Middlesex University London

Lecture 1

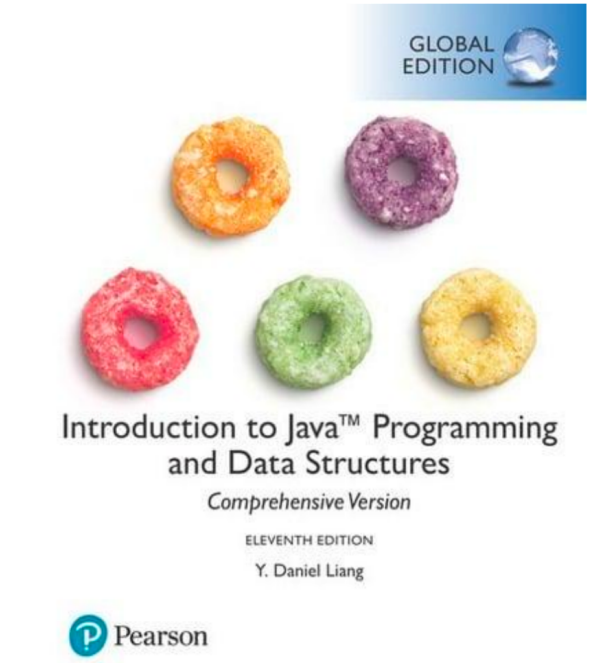
Introduction to the module and Java programming language

PURPOSE OF THIS MODULE

- This module introduces fundamental computational concepts and programing constructs relevant to understand and use of range of widely used programming languages.
- The module is mainly based on book:
- **"Introduction to Java Programming and Data Structure"**

  Eleventh Edition by Y. Daniel Liang

  (Available online via university library website)



STRUCTURE OF THE MODULE

Fundamentals of Programming

1. Introduction to 1st Java Program
2. Input -Output
3. Primitive data type
4. Selections
5. Loops
6. Revision
7. Methods

Object Oriented thinking

8. Array
9. 2D Array
10. Classes
11. OOP

User Interface and Error Handling

12. Inheritance Polymorphism, Array List
13. Exception handling
14. JavaFX
15. JavaFX

Input Process Storage Output

16. File writing, reading. Binary I/O
17. Networking

EXERCISES FOR EACH CHAPTER

Exercises from the book and sobs from the Workbook are prepared for this module and available on MyLearning (often with code examples), but we will use them ***only to practice and improve your programming skills, not to pass the module.***

They are ordered based on difficulty:

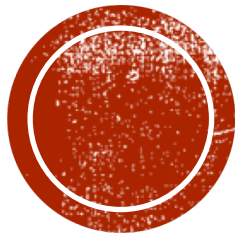
- Threshold
- Typical
- Excellent

BEFORE WE START WITH JAVA!

First download and install Java development kit (JDK). It is a package of tools for developing Java based software.

LINK to JDK for this module.
https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

On the website choose **Java Standard Edition** (Java SE) the right version for your OS. For most of you it will be the Windows x64 version or macOS x64.
Java SE is the core Java programing edition. It contains the capabilities needed to develop desktop and server applications.



INTRODUCTION TO JAVA

THIS WEEK:

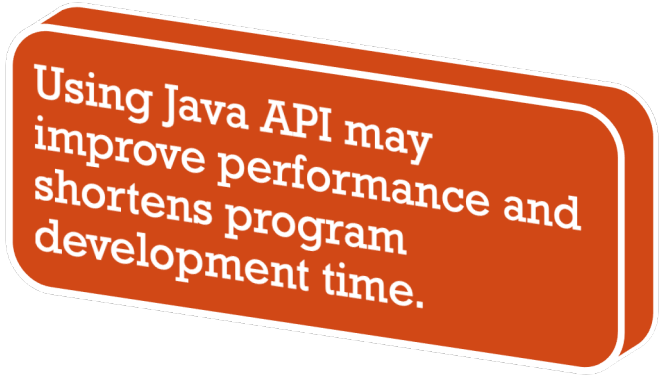
- What is Java
- JDK, API, IDE
- Netbean project
- Java Class
- Java Programs
- Java print, println, printf methods

WHAT IS JAVA?

