CS 152: Data Structures with Java
Hello World with the IntelliJ IDE

Instructor: **Joel Castellanos**

e-mail: joel.unm.edu

Web: http://cs.unm.edu/~joel/

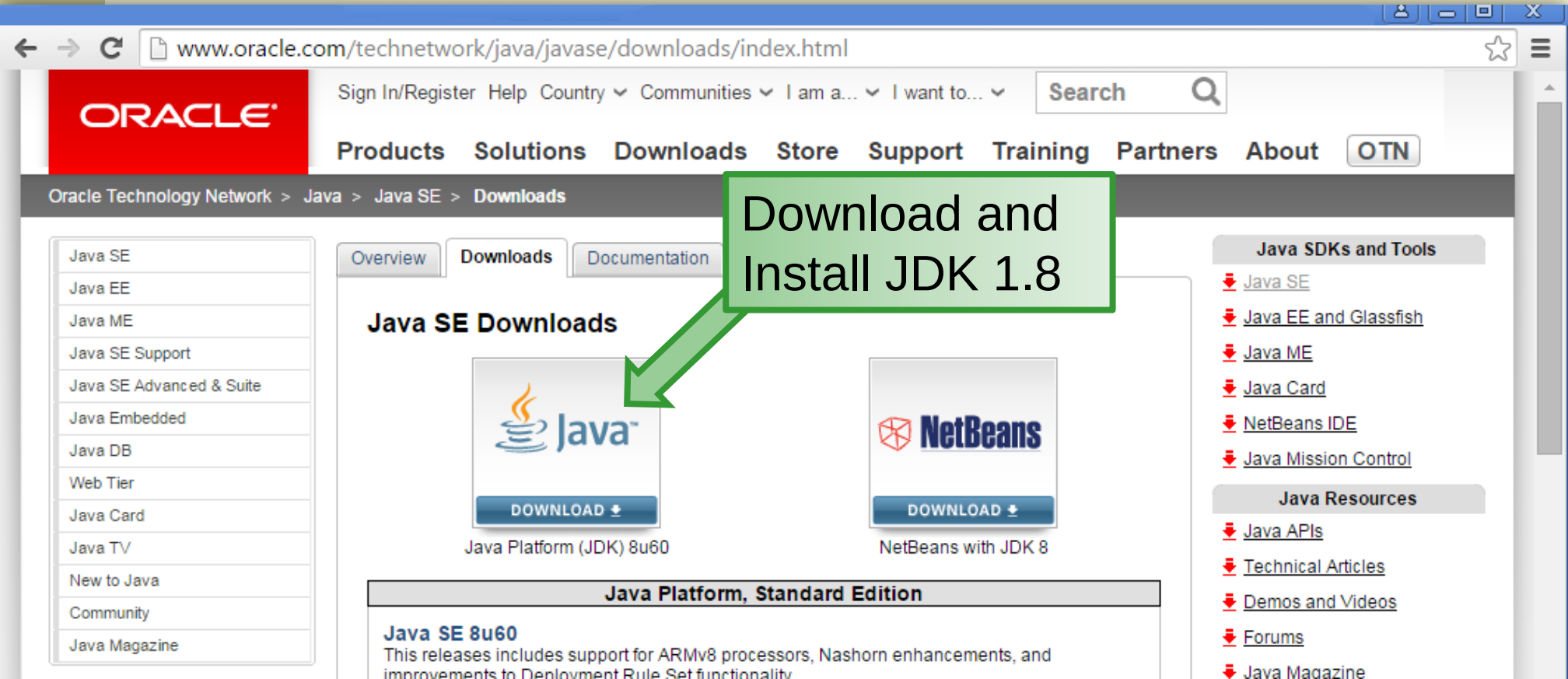
Office: Electrical and Computer Engineering building (ECE). Room 233

8/21/2017

Install Java Development Kit (JDK) 1.8

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Most computers already have the ***Java Runtime Environment*** installed. However, to create Java programs, the ***Java Development Kit*** is required. Note: the JDK includes a copy of the matching version of the JRE.

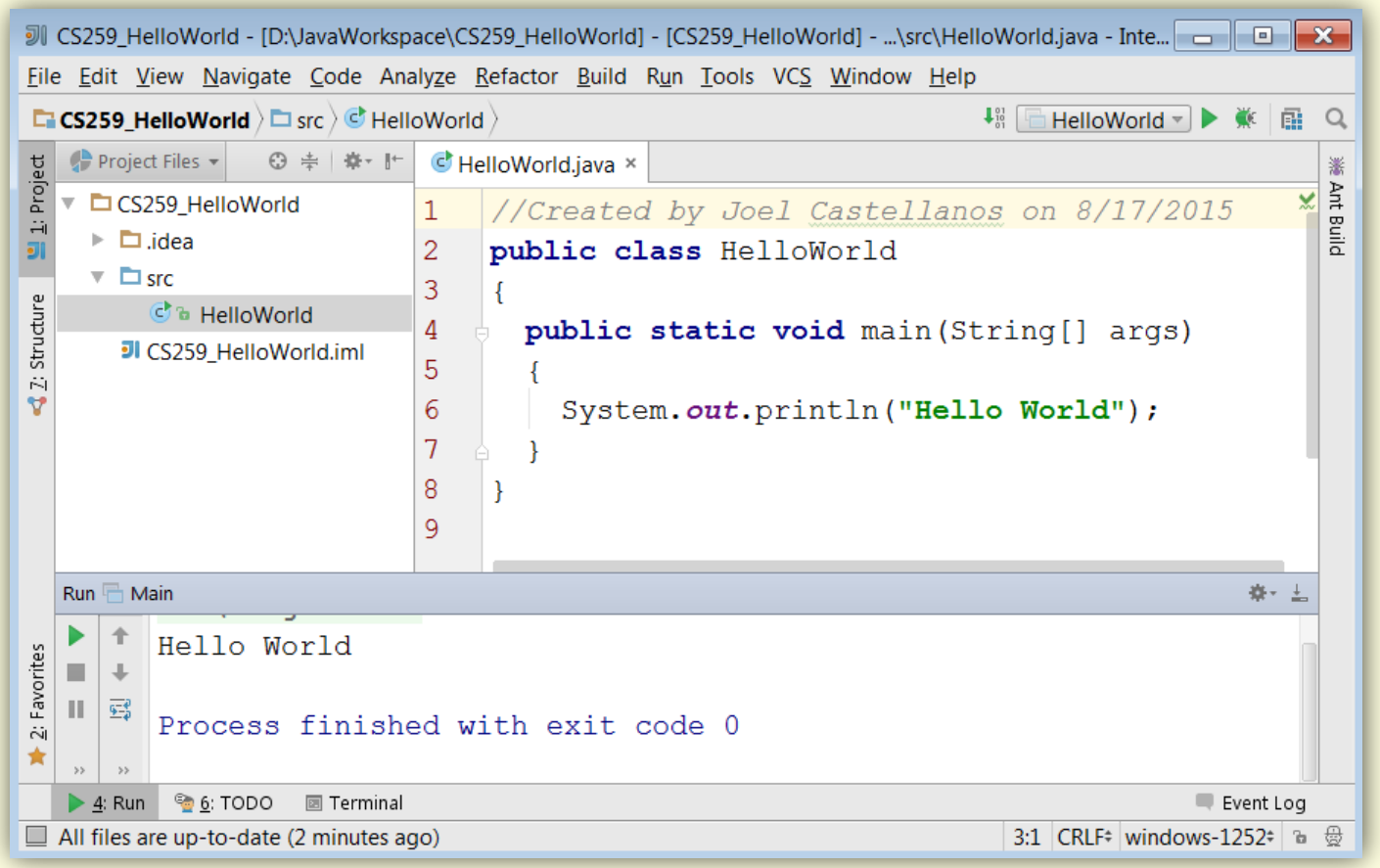


IntelliJ IDE (Integrated Development Environment)

