SENG275 – Lab 0
Due Wednesday Jan 20, 5:00 pm

Welcome to SENG275. Due to the virtual nature of our labs this semester, we'll need to download and install several pieces of software onto our personal computers. This week's task is to install and configure this software, and ensure that everything's ready for the weeks to come. The images in this document were captured from a windows machine; there may be differences from what you see depending on your particular technology.

Quick summary of what you need to do:

1. Install Microsoft Teams, Java 11, IntelliJ IDEA Ultimate and Git on your personal computer if they are not already there.
2. Clone the JPacman program from your course repository at gitlab.csc.uvic.ca
3. Run the program and take a screenshot of it running.
4. Log into the SENG275 Team on Microsoft Teams, and send your screenshot in a message to **TeachingTeam**.

Step-by-Step:

We need to install the following software:

1. Microsoft Teams
2. Java
3. IntelliJ IDEA Ultimate
4. Git
5. JPacman
6. Optional - Gradle

Proceed through the list at your own pace, and please let your TA know if you're encountering anything confusing. If you're not done by the end of the class, please submit your screenshot by next Wednesday at 5 pm Victoria time. If you have any questions after the class, send them to **@TeachingTeam** on Microsoft Teams.

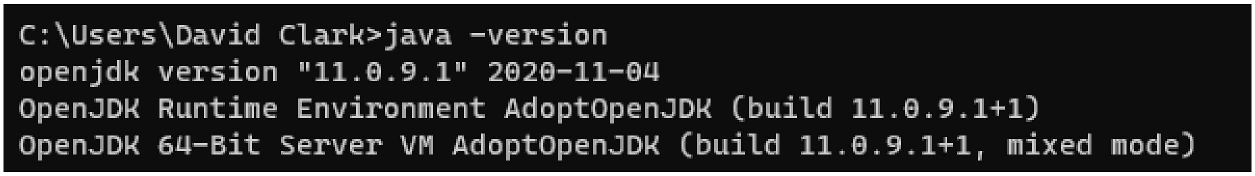
Installing Microsoft Teams

Microsoft Teams, and the rest of the Office 365 suite is available for free to all Uvic faculty, staff and students. Users can only access individual teams by invitation.

Go to https://onlineacademiccommunity.uvic.ca/O365/teams/ for instructions: first you'll have to log in to onlineservices.uvic.ca, and sign up for Office 365. After that, you can log into portal.office.com with your NetLink ID and passphrase and click Teams.

It may take up to 48 hours after you sign up for Teams for all the features to populate.

Installing Java

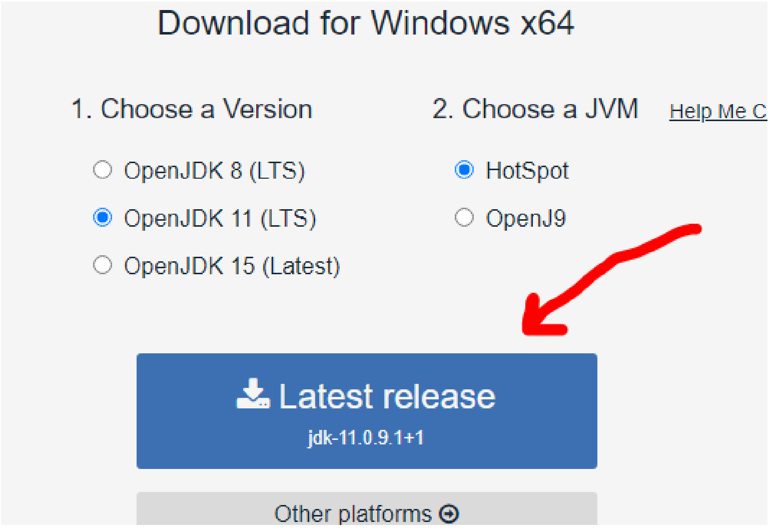


```
C:\Users\David Clark>java -version
openjdk version "11.0.9.1" 2020-11-04
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.9.1+1)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.9.1+1, mixed mode)
```

We have tested the course software with Java 11; other versions may work as well, but if you're having trouble running some of the software, try installing that version.

First, check to see if you already have a Java JDK installed on your machine. Try the following two commands at the terminal or command line: `**java -version**` and `**java --version**`. If either of them gives you a version number, then you have java installed, and can proceed to the next section.

If both of these commands gave you an error message, then you don't have Java installed on your machine. You need to download a Java JDK from one of the open-source implementations - I suggest AdoptOpenJDK. Your choice of version and JVM will depend on your platform; I recommend choosing Java 11 and the HotSpot JVM - some students have reported problems with Java 14, so it's probably best to stick to 11. You may have to reboot your computer to complete the installation - repeat the `**java -version**` and `**java --version**` terminal commands to confirm that your installation went ok.



