opened by clicking on them.

- The **code window** is where you will type your code. All the java files that you are working on can be edited in this window.

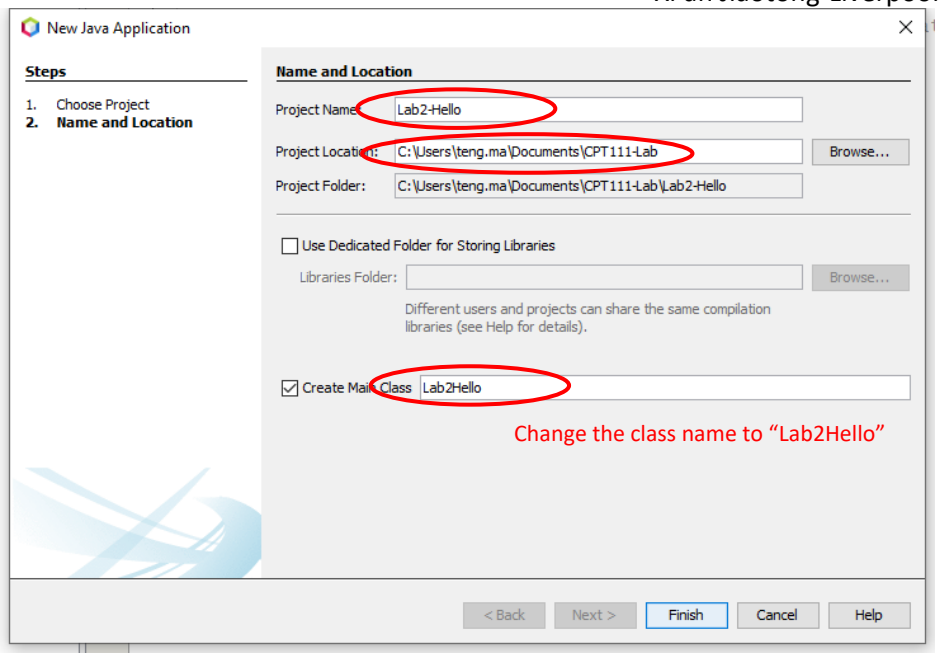
A First Java Project

- Create a new NetBeans project by navigating to "File" in the menu at the top of the screen, and then "New Project".

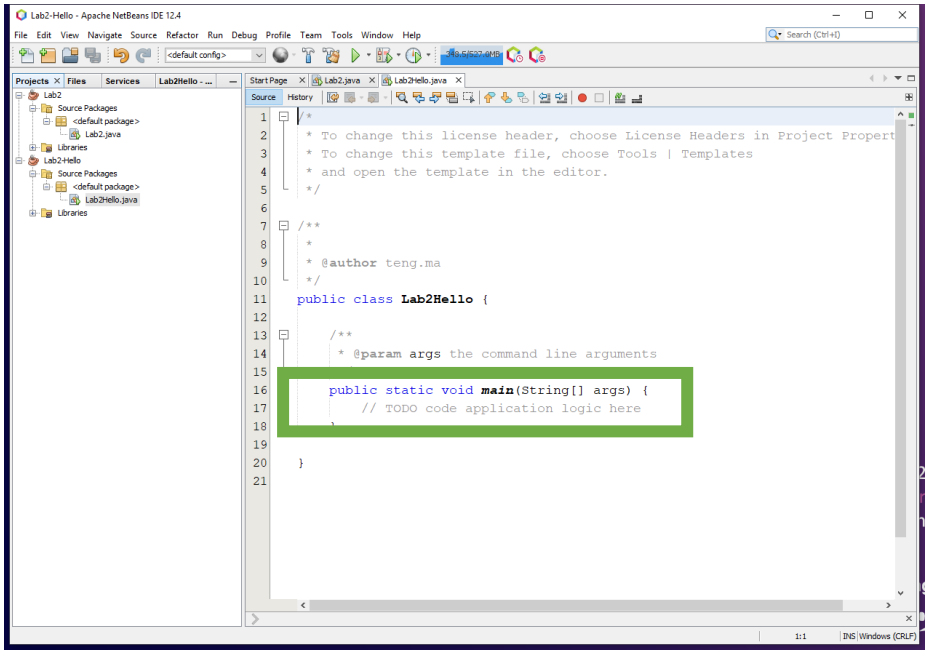
- Choose to create a new "Java Application"

- Call your project "Lab2-Hello". Make sure to set the folder location to the folder you created previously.

