CS5041 – Tutorial 1 – Processing

Part VI: Using IntelliJ with Processing

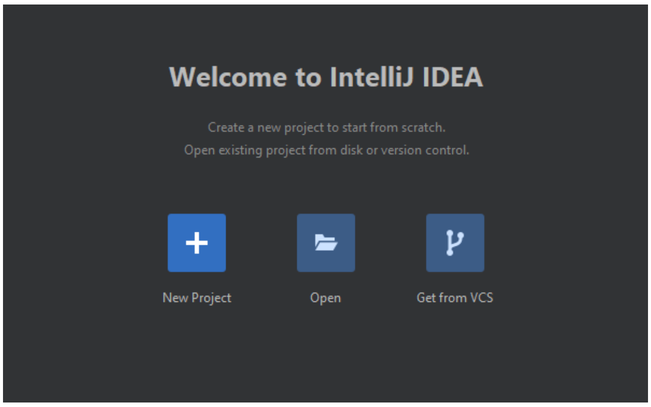
For larger projects, it can be more convenient to use a different IDE such as Eclipse or IntelliJ, rather than the native processing IDE. This is possible, since processing is just a Java library that can be imported into other Java projects. In the following, we explain how to use the processing library to create your graphics while coding in IntelliJ. You can get this application from here: https://www.jetbrains.com/idea/

Some external references:

- Creating, Running and Packaging Your First Java Application
- Adding external .jar files to IntelliJ projects.

Creating a New IntelliJ Project

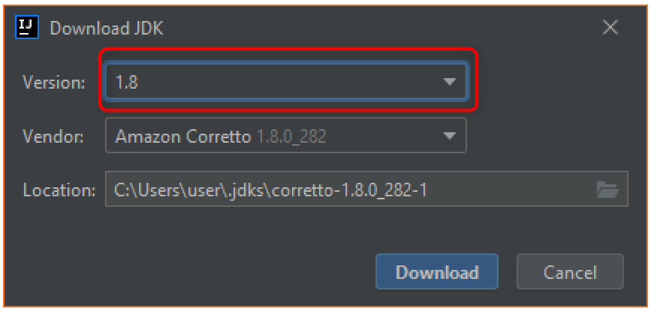
- Open IntelliJ IDEA
- Click Create **New Project** on the Welcome screen, or select **File | New | Project**. The **New Project** wizard opens



Welcome to IntelliJ IDEA

Create a new project to start from scratch.
Open existing project from disk or version control.

- On the left-hand pane, select **JAVA**
- On the right-hand pane, you will need a Java Development Kit, if you already have this – skip on, if not:
  - select 'Download JDK'
  - select version 1.8
  - leave the vendor as it is
  - select a location for the download
  - press 'download'
  - It should download and then select your JDK. Move to next step

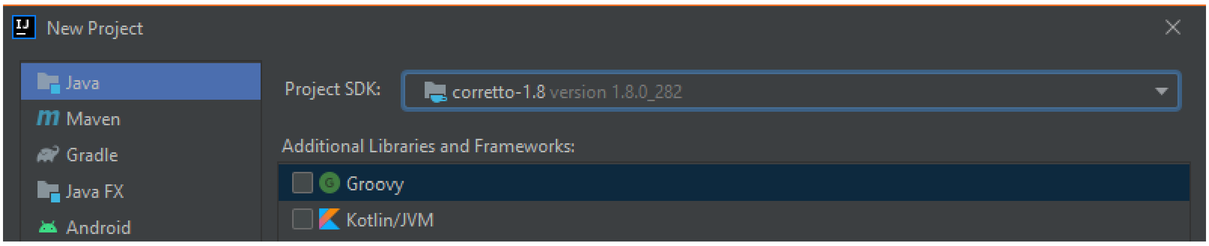


Download JDK

Version: 1.8

Vendor: Amazon Corretto 1.8.0_282

Location: C:\Users\user\.jdks\corretto-1.8.0_282-1



New Project

Project SDK: corretto-1.8 version 1.8.0_282

Additional Libraries and Frameworks:

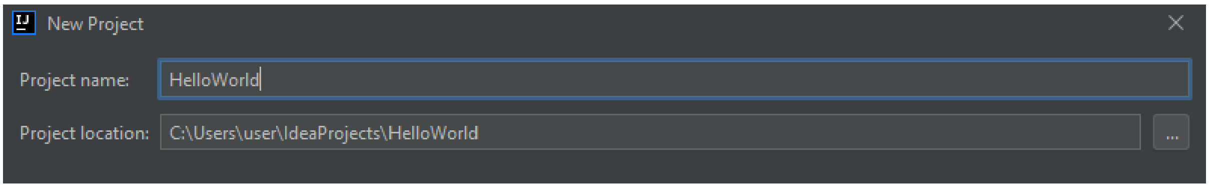
□ Groovy

□ Kotlin/JVM

- We want to do everything from scratch, so we don’t select any options for Additional libraries.
- Click **Next** at the bottom of the screen to move to ‘template’ settings

- We want to write from scratch so **don’t** select to create a project from template – instead, just hit next

- Give your project a name, for example, “HelloWorld”

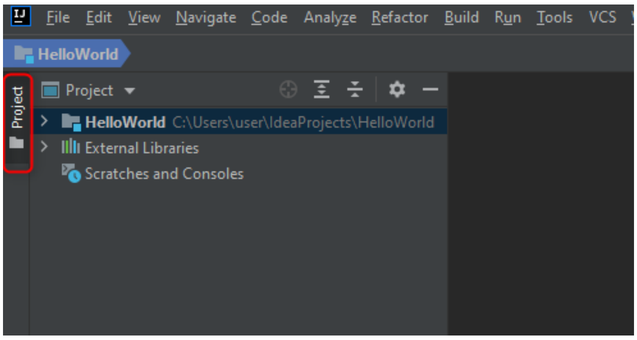


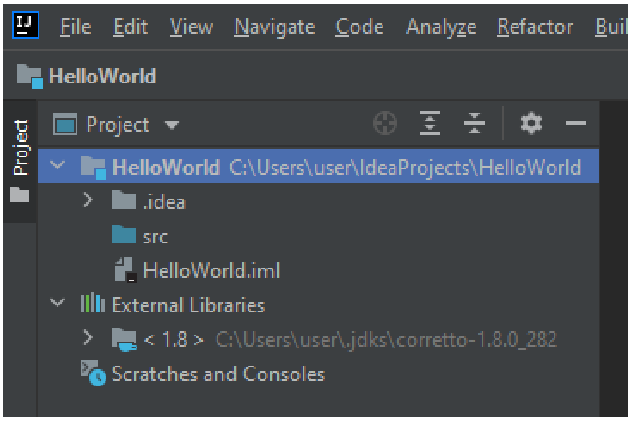
New Project

Project name: HelloWorld

Project location: C:\Users\user\IdeaProjects\HelloWorld

- Click **Finish**.
- IntelliJ will create your empty Java project
- In the next window, you may or may not see your project structure, if not, press ‘project’ to reveal the tree structure of your project. In your project structure view (left) you will see two top-level nodes:
- **HelloWorld**. This node represents your Java module. The .idea folder and the file HelloWorld.iml are used to store configuration data for your project and module respectively. The folder src is for your source code.
- **External Libraries**. This is a category that represents all the "external" resources necessary for your development work. Currently in this category are the .jar files that make up your JDK.

