Class 0 – Preparation tasks

Part A – Install IntelliJ IDEA for your personal computer

This module will be using the IntelliJ IDEA development environment to program in Java. IntelliJ IDEA is one of the most popular Java development environments used in the industry (the other is Eclipse IDE).

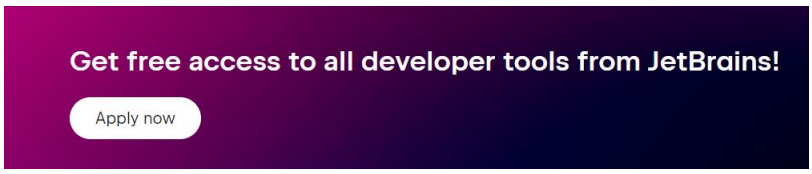
IntelliJ IDEA offers free educational licenses for all university students. To install IntelliJ IDEA in your own personal computer, you will need to download a copy of the software and request an educational license to use the software for free.

Step 1: Go to https://www.jetbrains.com/idea/

Step 2: Download the IntelliJ IDEA Ultimate version

Step 3: Apply for a free educational license at https://www.jetbrains.com/community/education/#students

Click on "Apply now"



The authorisation should be quick (student status is verified through your academic email address). Through this process you will create an account with a licence ID that can be accessed online.

Step 4: Install IntelliJ IDEA on your computer. After installation, when prompted you can authorise your copy using the credentials from your educational account.

You can now use IntelliJ IDEA on your personal computer.

Part B – Exploring IntelliJ IDEA

Getting started

We are going to write a new application straight away.

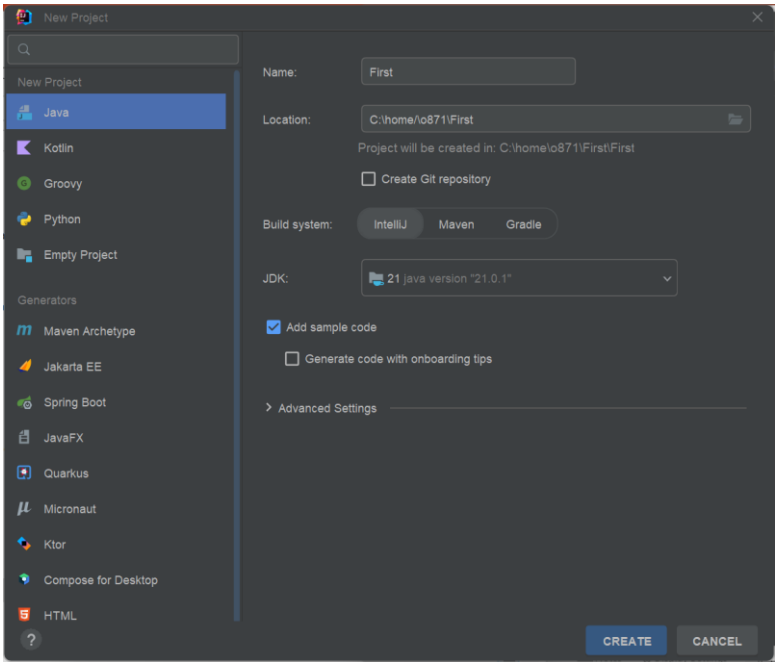
1. From the main screen chose New Project.

   From the popup select Java from the list on the left.

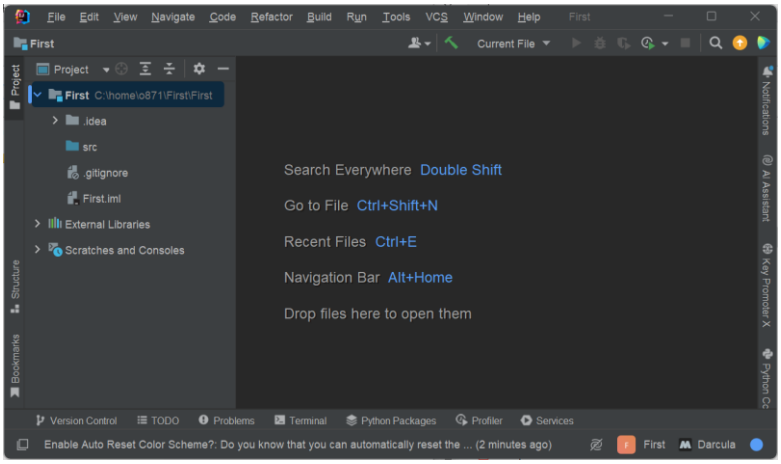
   On the right panel:

   - Specify the name of your project to be "First".
   - It doesn't matter where the Project Location is, but it is a good idea to make sure that you have specified a folder where you want to keep your project in general, e.g. C:\home\co871\First.
   - Select "Intellij" as the build system.

- Uncheck “Add sample code”, and then click CREATE.



2. The main development area of your project should now be open. You should see a view like this:



The panel on the left is the project navigator. Expand the First module, you will see the folder src. This is where your java source code should be added.

3. Add a package to your project: Right click on the src folder and select New -> Package. Name your package first (package names should be in lowercase).