Download for Windows x64

1. Choose a Version

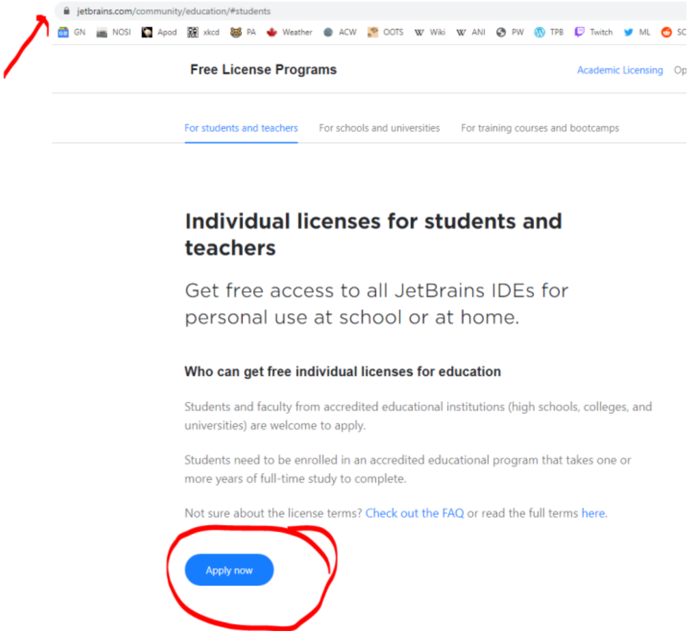
○ OpenJDK 8 (LTS)
● OpenJDK 11 (LTS)
○ OpenJDK 15 (Latest)

2. Choose a JVM

● HotSpot
○ OpenJ9

Installing IntelliJ

IntelliJ is a commercial development environment that is provided in two versions: the Ultimate edition, and the Community edition. We'll eventually be using some of the code coverage tools and automated testing capabilities that are only included in the Ultimate edition, so it's best to use that. Fortunately, every student at UVic has access to the Ultimate edition through NetBrains' Free Educational License program.



jetbrains.com/community/education/#students

Individual licenses for students and teachers

Get free access to all JetBrains IDEs for personal use at school or at home.

Who can get free individual licenses for education

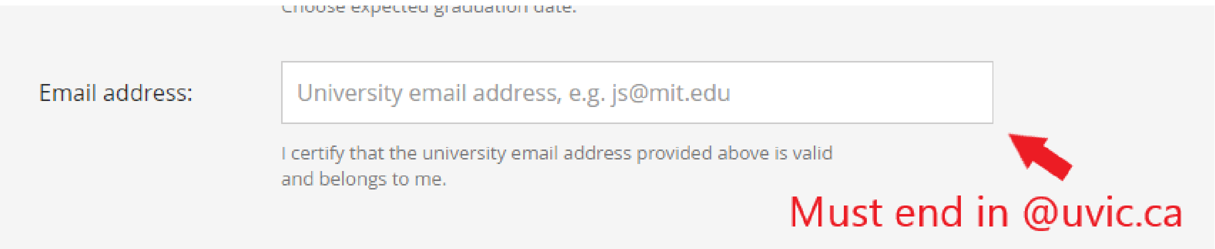
Students and faculty from accredited educational institutions (high schools, colleges, and universities) are welcome to apply.

Students need to be enrolled in an accredited educational program that takes one or more years of full-time study to complete.

Not sure about the license terms? Check out the FAQ or read the full terms here.

To install IntelliJ:

1. Go to the JetBrains website at jetbrains.com/community/education/#students
2. Click on the Apply Now button.
3. Fill out the JetBrains Products for Learning form. Where it asks to enter your email address, make sure you provide your UVic email address! (This will be your netlink login name, followed by @uvic.ca). Renewals and other information from JetBrains will come to your UVic email account, so make sure to check there for further information.



Choose expected graduation date.