- Press Finish to create your new project.



- As you chose to create a main class, the project will automatically create a Java file, which has a main method (This will be discussed in future weeks). Everything you need to do this week is within the main method.

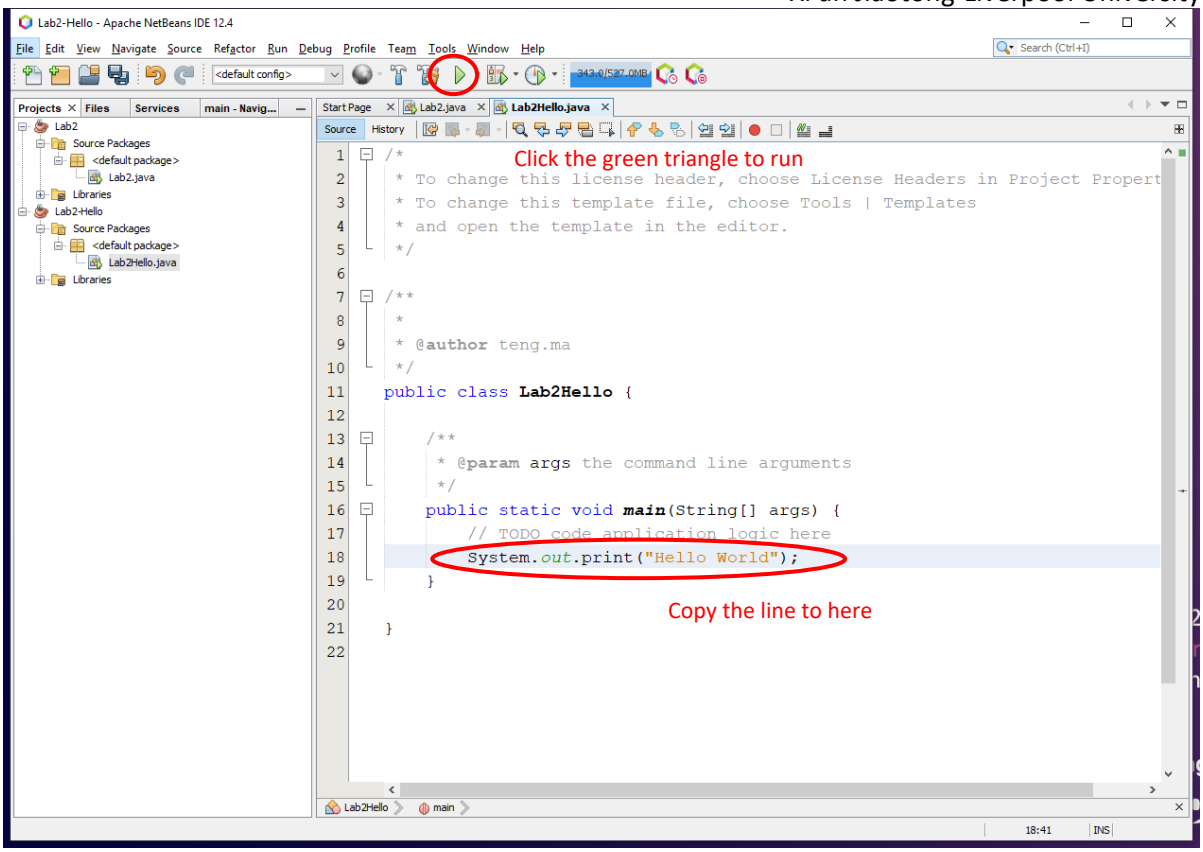


- In the main method, add a System.out, which is an instruction to output text to the **console window**. Copy the line below to your main method between two curly brackets.