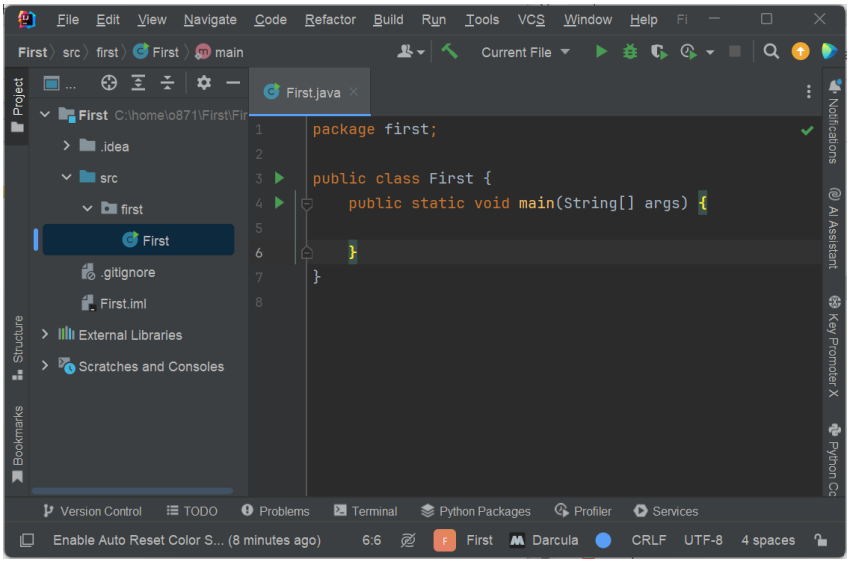
   A package in Java is a container for one or more classes. By convention we use the reverse domain name, e.g. kent.co871.demo.

4. Add a new class to your package. Right click on the first folder that has been created inside the src folder. Select New -> Java Class. Name your new class First. You should now have the new class open in the Editor window.

5. In order to run some code, you will need a static main method. Put your cursor inside the body of class First. Type the word main and wait for a couple of seconds. IntelliJ will prompt

you to generate the method main() for you, with the appropriate syntax. Accept that recommendation.

6. Right now your class does not have any code yet, but you can try running it for the first time. You can right-click on the class First on the project navigator and select Run 'First.main()'. Or click on the run button at the toolbar top-right corner of IDE.



7. When you select to run the program, a new window will appear at the bottom with the output of your application. Right now it will only show that the program finished with exit code 0.

8. Add a static int member variable called x to the class:

```
private static int x;
```

9. Add a statement in the main() method:

```
Random gen = new Random();
```

You will notice that the class name Random is coloured red. If you let your cursor rest at the end of the line for a second, IntelliJ will pop up with a message. What is that message? Hovering the mouse over the red word will also popup with a solution. Fix the problem by importing the java.util.Random class.

10. Now we want a random integer. On the next line, type:

```
x = gen.nextInt(100);
```

11. x is a bad name. Let's use a more meaningful one, say randomInt. Right-click on one of the instances of x and select Refactor->Rename. Rename x to randomInt.

12. On the next line, type sout. and wait for the pop-up menu and then choose the option to print System.out. Include the randomInt variable in the println() method.

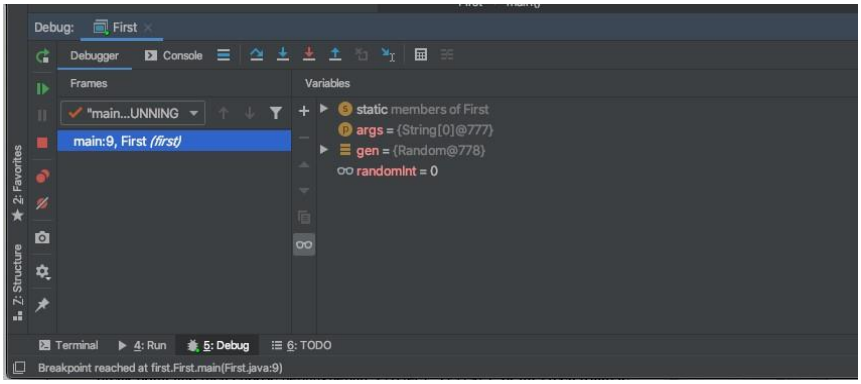
13. Run the program. The output window at the bottom of the screen should show the output of the run.

14. Write some code, it doesn't matter what. Introduce some syntax errors: what do you notice? Correct the errors and save your program.

15. Try a debug run. First add a break point: Click on the margin space on the left of the editor area near the line where you want to introduce a break point. A red dot should appear:

Then do a debug run: either click the toolbar button that looks like a green bug, or select Run->Debug 'First' from the menu.

16. At the bottom of the screen you should see the Debug window. It includes the list of variables on the right, and the call stack on the left. Right over the two panels there is a tool bar with buttons that control the execution of your program. Hover the mouse over the buttons to see what each button can do.



17. Use the buttons to step through the execution of your application.

Opening an existing project

If you have downloaded an IntelliJ project (i.e. from the course exercises).

- First store the downloaded project folder where you would like to keep your working projects.
- Then either choose Open from the File menu, or if you are on the Welcome screen select Open.

Note that when you open an IntelliJ project from another source, it is possible that the project is configured with a JDK version which is different to the one you have on your computer. IntelliJ will prompt you with a message at the top of the editor area, asking you to select a different JDK version. This will open the project configuration window, and you can select the JDK version you have on your computer. If you have an up-to-date JDK, this should be enough to work with the downloaded project. We use Java version 21 so you should download JDK 21 from https://jdk.java.net/archive/ and install it on your own computer.