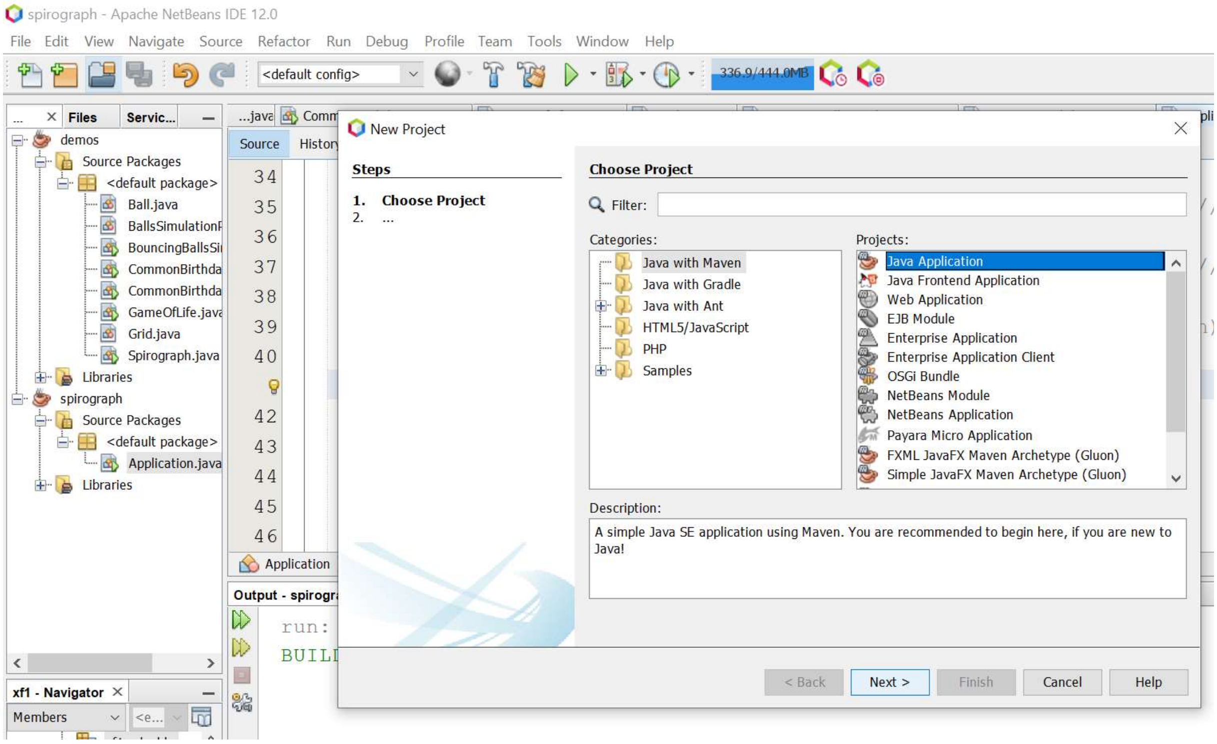
Practical 2: First Java Programs

Task 1

Let’s write our first Java program using Apache NetBeans.

Launch the Apache NetBeans IDE and start creating a new project by selecting “New Project” from the “File” menu. The following dialog appears:



Steps

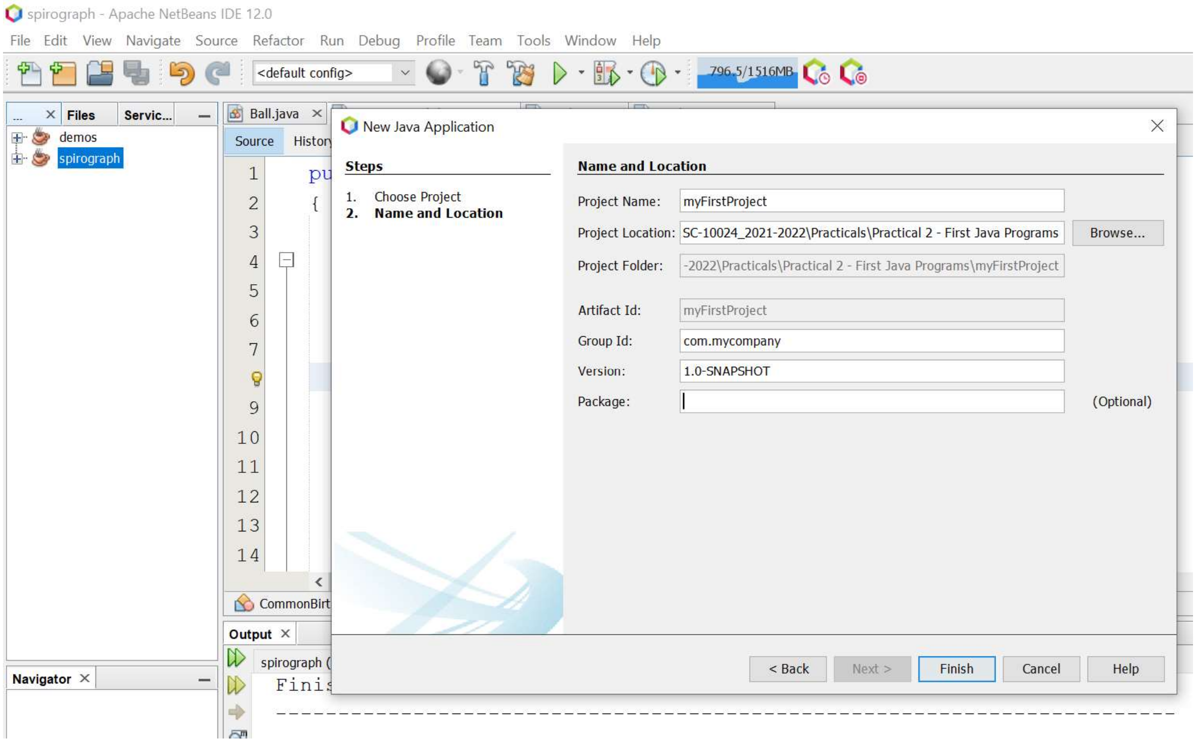
1. Choose Project
2. ...

Description:

A simple Java SE application using Maven. You are recommended to begin here, if you are new to Java!

Select “Java with Maven” category and “Java Application” in projects if they are not already selected and then click “Next”.

The following dialog should appear:



Name and Location

Project Name: myFirstProject

Project Location: SC-10024_2021-2022\Practicals\Practical 2 - First Java Programs

Project Folder: -2022\Practicals\Practical 2 - First Java Programs\myFirstProject

Artifact Id: myFirstProject

Group Id: com.mycompany

Version: 1.0-SNAPSHOT

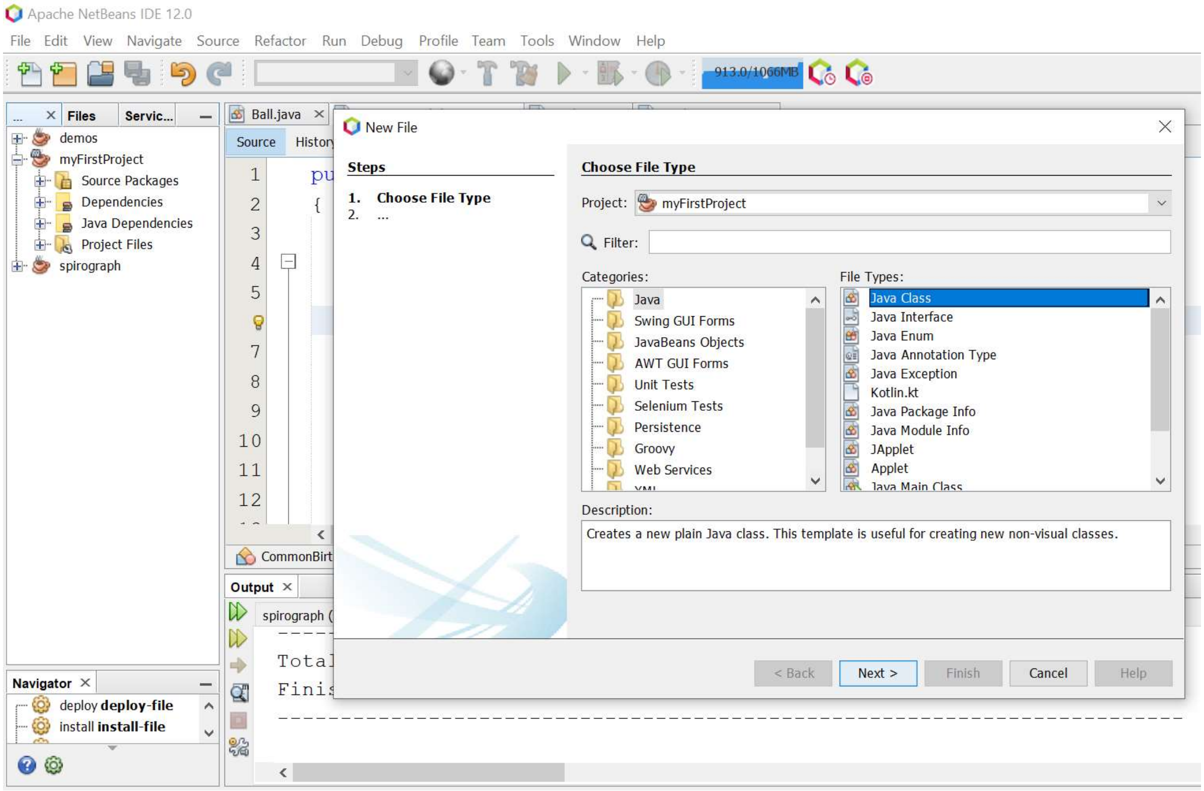
Package: (Optional)

