

SWEN20003

Object Oriented Software Development

Workshop 1

Eleanor McMurtry

Semester 2, 2020

Introduction

Welcome to Object Oriented Software Development! These workshops contain practical problems that will give you hands-on experience designing and implementing programs in Java. Participation **is assessed**. Each workshop is worth 1 mark: to be eligible for this mark you must

1. attend the Zoom session for your enrolled workshop
2. work with your assigned group
3. make a reasonable attempt at the question assigned to you

We highly encourage you to also attempt the other problems for each workshop. By consulting with your demonstrator and getting feedback on your solutions, you will be in a much stronger position for the subject's assessment.

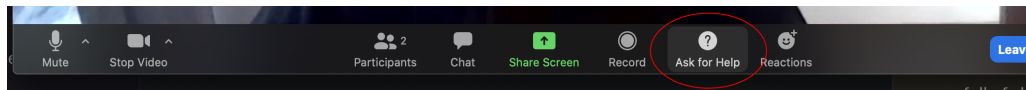
If you cannot attend

If for some reason you cannot attend your enrolled workshop in a particular week, please contact Ni Ding at ni.ding@unimelb.edu.au with your full name, enrolled workshop, and the reason you cannot attend. She will assign you to a different workshop for the week.

Zoom etiquette and primer

For this subject, we request that students **turn on video cameras**. Software engineering is a collaborative discipline and the workshops should be approached as such! If you are uncomfortable doing so, that's OK, but we recommend students use their cameras as many find this improves their learning. We recommend that you **use your real name** as your Zoom username; if you do not, your demonstrator will ask you for your University of Melbourne username so they can mark your participation.

When the workshop starts, your demonstrator will give a brief overview of the problem that will be assessed in the workshop. You will then be placed in breakout rooms with other students — you should work together to make an attempt at the problem. If you are stuck, or would like to show your solution to the demonstrator, press the “Ask for Help” button at the bottom of the screen, and your demonstrator will join the breakout room.



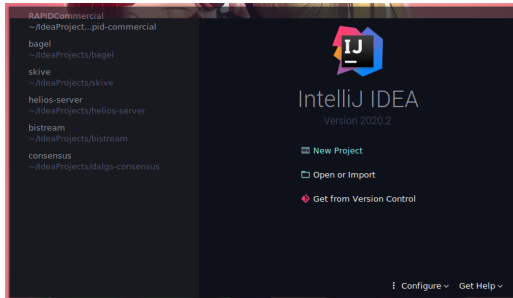
In your breakout rooms, **turn on your microphone** so you can discuss the problem with other students. Once the demonstrator has checked your solution, you will be awarded the participation mark, and can return to the main meeting room. The rest of the workshop time should be spent attempting the other problems for the week — feel free to discuss these problems and collaborate with other students. Your demonstrator will be there to give feedback and suggestions as you work through the problems.

Workshop

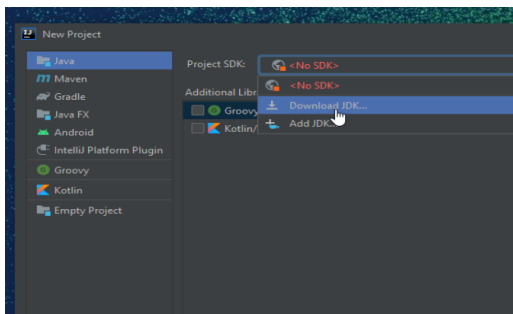
The focus of this workshop is getting set up with the Java development environment we will use, IntelliJ IDEA. To download IntelliJ (for Windows, MacOS, or Linux) head to <https://www.jetbrains.com/idea/download/> and click the Download button **under Community**. Install the software after downloading; the default settings are fine.

To get started with IntelliJ, try compiling a Hello World program following the below steps.

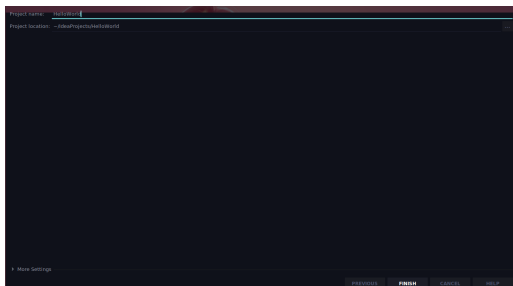
1. When you first open IntelliJ, you'll see the below screen. Click **New Project**.