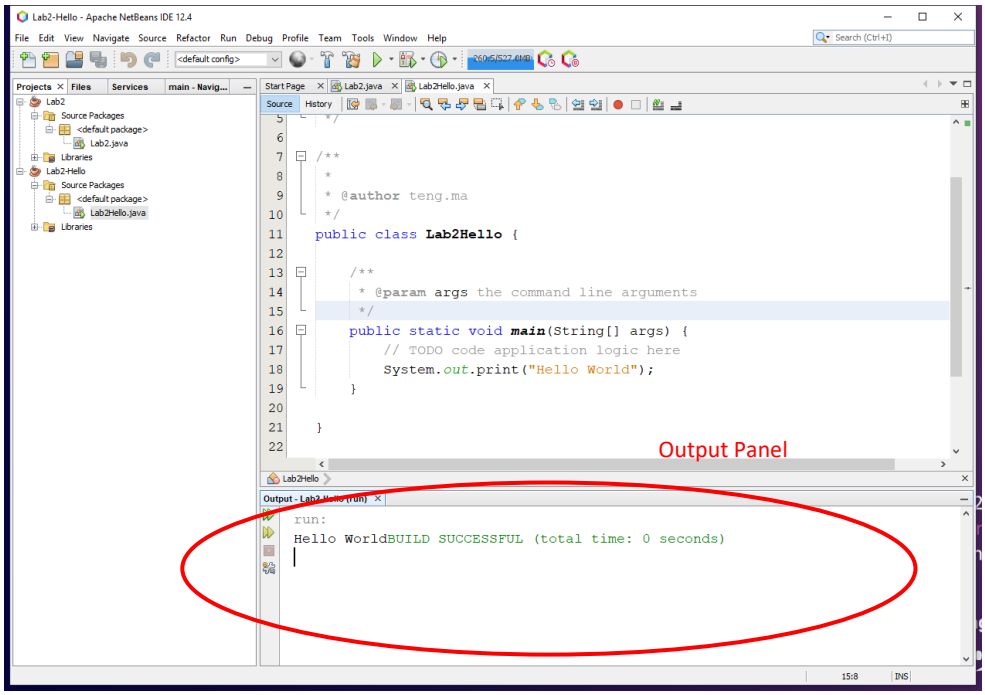
```
System.out.print("Hello World");
```

- Run your program by clicking on the green triangle at the top of the screen, or pressing "F6", or right click the file on the right panel and click "Run File" on the pop-up menu.

Department of Computing
Xi'an Jiaotong-Liverpool University



- Look at the output panel at the bottom of the screen, check that you can see the message. Ask if you can't see it!



- Add 4 extra system.out commands to print more hello messages on screen. Check that you can see the change in the results

- Change the System.out commands from System.out.print, to System.out.println, i.e.:
  `System.out.println("Hello World");`

- What is the difference? Do you understand it? If you do not, check your lecture notes or ask!

- Make some modifications to this System.out. Make it say hello to you (add your name). Add another line, very similar, and make the program say hello to someone else!

- Add several more System.out lines to say something interesting about yourself. Add a line to give an interesting fact about yourself.

- Add a line to give your school, name, and hometown.

- Add a line to give your name, age, and student number.

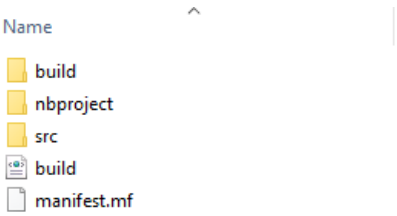
- Add a line to give your student number and major.

- The console output should look something like:

```
Hello Teng
My name is Teng, and I'm a lecturer in CPT
My name is Teng, my school was UoN, and my hometown is Zhengzhou.
Name: Teng, Age: 30, Student number: 12345678
…
```

Class and Java Files

The files you have worked on have been Java files, i.e. Lab2Hello.java. As discussed in the lecture, these are not the files that the computer runs. Navigate to the folder that was created for this project. You should see something like:



- Click on the "src" folder, and then click through the different packages until you reach the java files themselves. These are the files that you edit and make changes to.