- One of the most popular Programming languages
- Object Oriented Programming language (OOP)
- Independent from your OS
- Runs on JVM (Java virtual machine)
- Mature language with plenty of development happening

API APPLICATION PROGRAMMING INTERFACES

- Java application programming interface (API) is a list of all classes that are part of the Java development kit (JDK). It includes all Java packages, classes, and interfaces, along with their methods, fields, and constructors. These pre-written classes provide a tremendous amount of functionality to a programmer.
- A programmer should be aware of these classes and should know how to use them. A complete listing of all classes in
- Java API can be found at Oracle's website: http://docs.oracle.com/javase/8/docs/api



IDE - INTEGRATED DEVELOPMENT ENVIRONMENT

- IDE is a program which allows and help write the code. Editing, compiling, building, debugging, and online help are integrated in one graphical user interface. You do not necessary have to use one.

- The most popular Java IDEs are:
  - Eclipse (http://www.eclipse.org)
  - IntelliJ IDEA (http://www.jetbrains.com)
  - NetBeans (http://www.netbeans.org)

NETBEANS - Creating a new project

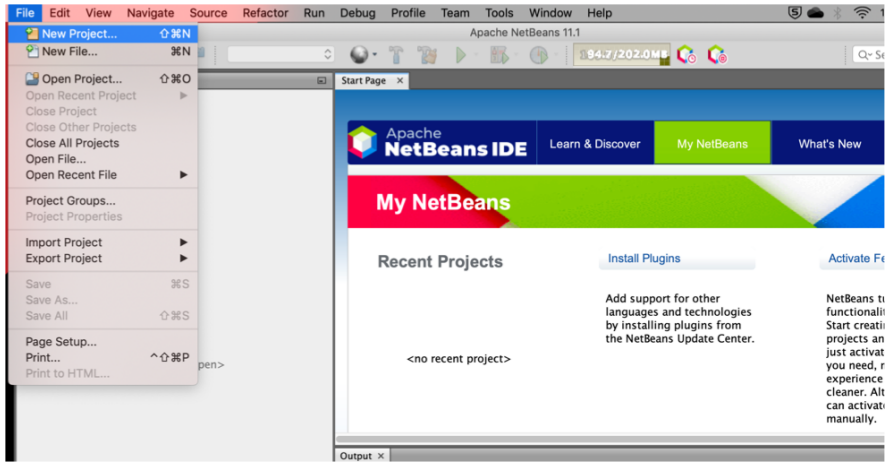


Fig 1. Choose File, New Project

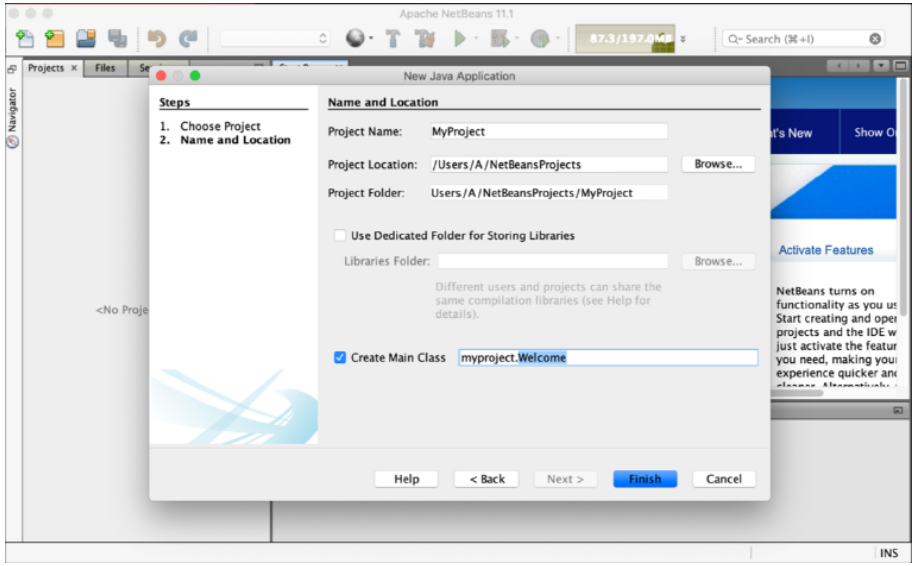


Fig 3. Type name of the project in the Project name, tick the Create main class option, rename the main class field and click Finish