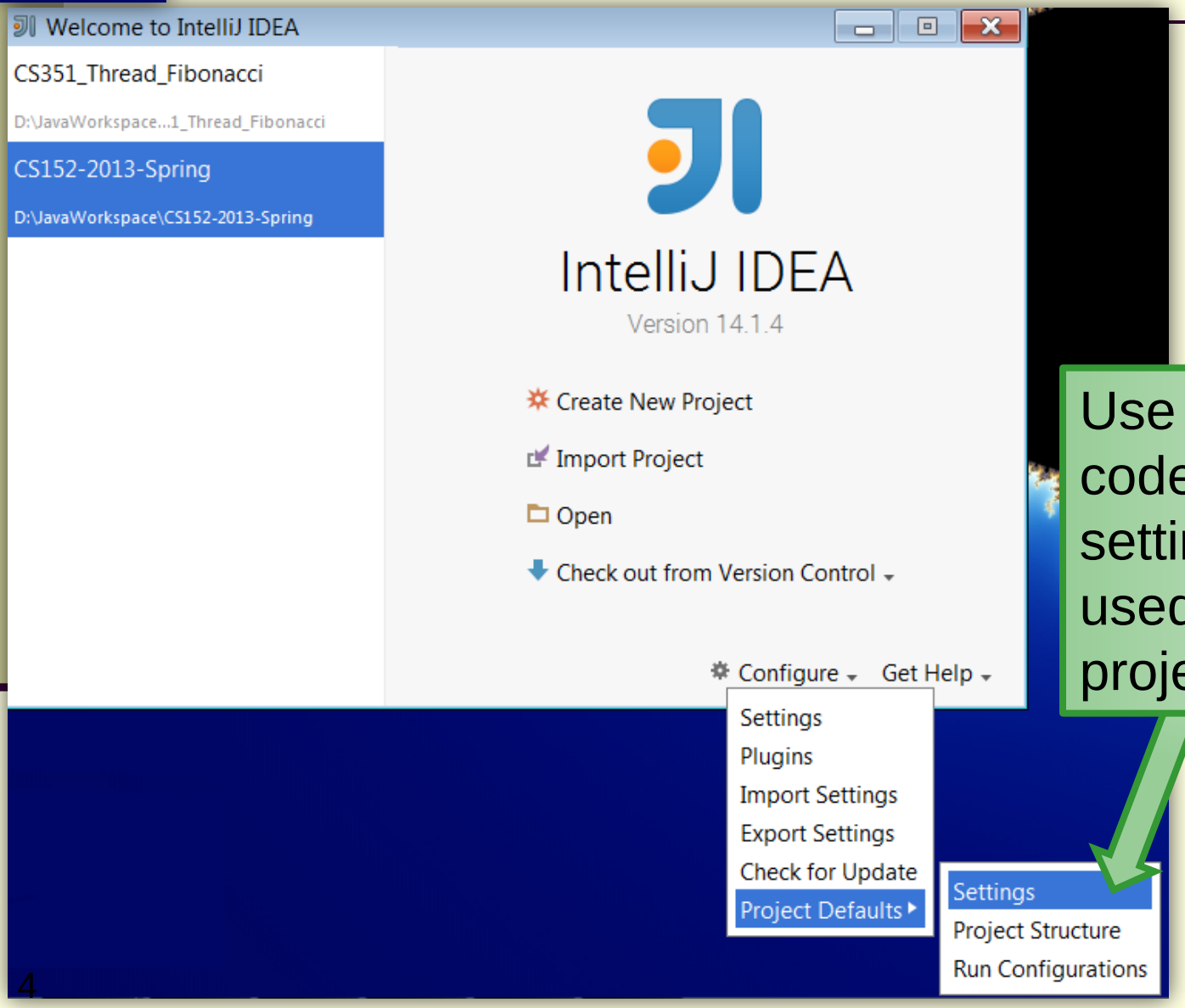
- https://www.jetbrains.com/idea/download/
- IntelliJ Community Edition (free) version 14.1.

Java is a ***Programming Language***.

IntelliJ is an ***Integrated Development Environment***.