- When you compile and run the project though, they are used to create class files, which run on the Java virtual machine. To see these, navigate to the project folder again, and click on "build".

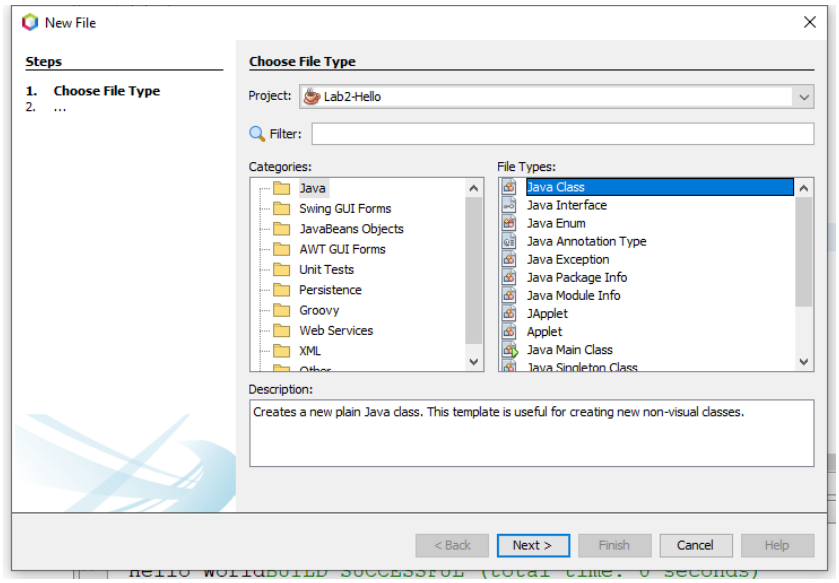
- Click on "Classes" and then click on the different package folders until you see a number of class files, with names such as hello.class.

- These are the class files that are created from your Java files, and are run on the virtual machine. It is important to understand the difference between these and the Java source files. If you are unsure, ask!

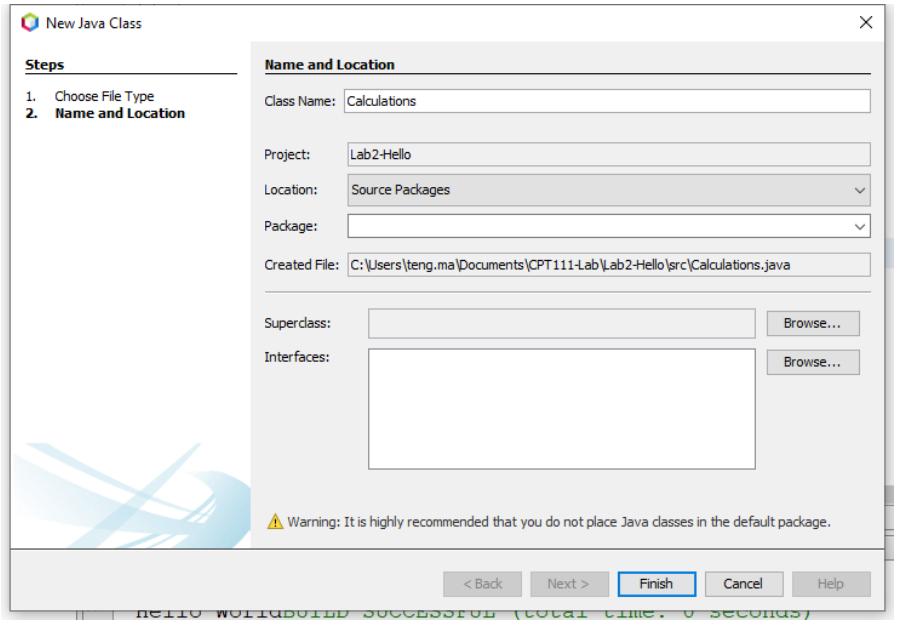
From Strings to Arithmetic

As well as displaying information, calculations can also be performed with Java. Here, we will create a new class and modify it to make use of integers, variables, calculations, and then display outputs.