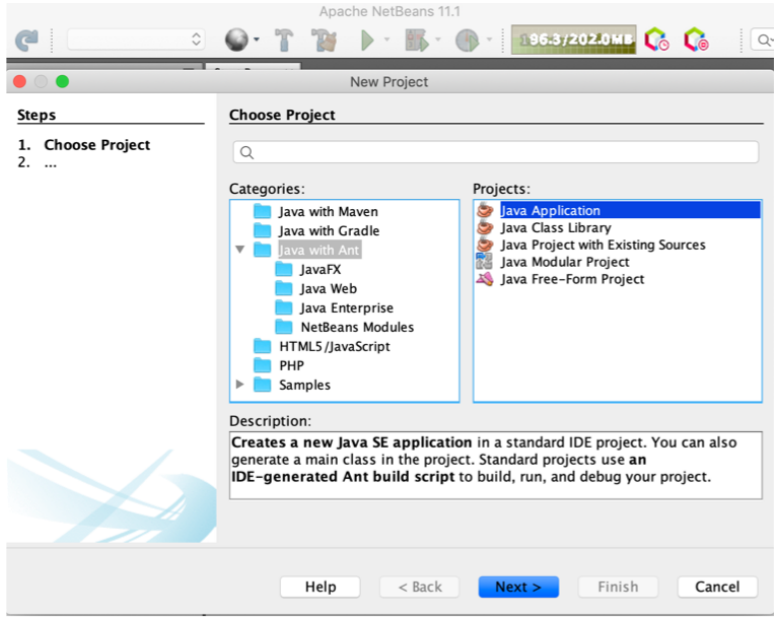


Fig 2. Select Java in the Category section and Java Application in the Project section and then click Next