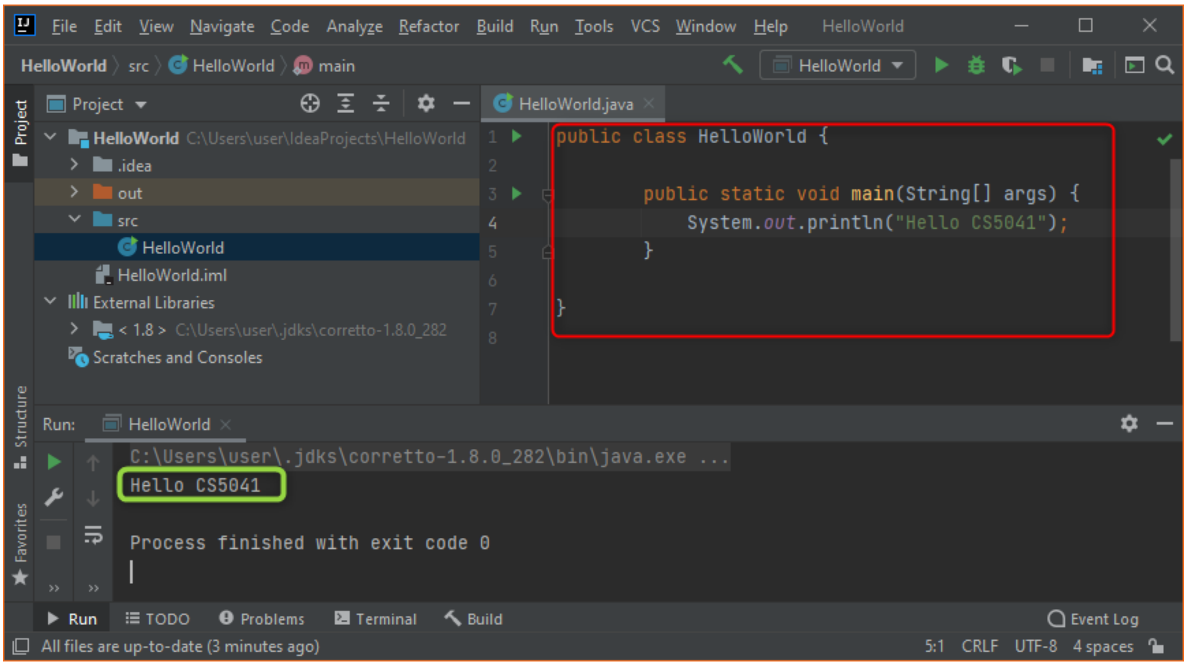




- Right click the **src** folder and select **New | Java Class**
- Name the class exactly as your project: **HelloWorld**. Press ‘Enter’ to create.
- To get some code started, copy and paste the following code into your editor

```
public class HelloWorld {

    public static void main(String[] args) {

            System.out.println(“Hello CS5041”);

    }

}
```

- Run this program by selecting **Run | Run** from the main menu
- When prompted, select (“1. HelloWorld”), after some time, the programme will run and output some text to the console, which should appear. Your window should look like this – note your code is highlighted in red and the output in the console is highlighted in green



```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello CS5041");
    }

}
```