- Create a new class (File > New File), and select "Java Class"



- Click "Next", Set the class name as "Calculations":



- and press "finish". This will give you an empty class with only this text in it:

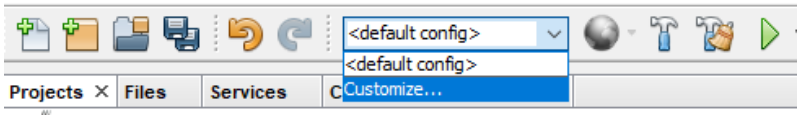
```
/*
 * To change this license header, choose License
 * To change this template file, choose Tools |
 * and open the template in the editor.
 */

/**
 *
 * @author teng.ma
 */
public class Calculations {

}
```

- You need to create a main method (discussed in future weeks). To do this, you can copy the following text into two curly brackets in class Calculations:

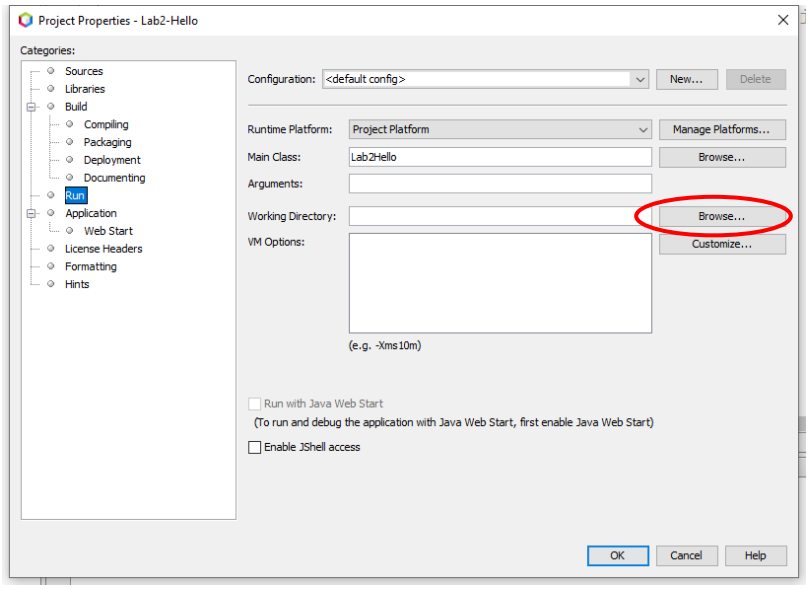
```
public static void main(String[] args) {

}
```

- To test it, add a System.out with your name in it, as you did in the previous program, and run it to make sure it works.

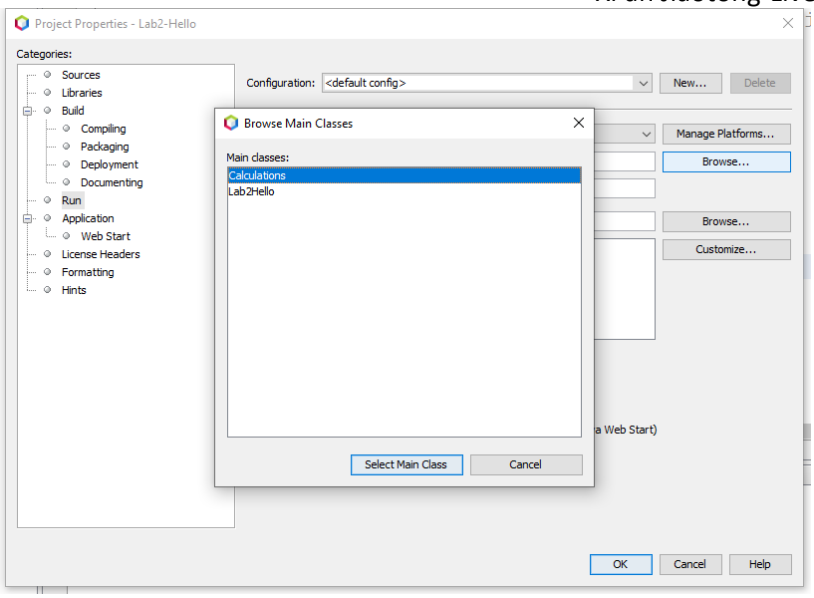
- You may find that when you run it, it still runs the other program. This is because your project now has two **main** classes, and it is running the other one. To solve this:
    1. Select the "customize" option in the toolbar