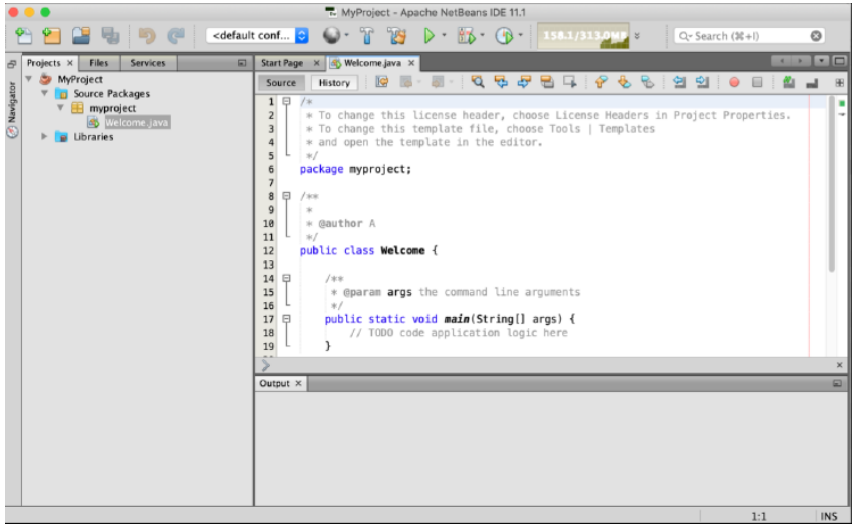
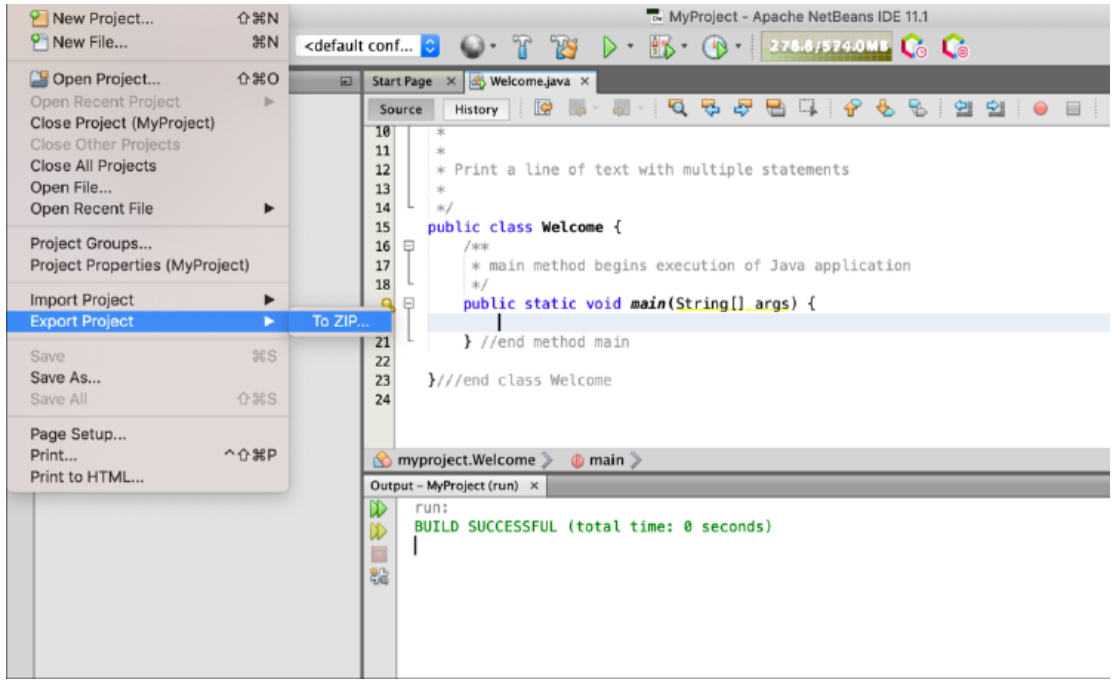
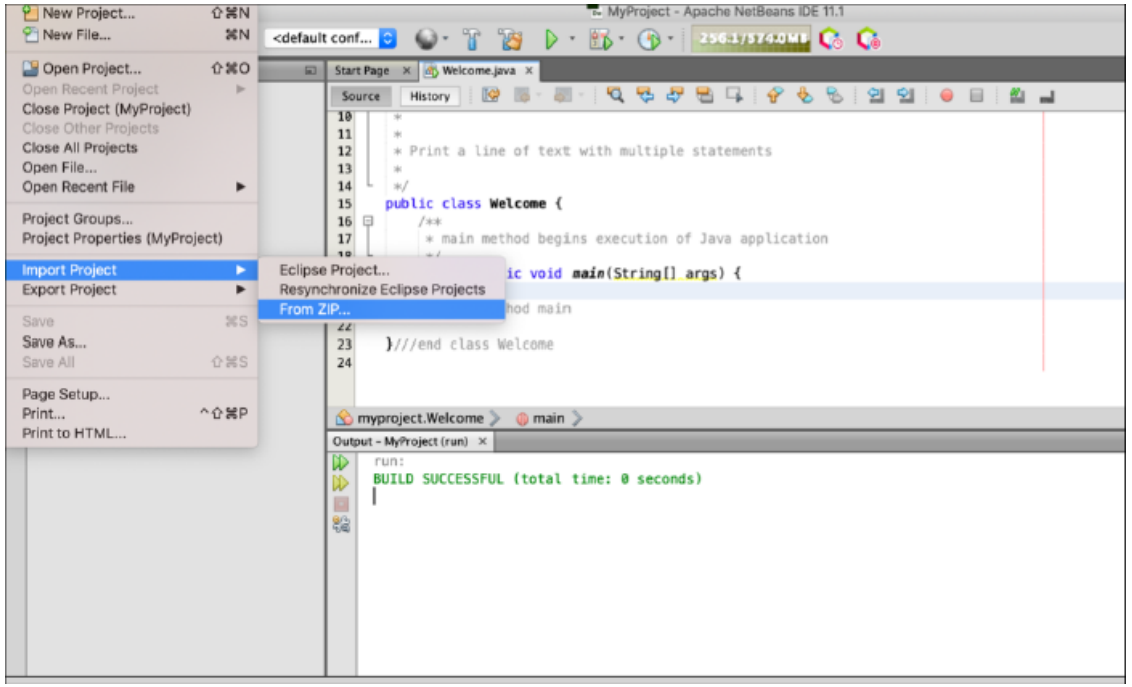