Name your project, for example “myFirstProject” and select the location (i.e. the folder) in which you want to place it. Remove the default Package name “com.mycompany.myfirstproject” and keep all other options same by default. Click “Finish”.

This process will create the necessary sub-folders and configuration/parameter files of your new project. **Please locate the project folder (with your file browser) and have a look at what has been created for you.**

Next, you will need to create your first *class*.

Select “New File” from the “File” menu. The following dialog should appear:



New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: myFirstProject

Filter:

Categories:

- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Selenium Tests
- Persistence
- Groovy
- Web Services
- XML

File Types:

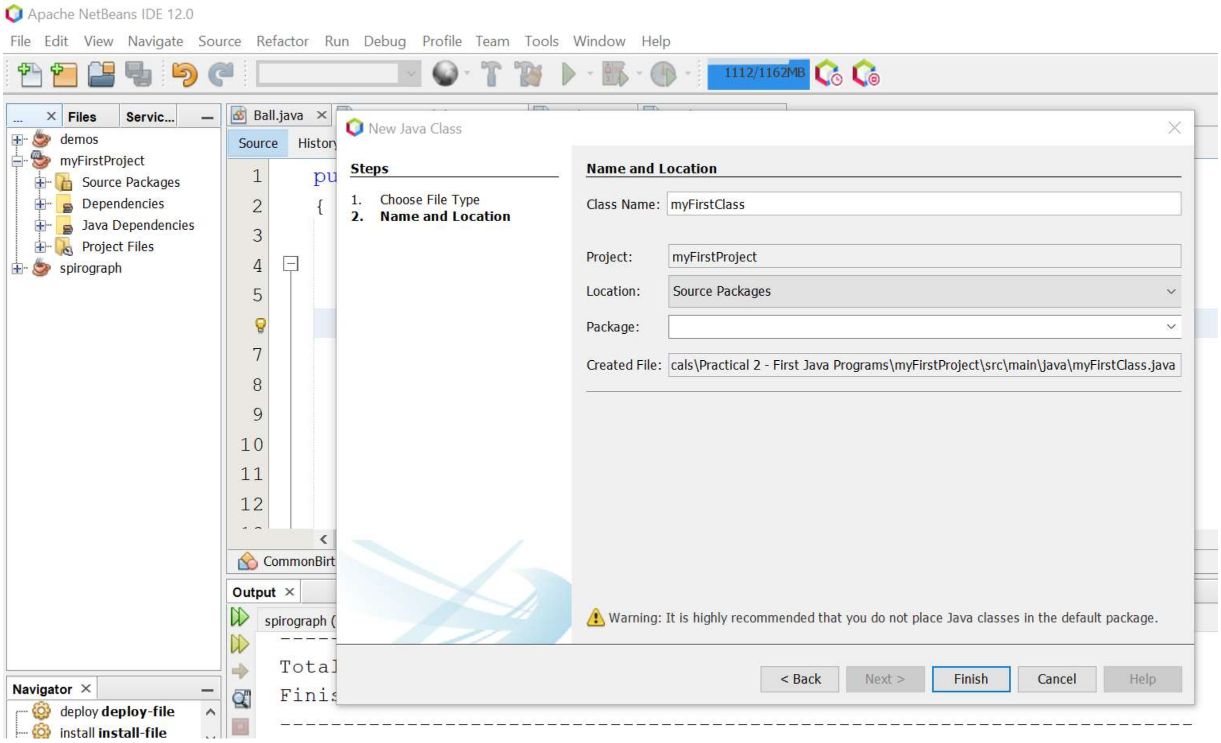
- Java Class
- Java Interface
- Java Enum
- Java Annotation Type
- Java Exception
- Kotlin.kt
- Java Package Info
- Java Module Info
- JApplet
- Applet
- Java Main Class

Description:

Creates a new plain Java class. This template is useful for creating new non-visual classes.