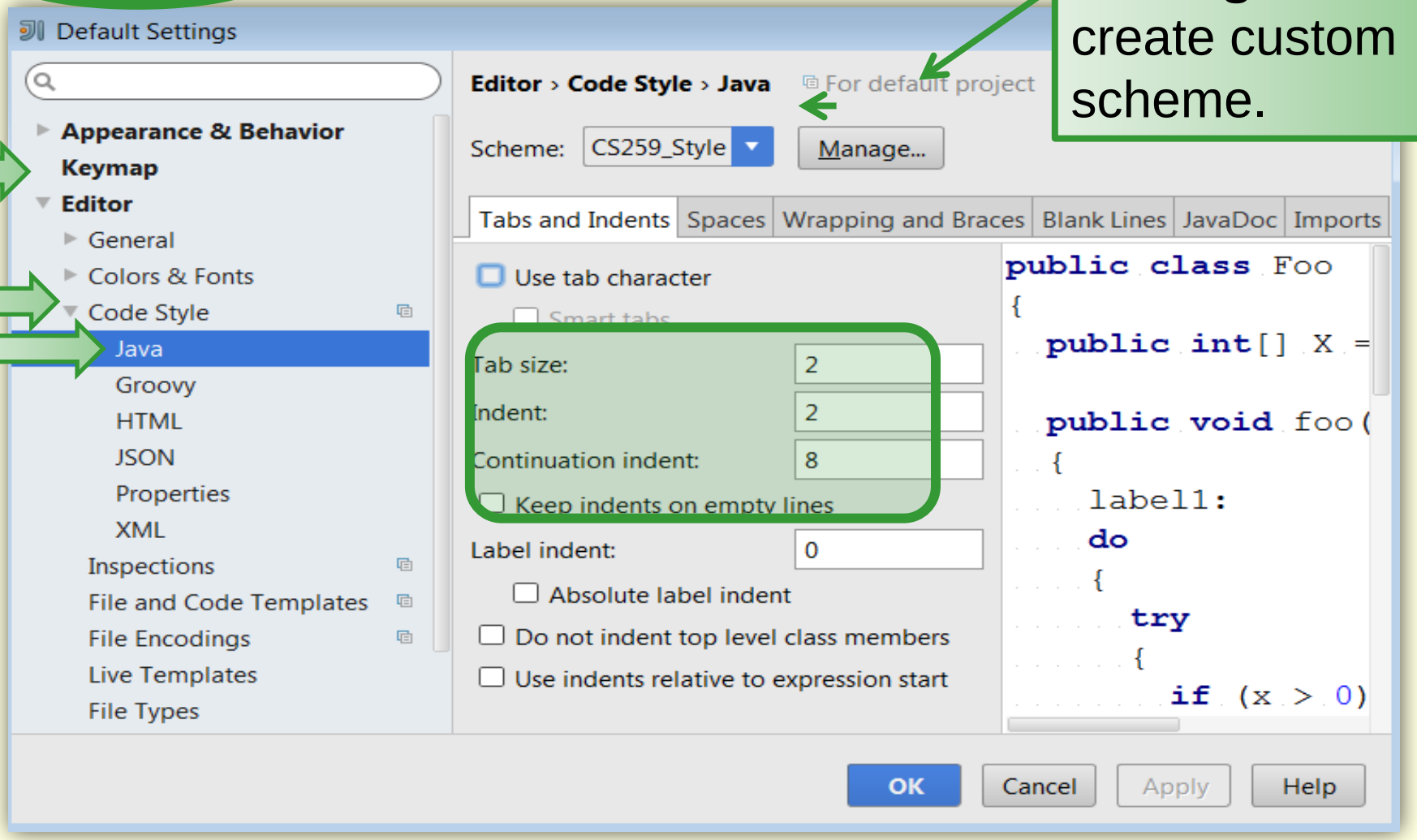


Project Default Settings (Code Style)

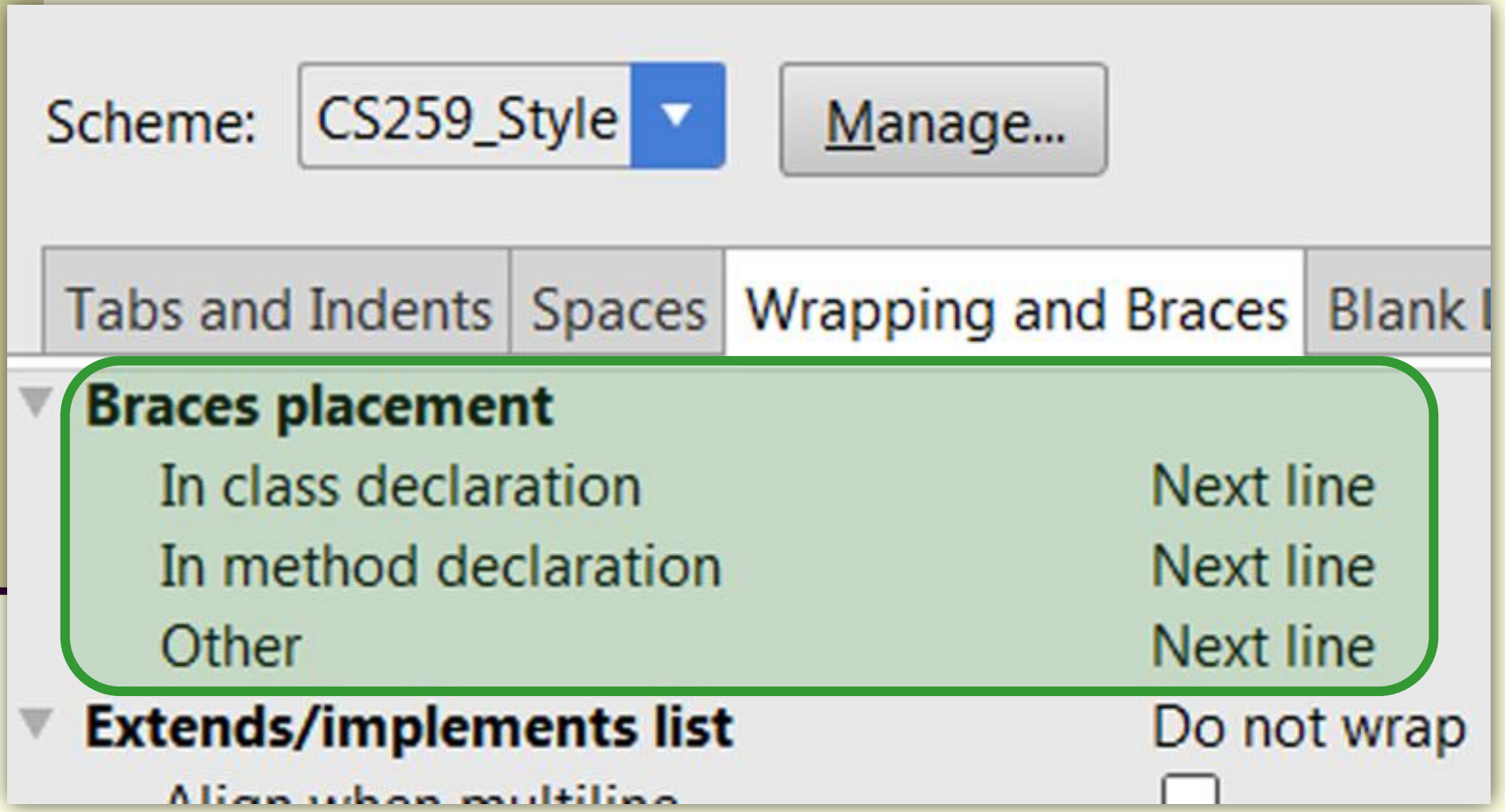


Use to change the code style and other settings that will be used for all new projects created.