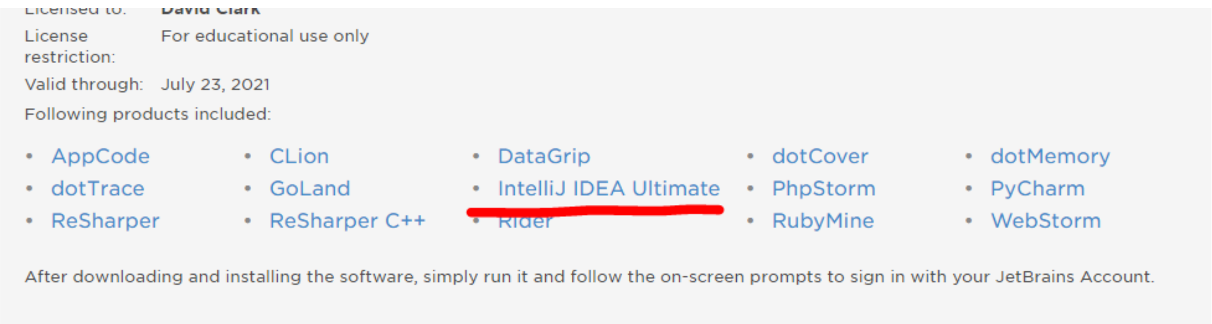
Email address: University email address, e.g. js@mit.edu

I certify that the university email address provided above is valid and belongs to me.

Must end in @uvic.ca

4. The confirmation email should arrive shortly - click on the link in the email to confirm your submission.
5. You'll have to scroll down to the bottom and approve the license agreement.
6. Now you can create a JetBrains account -

   Do NOT use your UVic name and password!
7. Once you've signed into the JetBrains website, you'll see a list of software that you're able to download and install for free. We want the IntelliJ IDEA Ultimate edition - it's available for Windows, Mac and Linux machines, so make sure you choose the appropriate version.

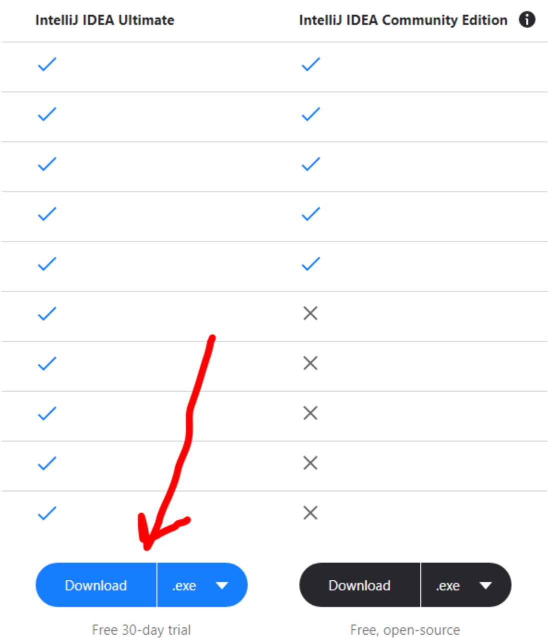


Licensed to:	David Clark
License restriction:	For educational use only
Valid through:	July 23, 2021

Following products included:

- AppCode
- CLion
- DataGrip
- dotCover
- dotMemory
- dotTrace
- GoLand
- IntelliJ IDEA Ultimate
- PhpStorm
- PyCharm
- ReSharper
- ReSharper C++
- Rider
- RubyMine
- WebStorm

After downloading and installing the software, simply run it and follow the on-screen prompts to sign in with your JetBrains Account.