Confirm that you will be creating a new file for “myFirstProject” (if you gave it that name). This is because Apache NetBeans allows you to have more than one project open.

Choose the “Java” category and the “Java Class” file type and then click “Next” for the following dialog:



Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: myFirstClass

Project: myFirstProject

Location: Source Packages

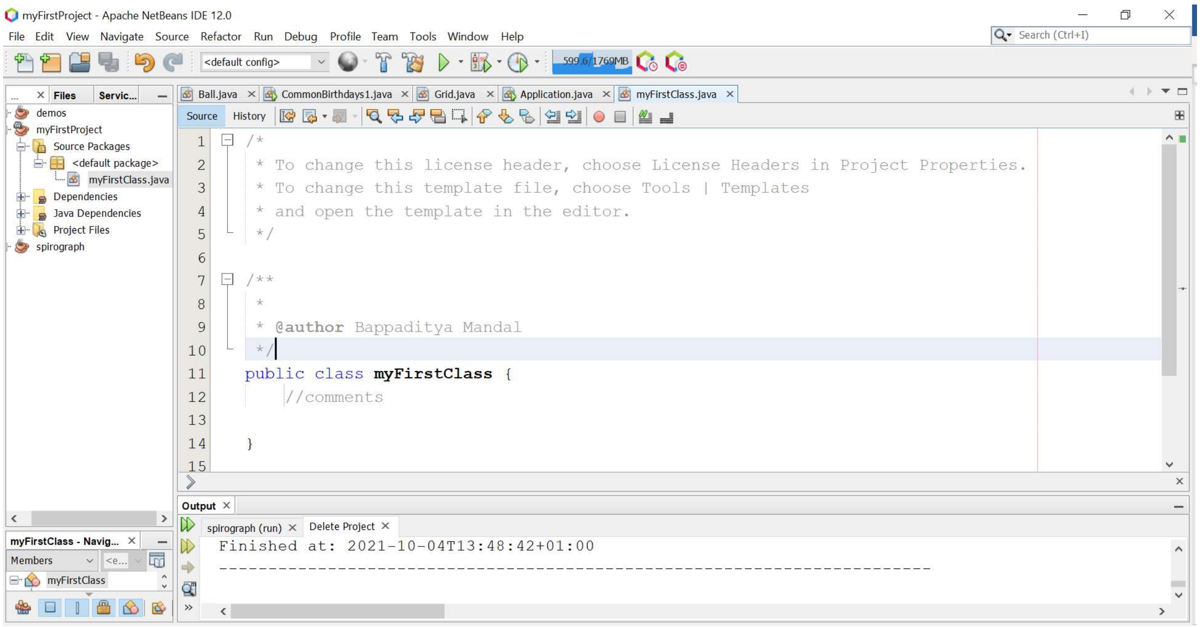
Package:

Created File: cals\Practical 2 - First Java Programs\myFirstProject\src\main\java\myFirstClass.java

Warning: It is highly recommended that you do not place Java classes in the default package.

Name your new class, for example “myFirstClass” and then click “Finish”.

The class file will be created and opened for you to edit. Notice that there are some contents in it already:



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

/**
 *
 * @author Bappaditya Mandal
 */
public class myFirstClass {
    //comments

}
```

```
Finished at: 2021-10-04T13:48:42+01:00
------------------------------------------------------------------------
```

The greyed-out lines at the start are *comments*. Comments are ignored by the compiler and they are a convenient way to annotate your code with your own notes. Anything between * and */ as well as lines starting with // are comments.

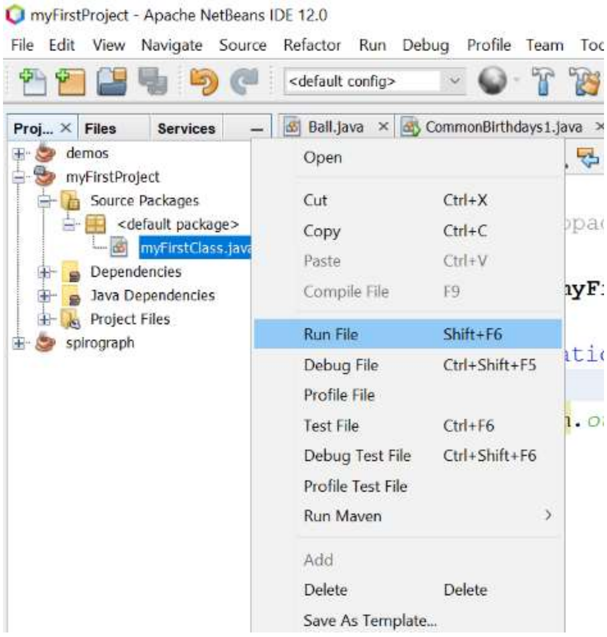
The executable code of the class starts with:

```
public class myFirstClass
```

Append the code to look like this:

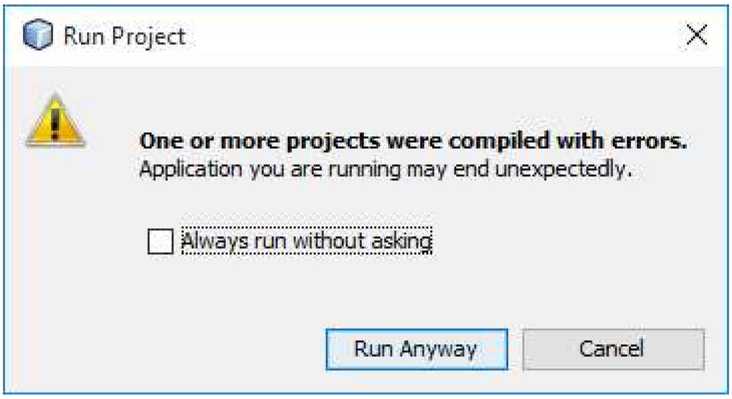
```
public class myFirstClass
{
    public static void main(String[] args) {

        System.out.println("Hello world!");

    }
}
```

Save the changes by pressing save button on top left "Save All" or keyboard 'Ctrl + S'. Any changes you have made in the file you should always save them. If you don't need any line of code comment them using "//". After you have typed the above code right-click on the newly created class file in the project tree panel and select "Run file" to execute your program:



If you have not introduced any *syntax errors* in the above code, your program will compile successfully and the result will be displayed in the output panel at the bottom (otherwise called the *console*) along with the message "BUILD SUCCESSFUL".

If your code contains syntax errors however, you will be presented with the following dialog:



One or more projects were compiled with errors.
Application you are running may end unexpectedly.

□ Always run without asking

In this case you should click "Cancel" and revise your code to fix the error before running the project again. If the problem persists, ask for help.

Have a look at the output of your program once it has been compiled and run successfully. Which line in your code do you think is primarily responsible for the output? What do you think the `System.out.println()` *method* does?

Task 2

Replace the line:

```
    System.out.println("Hello world!");
```

with:

```
    System.out.println("Hello there!");
```

Run your changed program to see the effect of your change on the output.

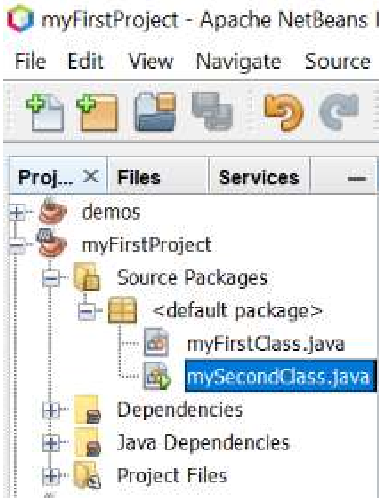
Task 3

Below is another program to try. **Try to guess what the output of this program before you run it.**

Instead of changing your previous class, create a new one. Make sure however that you give it a different name (for example `mySecondClass`) to distinguish between the two.

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

            String name;
            name = "Bappa";
            System.out.println("My name is " + name);

        }
    }
```

Again, to run the new class, you need to right-click on its file in the project's tree and select "Run File":



Task 4

And here below is yet another program to try (in a new class). **Again, try to guess what the output will be before you compile and run it.**

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

        int age;
        age = 20;
        System.out.print("I wish I w");
        System.out.println("as " + age + " years old.");

    }
}
```

Get into the habit of studying your programs before and after you run them. Always try to predict the outcome and if the true outcome contradicts your prediction try to understand why by looking back at the code. This is the best way to learn programming.

Task 5

Add the following lines after the `System.out.println(...)` line above. **Guess what the output will be before you run your program.**

```
System.out.print("I like to say that");
System.out.print(" I am " + age + " and " + 17);
System.out.println(" years old");

System.out.print("...but I am ");
System.out.print("actually " + age + 17);
System.out.println(" years young!");

System.out.print("I couldn’t be really. I am");
System.out.print(", truly and honestly, ");
System.out.println(age + 17 + " years young!");
```

You are bound to have surprises. Please feel free to ask a demonstrator for an explanation.

Task 6

Make a list of what you have learned today. Why don’t you create your own programs or make changes to the above examples to practice what you have learned.