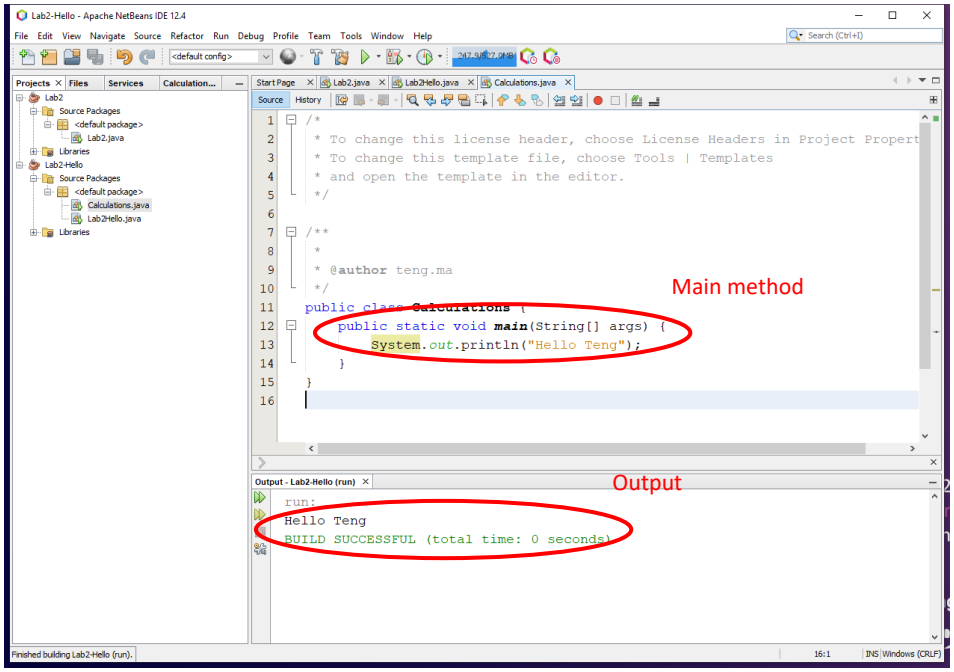
    2. In the window that appears, next to the "Main Class", click the "Browse" button