Code Style: IntelliJ → File → Settings…

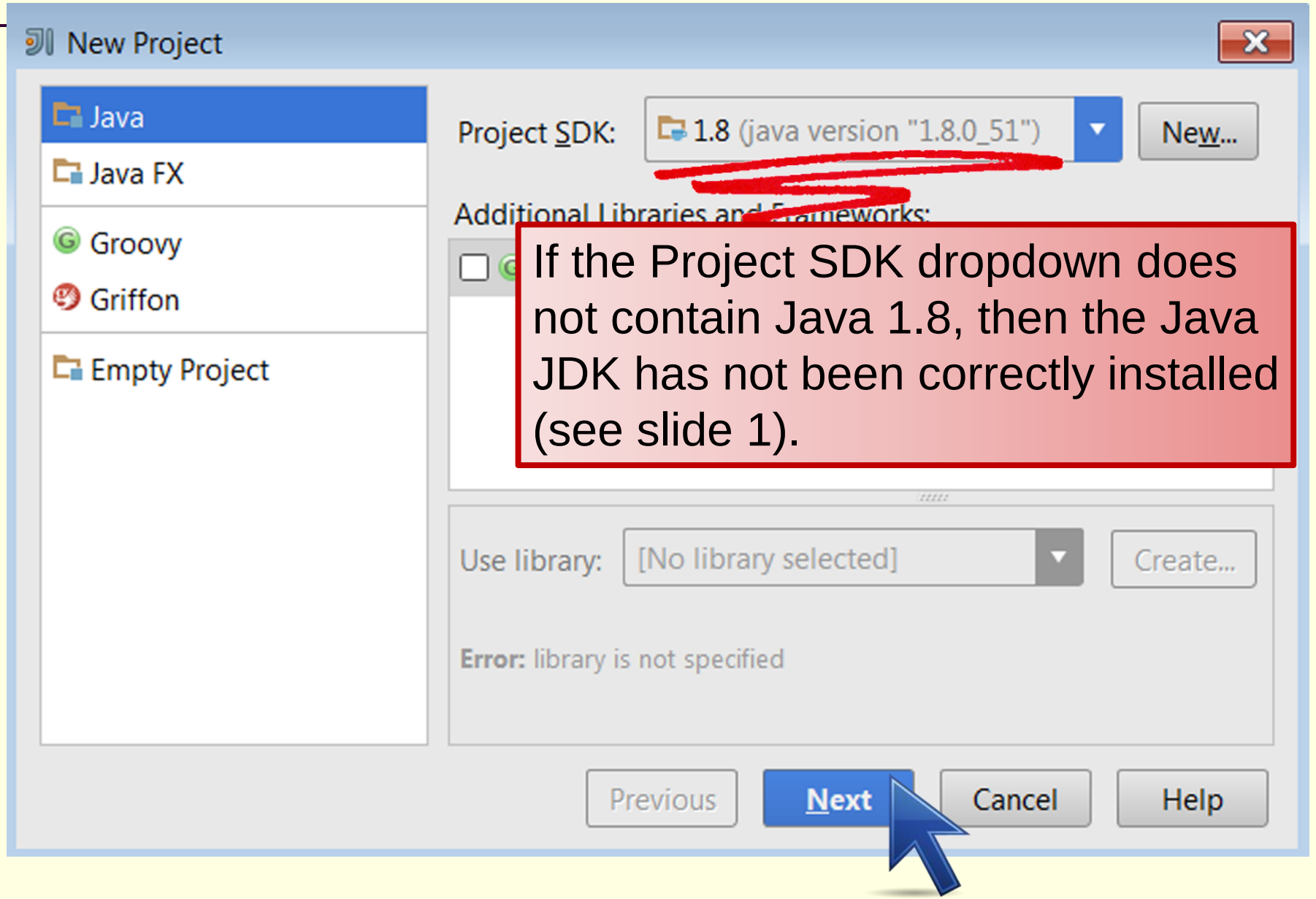


Code Style: Braces

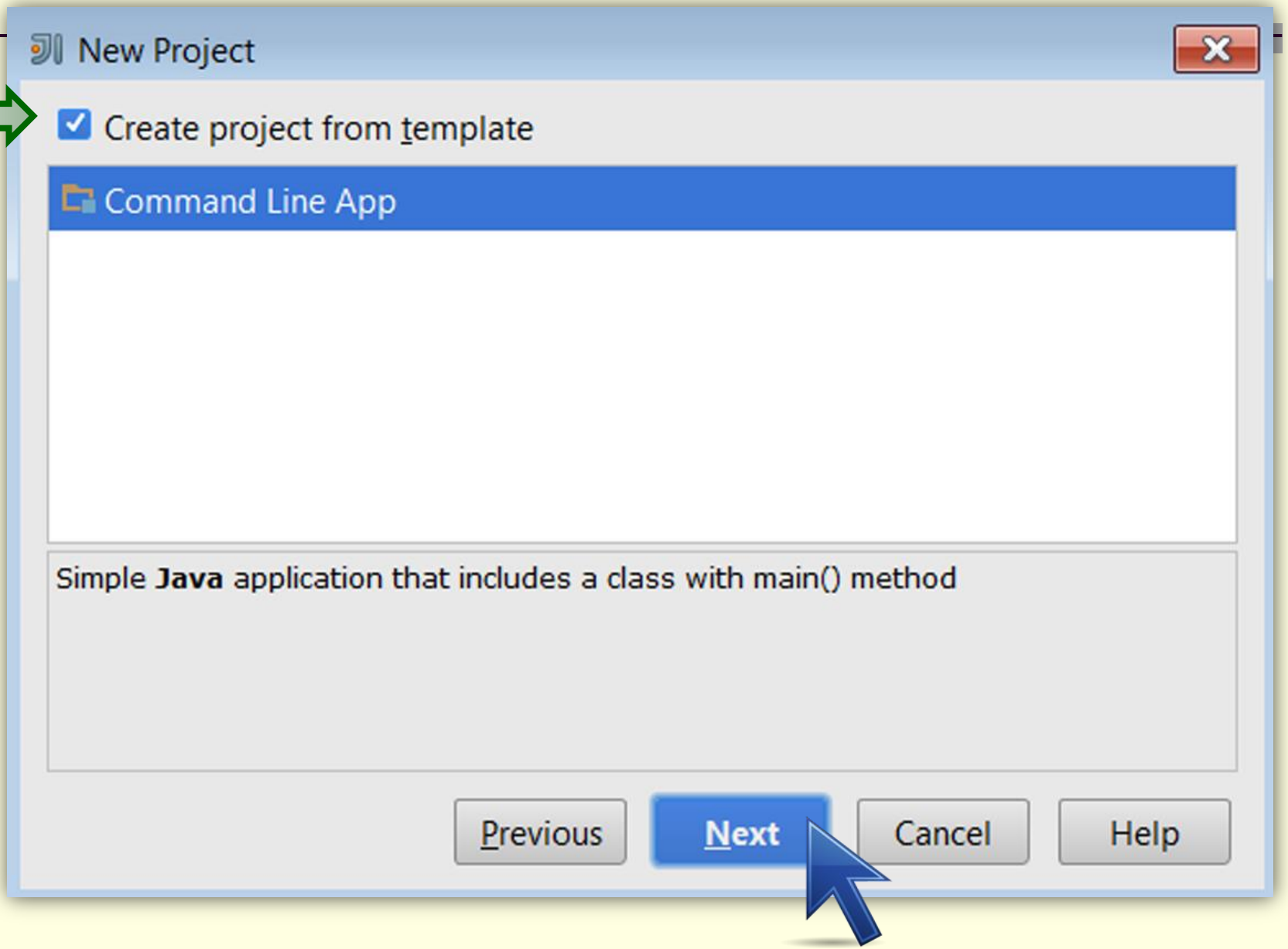


IntelliJ: Create New Project (1 of 5)