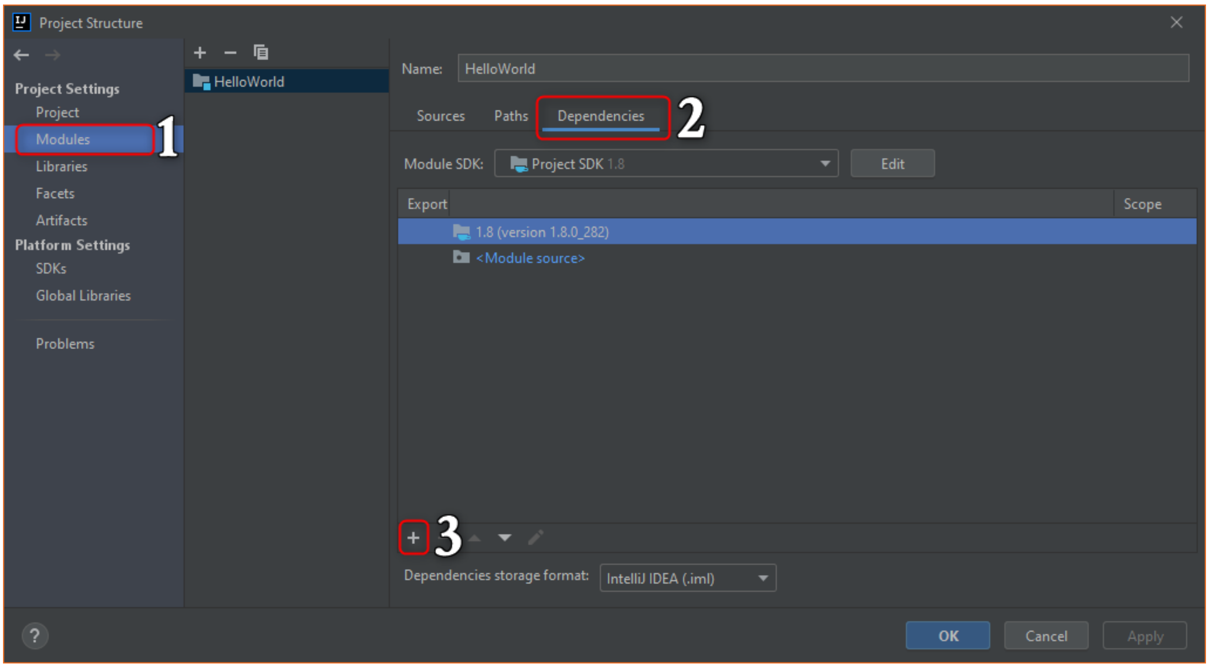
```
C:\Users\user\.jdks\corretto-1.8.0_282\bin\java.exe ...
Hello CS5041

Process finished with exit code 0
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help HelloWorld
HelloWorld › src › HelloWorld › main
Project
HelloWorld C:\Users\user\IdeaProjects\HelloWorld
.idea
out
src
HelloWorld
HelloWorld.iml
External Libraries
< 1.8 > C:\Users\user\.jdks\corretto-1.8.0_282
Scratches and Consoles
HelloWorld.java
Run: HelloWorld
Run TODO Problems Terminal Build Event Log
All files are up-to-date (3 minutes ago)
5:1 CRLF UTF-8 4 spaces

Adding the Processing Library

- Click **File | Project Structure** from the main menu.
- On the left-hand side select **Modules** and then the **Dependencies** tab to the right

- Now select the ‘+’ sign – this used to be on the right, but may now be down the bottom. On my latest version the window for these last three steps, looks like this:



Name: HelloWorld

Module SDK: Project SDK 1.8

1.8 (version 1.8.0_282)

<Module source>

Dependencies storage format: IntelliJ IDEA (.iml)

- Click the **+ button** and select **1 JARs or directories**
- Select folder where you have placed **Processing**. Find the core.jar file in **processing-3.x|core|library|core.jar**. Select the jar file and click **OK**
- Click **Apply** and then **OK** again to close the dependency dialog
- You should see the **processing core.jar** now under the **External Library node** to the left.

Creating a Processing Project in IntelliJ

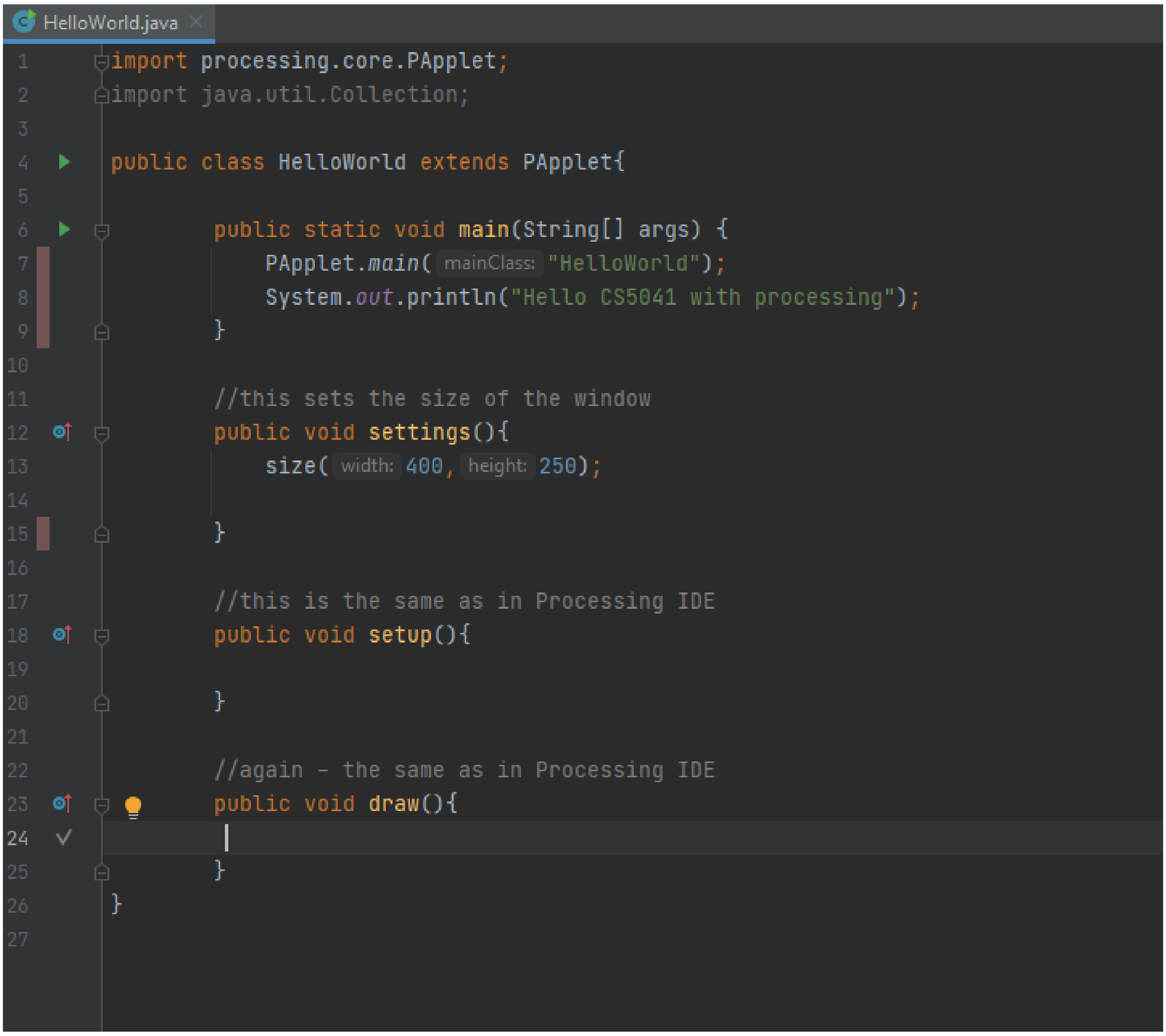
- You can now import the processing library into your Java project using the following code:

```
1. import processing.core.PApplet;
2.
3. public class HelloWorld extends PApplet{
4.
5.     public static void main(String[] args) {
6.         PApplet.main("HelloWorld");
7.
8.     }
9.
10.  }
```

The Processing window is a Java programme called a ‘PApplet’. This is technically a Java class and so it has its own main() function. We are making our program a PApplet for this to work correctly and so we extend the PApplet class on line 3 above. This means that we inherit all the variables and functions from that class and then build our program on top of this. If you run the code above, you will see the normal Processing window pop up as you are used to (although it will probably be quite small).

Note that if you have got anything wrong, you will see errors pop up in the console and you can deal with them there.

To make this into something you can work with you can now include draw(), setup() and settings() which sets the size of the window. In blank form your code window will now look like this:



```java
import processing.core.PApplet;
import java.util.Collection;

public class HelloWorld extends PApplet{

        public static void main(String[] args) {
            PApplet.main( mainClass: "HelloWorld");
            System.out.println("Hello CS5041 with processing");
        }

        //this sets the size of the window
        public void settings(){
            size( width: 400, height: 250);

        }

        //this is the same as in Processing IDE
        public void setup(){

        }

        //again - the same as in Processing IDE
        public void draw(){

        }
}

```