    3. Select the class that you have created. This will set it as main.



4. Run the project to check that your message displays on the **output panel**

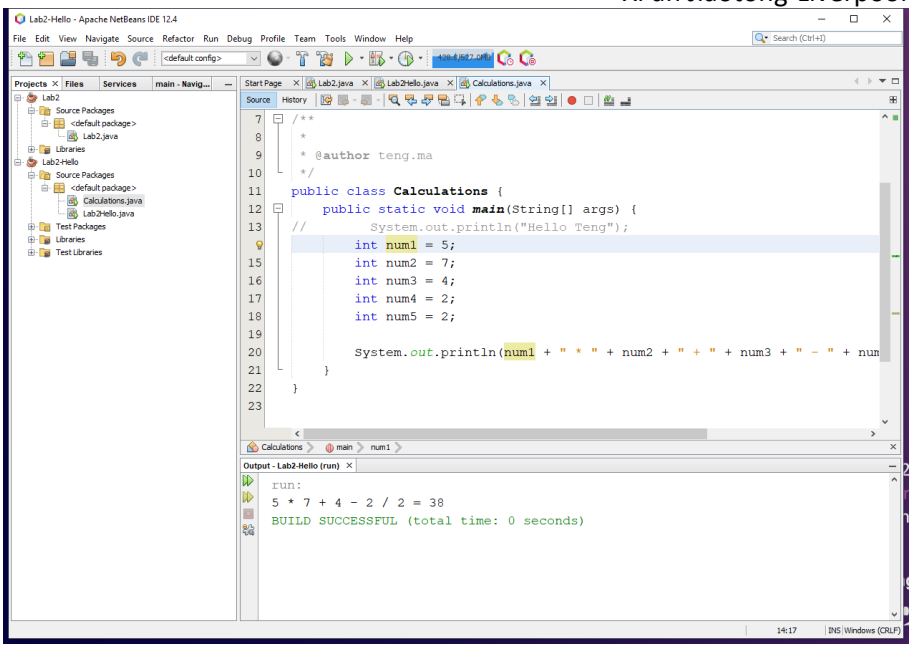


- Once the program is running, you need to change the main method. Add an extra System.out to display a calculation. Make it as simple as complicated as you like. An example is:

```
System.out.println("5 * 7 + 4 - 2 / 2 = 38");
```

- You should create your own! Make sure it displays correctly in the **console**.

- You can use integer variables to store numbers. Similar to your previous program, replace the numbers in your System.out with variables (i.e. int num1 = 5;). This means you must **declare** and **initialise** the variables, and then use them in the System.out.



- This will make it easy to change the numbers. Experiment with changing your number variables to make sure they display on screen. Does it look the same with the variables?

- You can concatenate your variables and text together with a +, i.e. `num1 +" * "+ num2`

  Hint, you may need to use + " " + spaces to make it look nice!

- One thing you should have noticed is that when you change the numbers, the answer is not changed. This is because you are not doing the calculation. You need to create an additional variable that will do the calculation. For the sum above, it should look like:

  ```
  int calculation = num1 * num2 + num3 - num4 / num5;
  ```

- Remember, you should use your own calculation! Change your System.out to display the result of the calculation. Did it calculate correctly?

- Experiment by trying some different numbers. Congratulations, you have made your first calculation in Java!

Creating and Using Variables

- In your lab, you created a class called Lab2Hello, which looked like:

```
Hello Teng

My name is Teng, and I'm a lecturer in CPT

My name is Teng, my school was UoN, and my hometown is Zhengzhou.

Name: Teng, Age: 30, Student number: 12345678

…
```

- You'll notice that my name is repeated 4 times in the above text. If I wanted to change it, I COULD change all the code each time. This could be difficult in a big program and could

cause errors if I forgot to change one.

- By creating a variable and using it, we can save time, and prevent errors.

- **Create** and **initialise** a new String variable, at the start of the main method, and give it a value equal to your name

- Remember, a variable must have a type (in this case, a String), and a useful name. Something like "myName".

```
String myName = "Teng";
```

- Replace your name in the System.outs with your new variable.

- You can concatenate your strings together with a +, i.e.

```
System.out.println("Hello " + "Teng");
System.out.println("Hello "+ myName);
```

- Now create int (for numbers) and other String variables to cover all the different bits of information, such as ID number, Major, Hometown etc.

- You should have a series of statements with int and String variables combined in the System.outs. Run it to ensure it looks correct.

- The big test…. Can you replace all the variable values with the details of a classmate, and have your program still display output that makes sense?

- To do this, replace the value of your name variable with your classmate's name, replace your interest with their interest, and run the program.

More Arithmetic in Java

- In your lab 2, you created a class called Calculations, which contained a calculation such as

```
int calculation = num1 * num2 + num3 - num4 / num5;
```

- Modify your Calculations class further to add 3 extra calculations using the same variables. Calculate the answer on paper first.

- Store each of your calculations in a different variable and display them in a System.out, with an output such as "The answer for calculation one is…"

- Check that your answer matches your calculation. If not, try to learn why.

- Try a div calculation, something like num1/num2. Experiment with several numbers. Do they give you the answer you expect? If not, why not? If you are not sure, in the lab.

- Try 1 / 2, 18 / 5, as well as other numbers.