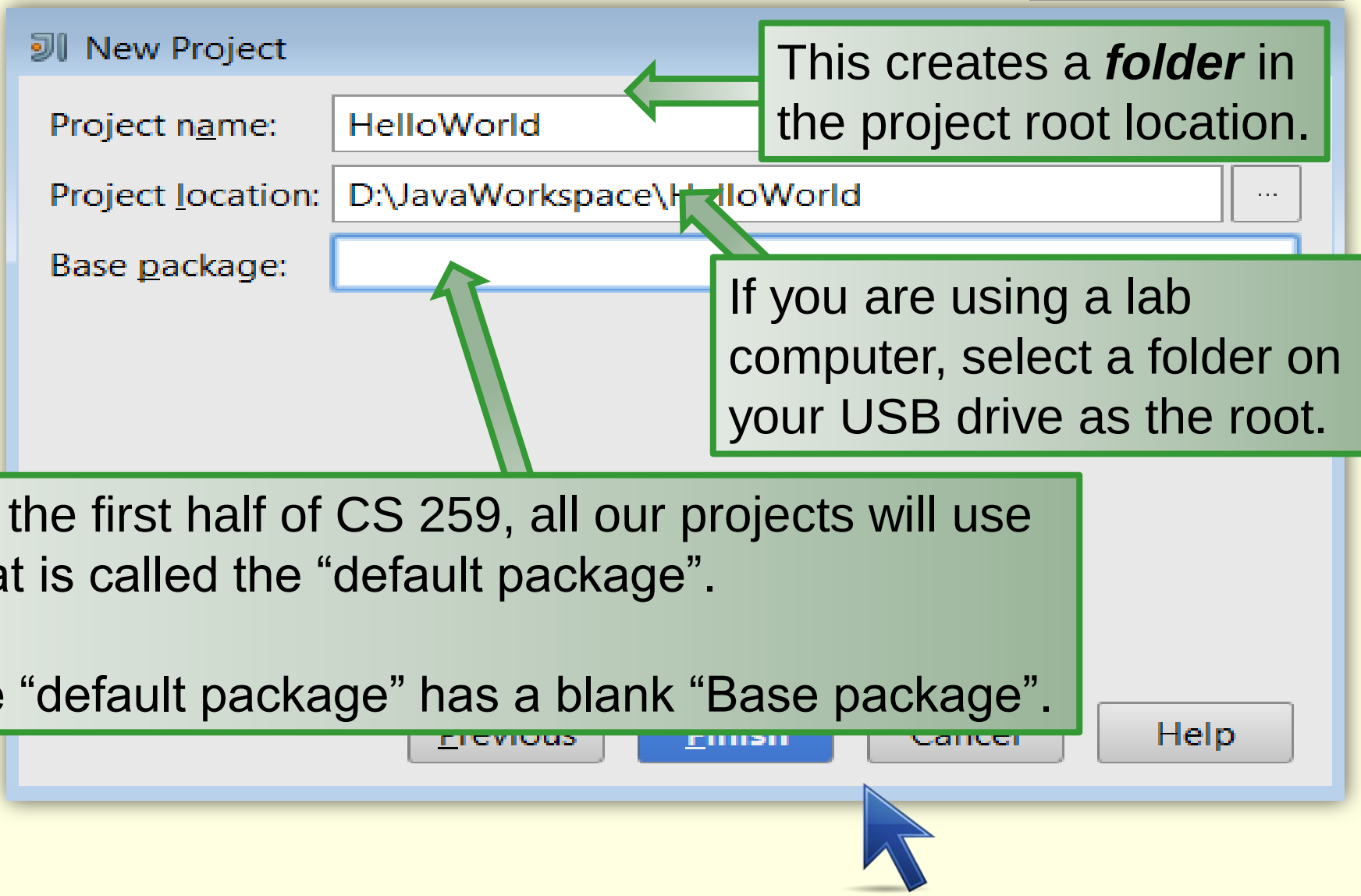
IntelliJ: Create New Project (2 of 5)



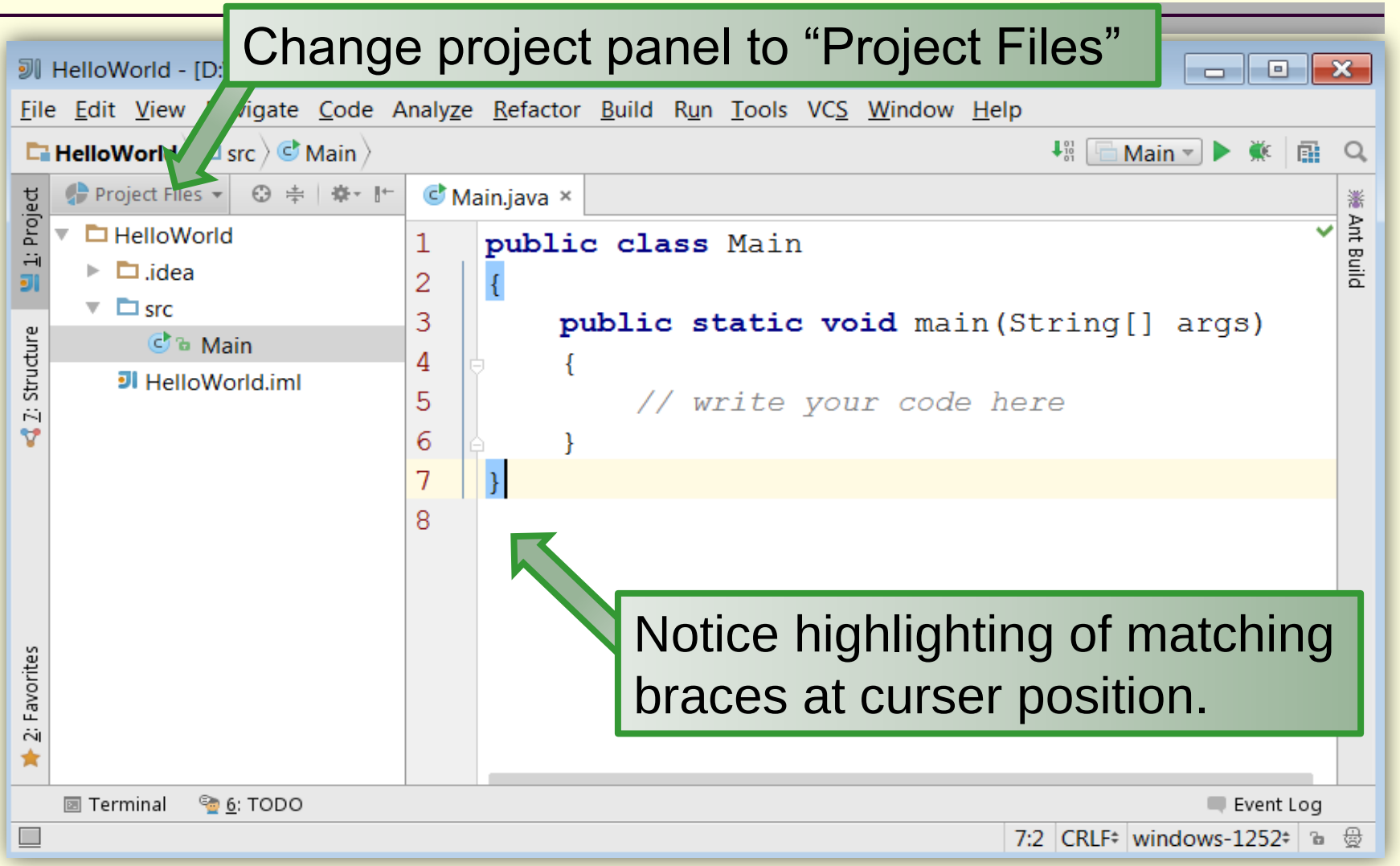
IntelliJ: Create New Project (3 of 5)