Fig 4. Project name of myproject has been created with a class called Welcome

NETBEANS - EXPORTING/IMPORT POROJECT

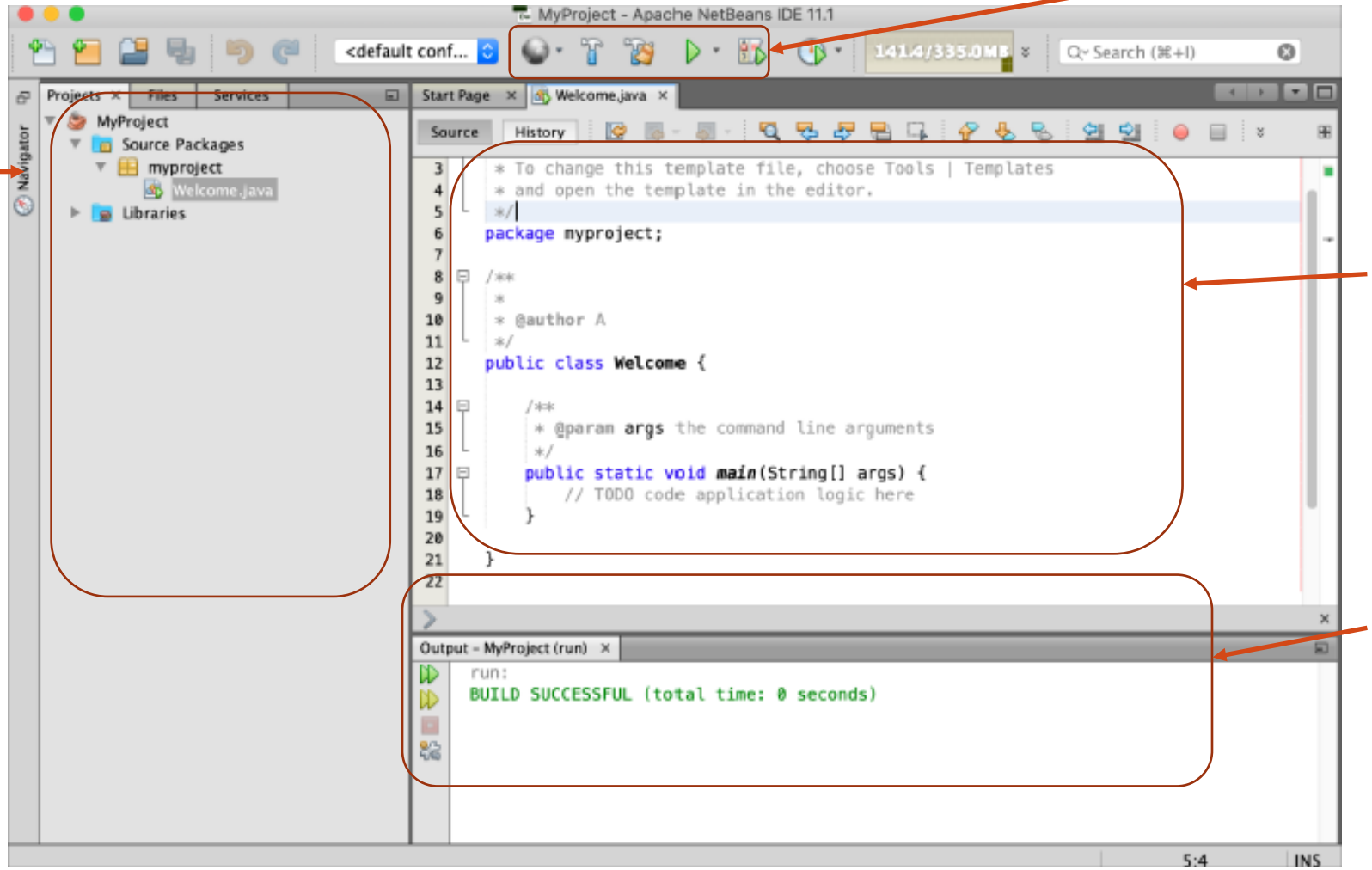




NETBEANS ENVIRONMENT

Control buttons use to compile and start the program

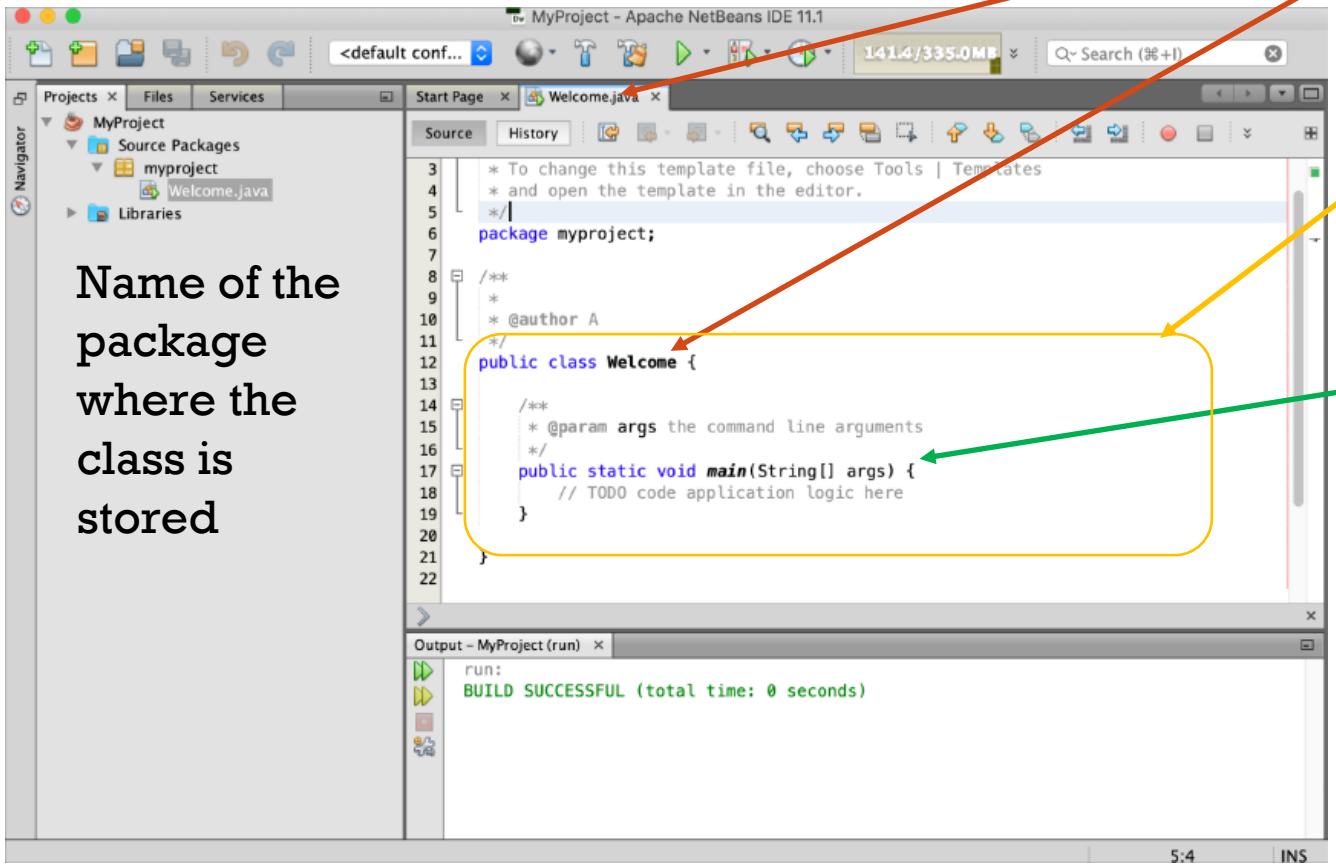