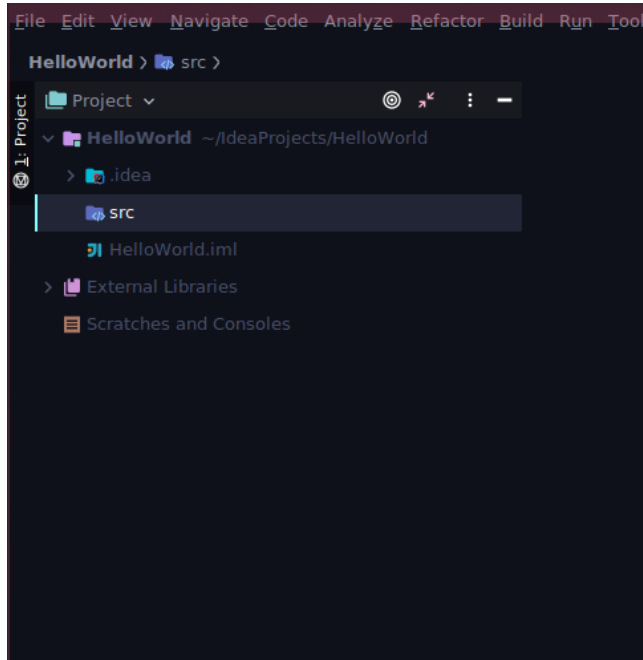
2. Click **<No SDK>** and select **Download JDK...**. The default option is fine; wait for it to download and install, then click **Next**.



3. Give your project a name, e.g. **HelloWorld**.



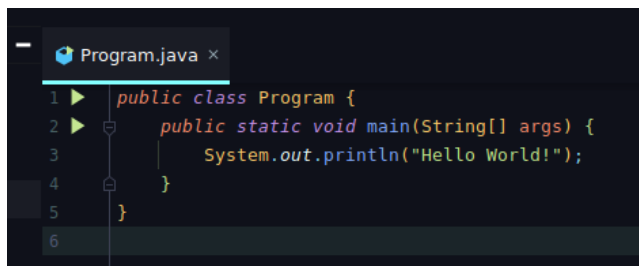
4. Your Java code should go in the `src` folder of your project. Right-click this folder, and click **New** → **Java Class**.



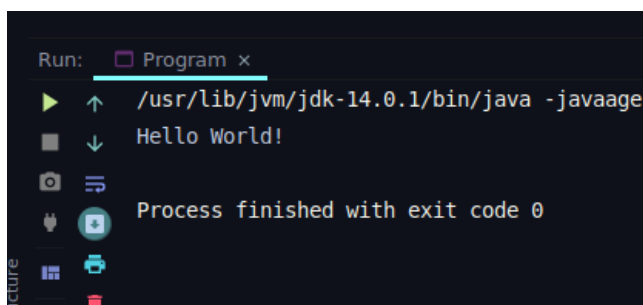
5. Give your class a name, e.g. `Program`. Remember the file name needs to match the name of the class in the code, and class names must start with a capital letter.



6. Add this Hello World code, and press the **Play** button next to the class name (here the button is green). Then click **Run** 'Program.main'. **Note:** the button may take a minute or two to appear the first time you run IntelliJ, since it needs to finish setting up the JDK.



7. Your program's output will appear at the bottom of the screen.



When you're assigned to your groups, make sure everybody in the group has set up IntelliJ before continuing with the assessed problem.

Problems

1. Write a program to print a triangle of some constant size. For example, a triangle of size 5 should look like this:

```
#  
##  
###  
####  
#####
```

2. Write a program to print a diamond of some constant **odd** size. For example, a diamond of size 5 should look like this:

```
  #  
 ###  
#####  
  ###  
  #
```

3. Try compiling and running the example code from lectures using IntelliJ.
4. Continue working through the Grok worksheets.