IntelliJ: Create New Project (4 of 5)



IntelliJ: Create New Project (5 of 5)



Application: Add Inputs (Similar to Listing 1.1)

```
1. import java.util.Scanner;
2.
3.
4. public class Toy
5. {
6.   public static void main(String[] args)
7.   {
8.     System.out.println("Hello World");
9.     System.out.println("I want two numbers!");
10.
11.    Scanner keyboard = new Scanner(System.in);
12.    int n1 = keyboard.nextInt();
13.    int n2 = keyboard.nextInt();
14.
15.    System.out.println(n1+n2);
16.  }
17.}
```

Filename ***MUST*** be **Toy.java**

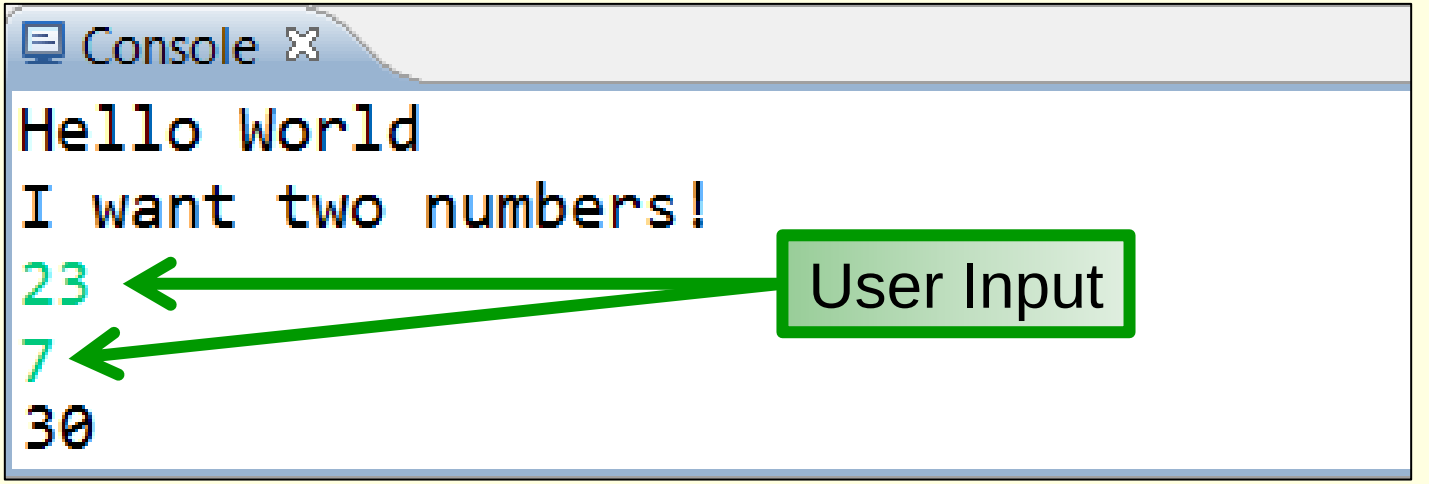
From IntelliJ Project Files panel, Right-click filename and select:

Refactor → Rename

Compile, Run, and the Console

```
1. System.out.println("Hello World");
2. System.out.println("I want two numbers!");
3.
4. Scanner keyboard = new Scanner(System.in);
5. int n1 = keyboard.nextInt(); //reads characters until ↵
6. int n2 = keyboard.nextInt();
7.
8. System.out.println(n1+n2);
```

Compile and Run



Keyword: `import`

```
1. import java.util.Scanner;
2.
3.
4.
5.
6.
7.
8. public class Toy_1_1
9. {
10.   public static void main(String[] args)
11.   {
12.     System.out.pri
13.
```

Gets the **`Scanner`** class from the ***package*** (library) **`java.util`**.

All import statements are placed at the top of the source file.

Note: The Java programming language is ***case-sensitive***.

Keyword: `class`

```
1.  import java.util.Scanner;
2.
3.  public class Toy
4.  {
5.     publ
6.     {
7.        Sy
8.        Sy
9.
10.       Scanner keyboard = new Scanner(System.in);
11.       in
12.       in
13.
14.       Sy
15.    }
16. }
```

class header.
In Java, all code is enclosed in a class.
Class names are your choice but.... *should* start with an upper-case letter.

The *class header* must be followed by an *open curly bracket,* `{`, and ended with a matching *close curly bracket*, `}`.

All code within those brackets is part of the class.

Method Signature

```
1. import java.util.Scanner;
2.
3. public class Toy
4. {
5.    public static void main(String[] args)
6.    {
7.       Sy
8.       Sy
9.
10.      Sc
11.      in
12.      in
13.
14.      Sy
15.   }
16.}
```

Line 5 is a ***method signature***.

Every Java application must have a **`main`** method.

By convention, method names start with a lower-case letter.

A method signature must be followed by an opening curly bracket '**`{`**' and a matching closing curly bracket '**`}`**'.

Code within the brackets is part of the method.