Project window containing project files



Code editor.
This is the place where the code goes

Console/Terminal
Its where the output of the program is displayed

JAVA CLASS



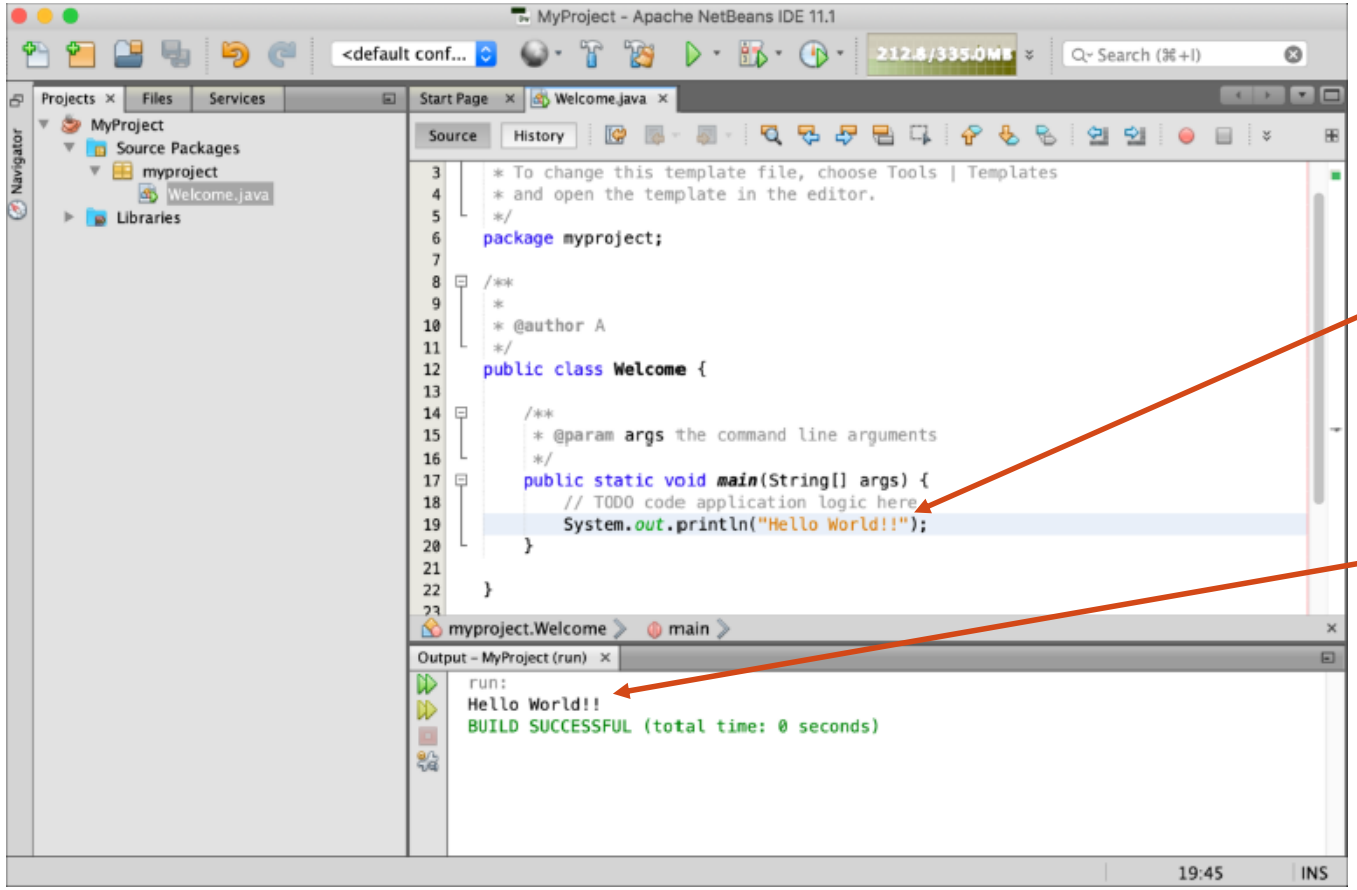
Name of the package where the class is stored