Keyword: `public`

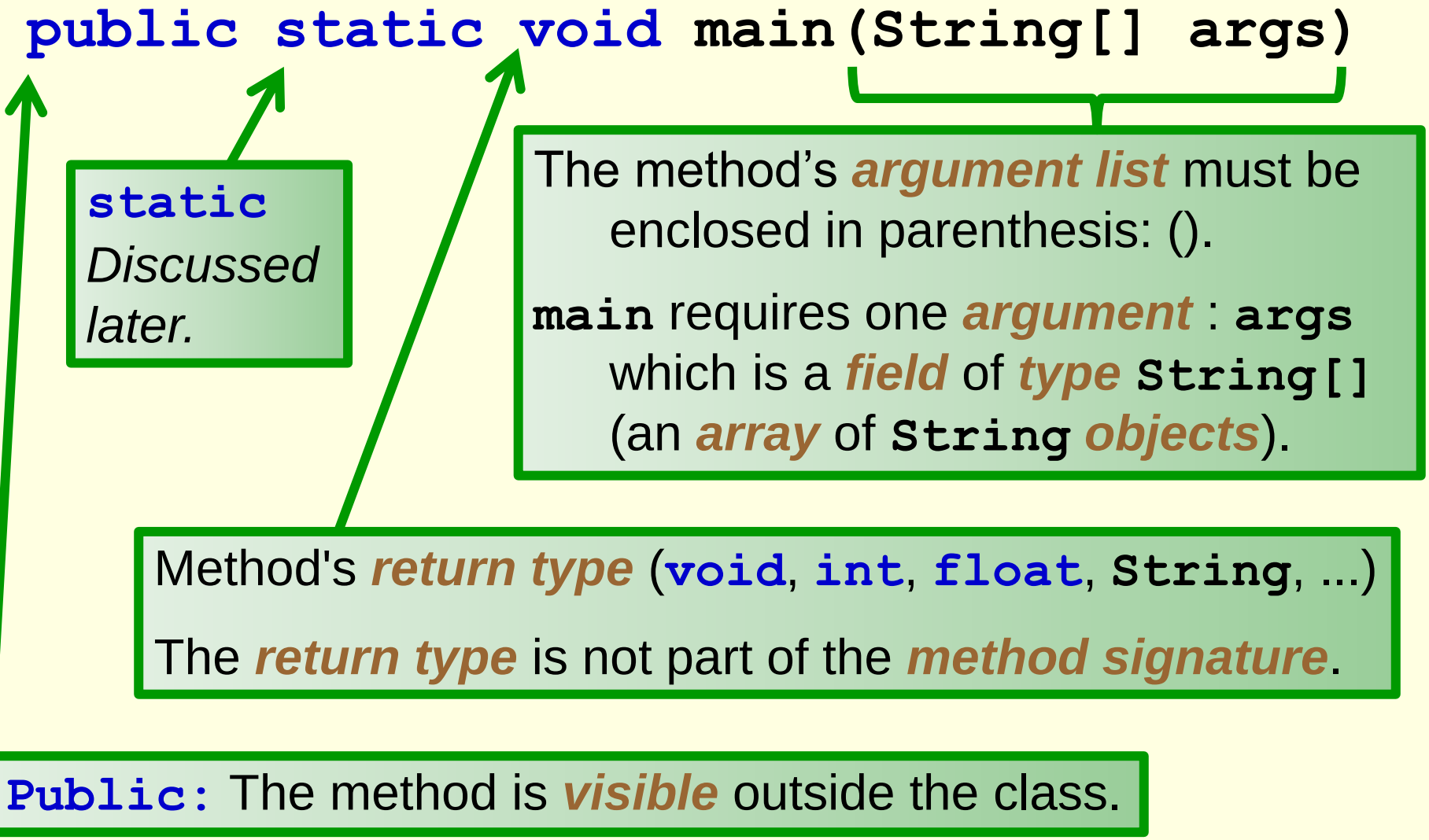
```
1. import java.util.Scanner;
2.
3. public class Toy
4. {
5.    public static void main(String[] args)
6.    {
7.       System.out.println("Hello World");
8.
9.
10.
11.
12.
13.
14.
```

The keyword **`public`** means that the ***class***, ***field*** (variable), or ***method*** is ***visible*** to other classes.

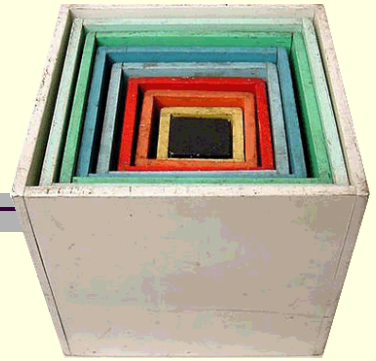
The outermost class in each Java source file must be **`public`**.

The **`main`** method must be **`public`**.

Parts of a Method Signature



Block Structure



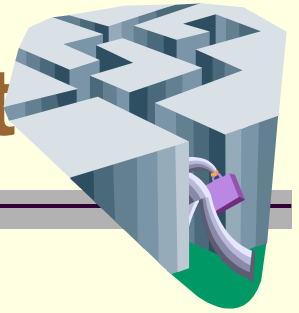
```
1. public class Toy
2. {
3.    // Comment inside Toy.
4.    public static void main(String[] args)
5.    {
6.       // Comment inside main which is inside Toy.
7.       System.out.println("Hello World");
8.    }
9. }
```

The ***nesting*** of curly brackets defines the block structure.

Good programming ***style*** requires block indentation.

In CS-259, we will use exactly two spaces per level.

Program Execution Entry Point



```
1. public class Toy
2. {
3.    public static void main(String[] args)
4.    {
5.
```

When a Java program runs, code ***execution*** starts at **`main`**.

Every Java ***application*** must have **at least one** class with a **`main`** method.

It is ok for an application to have more than one class with **`main`**.

When a user runs a Java application, the user must specify which class contains the **`main`** he or she wants to use.

Method Calls

```
1. System.out.println("Hello World");
2. System.out.println("I want two numbers!");
```

Line 1 ***Calls*** the **`println`** ***method*** of the ***object*** **`System.out`** with the ***argument*** **`"Hello World"`**.

The **`println`** ***method*** takes a **`String`** ***object*** for an ***argument*** (input).

The **`println`** ***method*** displays its argument in the Java Console and then displays a newline character, **`'/n'`**.

Keyword `new`: Instantiating a Class

```
Scanner keyboard = new Scanner(System.in);
```

Declares a ***field*** (with the programmer defined name **`keyboard`**) to be of ***type*** **`Scanner`**.

The ***type*** **`Scanner`** is defined in **`import java.util.Scanner`**

argument list

Creates an ***instance*** of the **`Scanner`** class.

The = symbol ***assigns*** **`keyboard`** to the ***memory address*** of the newly created ***instance*** of **`Scanner`**.

Reading User Input from the Keyboard

```
1. Scanner keyboard = new Scanner(System.in);
2. int n1 = keyboard.nextInt();
```

Line 2 does many things:

1) ***Defines*** **`n1`** to be a ***field*** of ***type*** **`int`**.
2) ***Allocates*** memory for **`n1`**.
3) ***Calls*** the **`nextInt`** ***method*** of the keyboard ***instance*** of **`Scanner`**. The **`nextInt`** ***method***:
   i. ***Reads*** input from the keyboard.
   ii. ***Echoes*** the input to the ***Console***.
   iii. ***Tries to convert*** the user input to an **`int`**.
   iv. ***Returns*** the converted input.
4) ***Assigns*** the returned **`int`** value to **`n1`**.