File name (in this case: **Welcome)** should be the same as the class name
It should start with an uppercase letter

{open and close brackets }
begins and ends the class body

The main method is the entry point where program is executed and contains statements

1ST JAVA PROGRAM



Java statement prints 'Hello World!' string of characters to the console

Output from running of the program

PRINT, PRINTLN METHOD

```
 */
public class Welcome {
    /**
     * main method begins execution of Java application
     */
    public static void main(String[] args) {

        System.out.print("Welcome to ");
        System.out.println("Java programming ");
    } //end method main

}///end class Welcome
```

myproject.Welcome › main

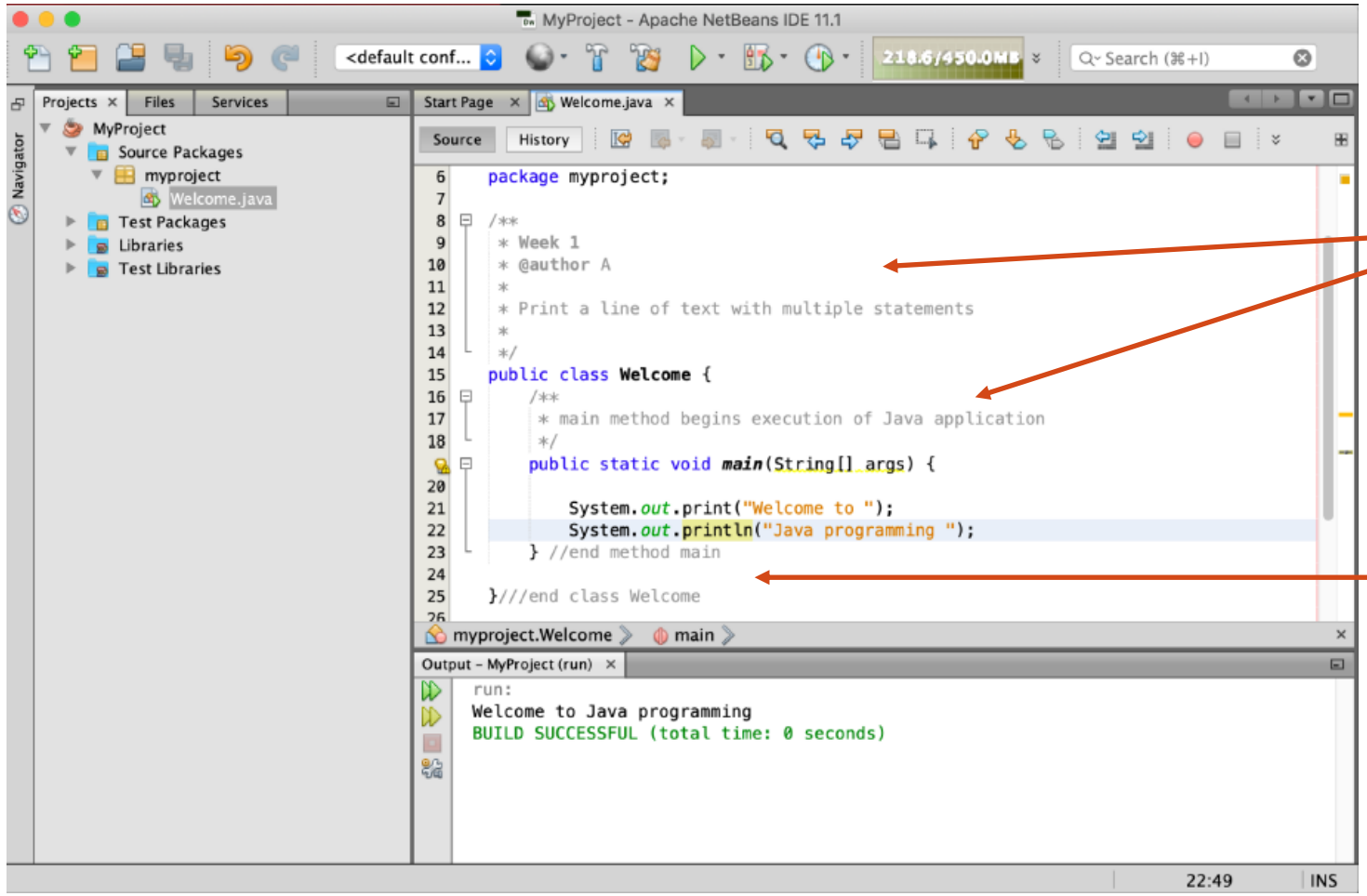
put – MyProject (run)

```
run:
Welcome to Java programming
BUILD SUCCESSFUL (total time: 0 seconds)
```

print- is used to display the text *Without* the line separator.

println - is a statement used for displaying the program output in text format using a console/terminal *with* the line separator.

JAVA COMMENTS



Multi-line comments

Single line comments

COMMON ESCAPE SEQUENCES

Escape sequence	Description
\n	Newline. (Linefeed)
\t	Tab.
\r	Carriage return.
\\	Backslash
\”	Double quote

EXAMPLE OF ESCAPE SEQUENCING

```
14   */
15  public class Welcome {
16      /**
17       * main method begins execution of Java application
18       */
19      public static void main(String[] args) {
20
21          System.out.print("Welcome to \n");
22          System.out.print("Java\r programming ");
23      } //end method main
24
25  }///end class Welcome
26
```

Output – MyProject (run)

```
run:
Welcome to
 programming BUILD SUCCESSFUL (total time: 0 seconds)
```

PRINTLN FOR ARITHMETIC OPERATIONS

```
public class Welcome {
    /**
     * main method begins execution of Java application
     */
    public static void main(String[] args) {
        System.out.println(5==8);
        System.out.println(4+7);
        System.out.println(5%3);
        System.out.println(2-6);
    } //end method main

}///end class Welcome
```

put – MyProject (run) ×

```
run:
false
11
2
-4
BUILD SUCCESSFUL (total time: 0 seconds)
```

PRINTF METHOD

```
public class Welcome {
    /**
     * main method begins execution of Java application
     */
    public static void main(String[] args) {
        System.out.printf("%s%n%s", "Welcome to ", "Java Programming");
    } //end method main

}///end class Welcome
```

printf - is used to display format data

In this case *%s* is a place holder for a string

```
ut – MyProject (run)
run:
Welcome to
Java ProgrammingBUILD SUCCESSFUL (total time: 0 seconds)
```

PRINTF FORMAT SPECIFIERS:

Place holder	description
%c	Character (Unicode character)
%d	decimal (integer) number (base 10)
%e	exponential floating-point number (float, double)
%f	floating-point number
%i	integer (base 10)
%o	octal number (base 8)
%s	a string of characters
%u	unsigned decimal (integer) number
%x	number in hexadecimal (base 16)
%%	print a percent sign
\%	print a percent sign
%n	the new line character

Can be used later on in the course

KEY THINGS YOU SHOULD UNDERSTAND AFTER WEEK 1

- Compile and run a basic class in NetBeans
- Understand the basic class structure with a main method
- Knowing how to print a line to the console
- Java is a 'strict' language in terms of syntax
- SOBs 1-4

MASTERING YOUR SKILLS

- Make sure that your SOBs for this week are finish
- Read chapter 1 from the Liang book
- Do exercises 1.1 to 1.17 from pages 52, 53 respectivly.