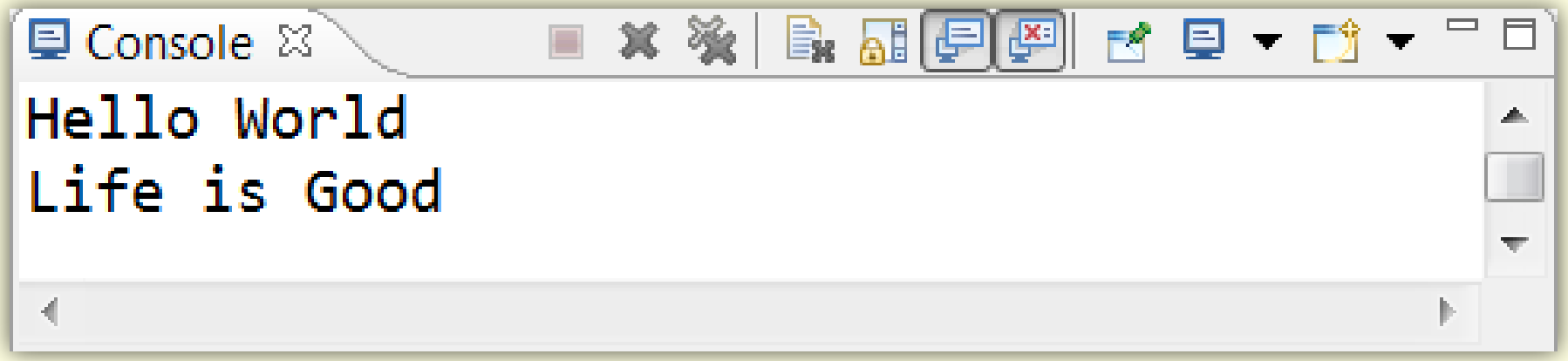
End of Statement ; Start of Block {

```
1.  import java.util.Scanner;
2.
3.  public class Toy
4.  {
5.    public static void main(String[] args)
6.    {
7.      System.out.println("Hello World");
8.      System.out.println("I want 2 numbers!");
9.
10.     Scanner keyboard = new Scanner(System.in);
11.     int n1 = keyboard.nextInt();
12.     int n2 = keyboard.nextInt();
13.
14.     System.out.println(n1+n2);
15.   }
16. }
```

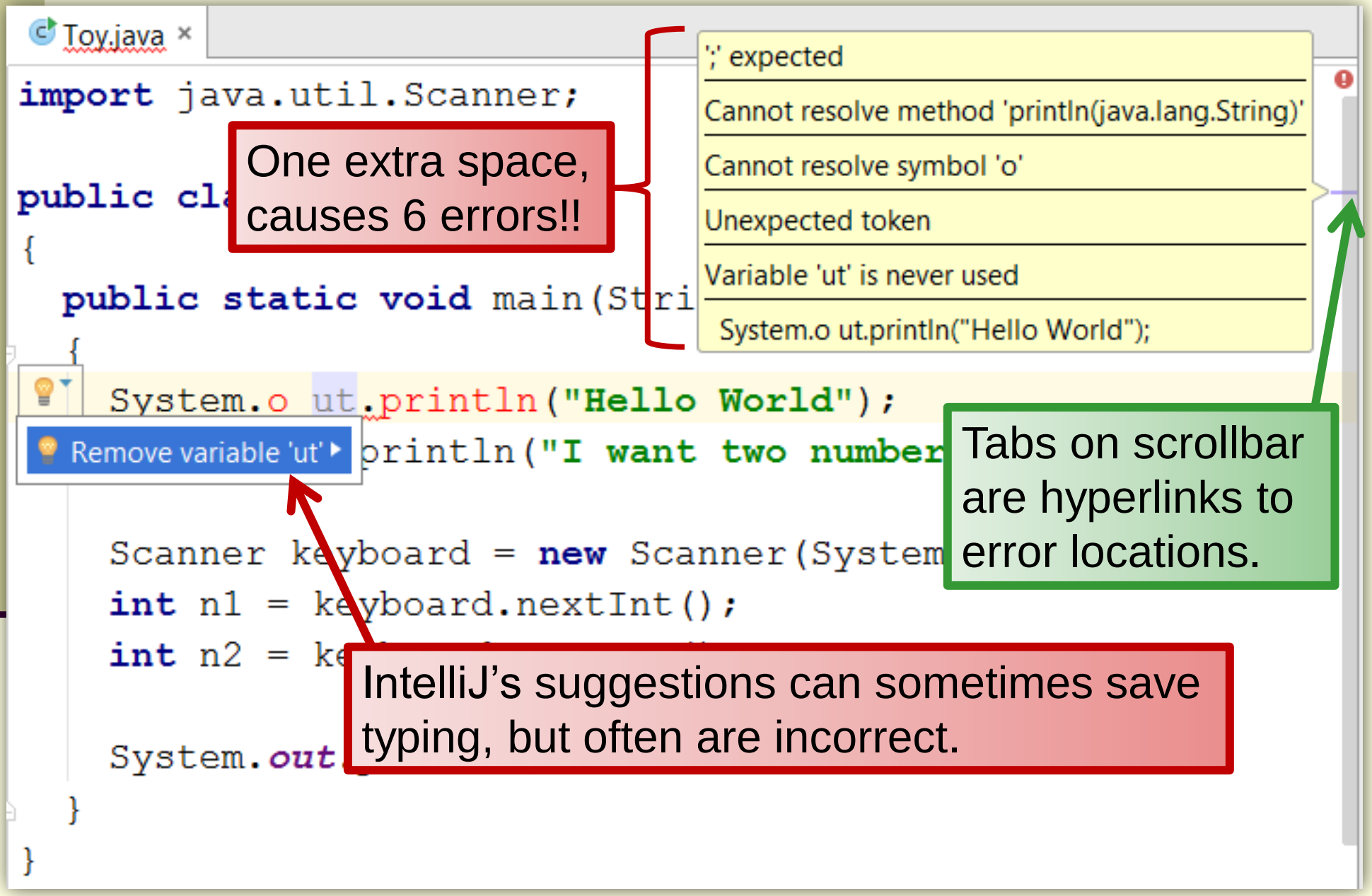
Whitespace: Ignored Between Identifiers

```
1.  System.out.println("Hello World");
2.
3.  System.     out.
4.      println    (
5.          "Life is Good"
6.      );
```

Single ***statement*** split across multiple lines.



Error: Whitespace Within Identifiers



Find and Fix the Syntax Error: #1

```
1. import java.util.Scanner;
2.
3. public class Toy
4. {
5.   public static void main(String[] args)
6.   {
7.     System.out.println("Input 2 numbers");
8.
9.     Scanner in = new Scanner(System.in);
10.    int n1 = keyboard.nextInt();
11.    int n2 = keyboard.nextInt();
12.
13.    System.out.println(n1+n2);
14.  }
15.}
```

Find and Fix the Syntax Error: #2

```
1. import java.util.Scanner;
2.
3. public class Toy
4. {
5.   public static void main(String[] args)
6.
7.     System.out.println("Input 2 numbers");
8.
9.     Scanner bob = new Scanner(System.in);
10.    int n1 = bob.nextInt();
11.    int n2 = bob.nextInt();
12.
13.    System.out.println(n1+n2);
14.  }
15.}
```

Find and Fix the Syntax Error: #3

```
1. import java.util.Scanner;
2.
3. public class Toy
4. {
5.   public static void main(String[] args);
6.   {
7.     System.out.println("Input 2 numbers");
8.
9.     Scanner in = new Scanner(System.in);
10.    int n1 = in.nextInt();
11.    int n2 = in.nextInt();
12.
13.    System.out.println(n1+n2);
14.  }
15.}
```

Setting IntelliJ to **Project Files** View