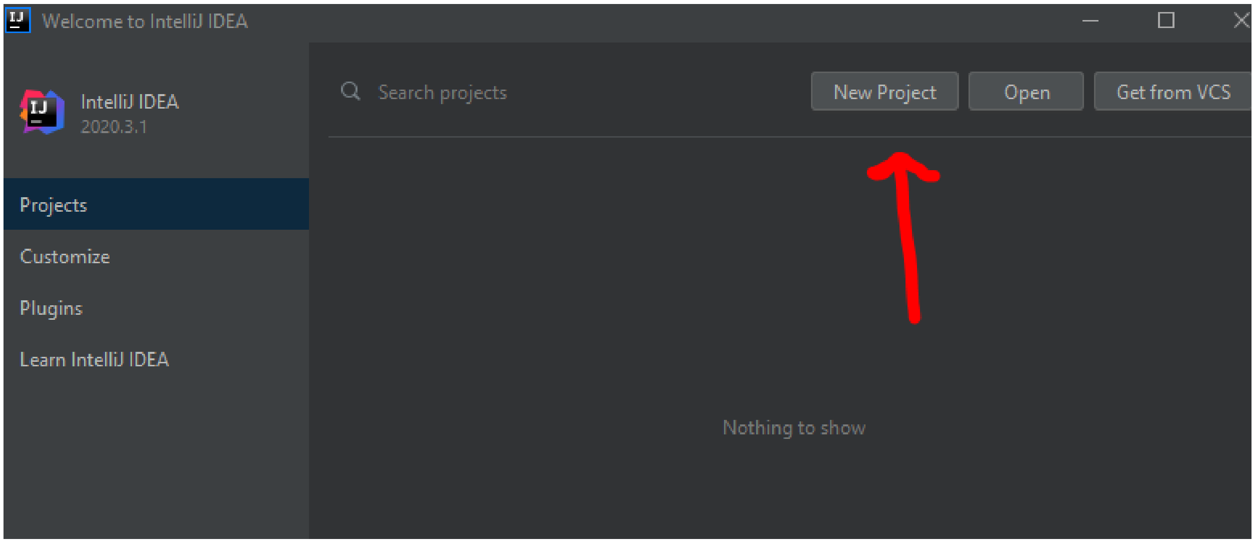


IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition
✓	✓
✓	✓
✓	✓
✓	✓
✓	✓
✓	×
✓	×
✓	×
✓	×
✓	×
Download .exe	Download .exe
Free 30-day trial	Free, open-source

8. Install the software. Choose where you want to install it on your computer, and read the installation options - its safe to leave the boxes unchecked.

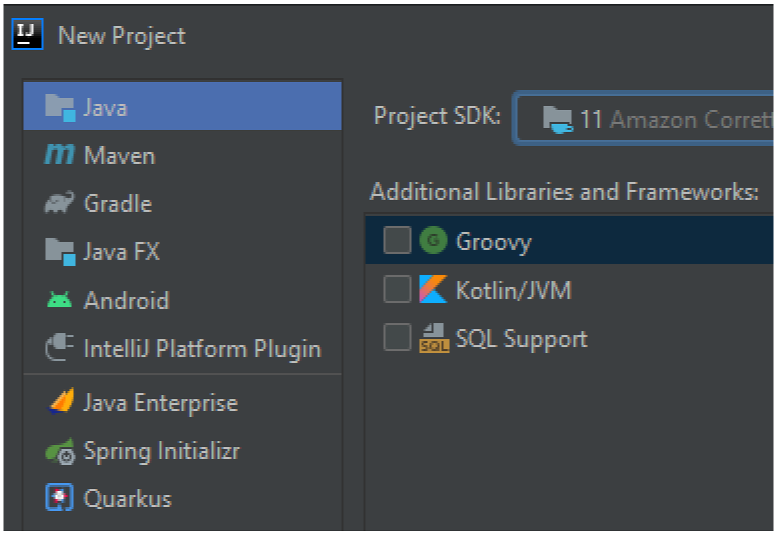
Let's make sure everything worked:

1. Run IntelliJ. You should see the Welcome to IntelliJ IDEA window open up.
2. Click New Project.



3. You'll see the New Project dialog - note that if you don't have Java installed on your machine, IntelliJ will use its built-in Java SDK, which is currently Java 11. The version of

Java that's included with IntelliJ is limited, and JetBrains suggests we install a standalone version, which we did above. Click Next to create a basic Java project.



Project SDK: 11 Amazon Corrett

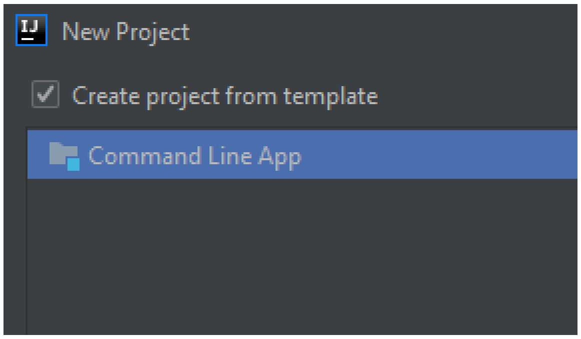
Additional Libraries and Frameworks:

□ Groovy

□ Kotlin/JVM

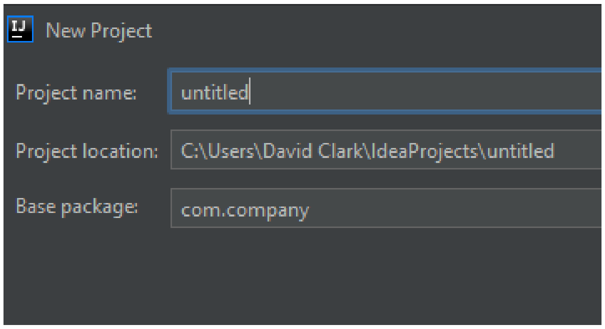
□ SQL Support

4. Enable Create Project from Template, and choose Command Line App. Click Next, and then Finish.



☑ Create project from template

Command Line App

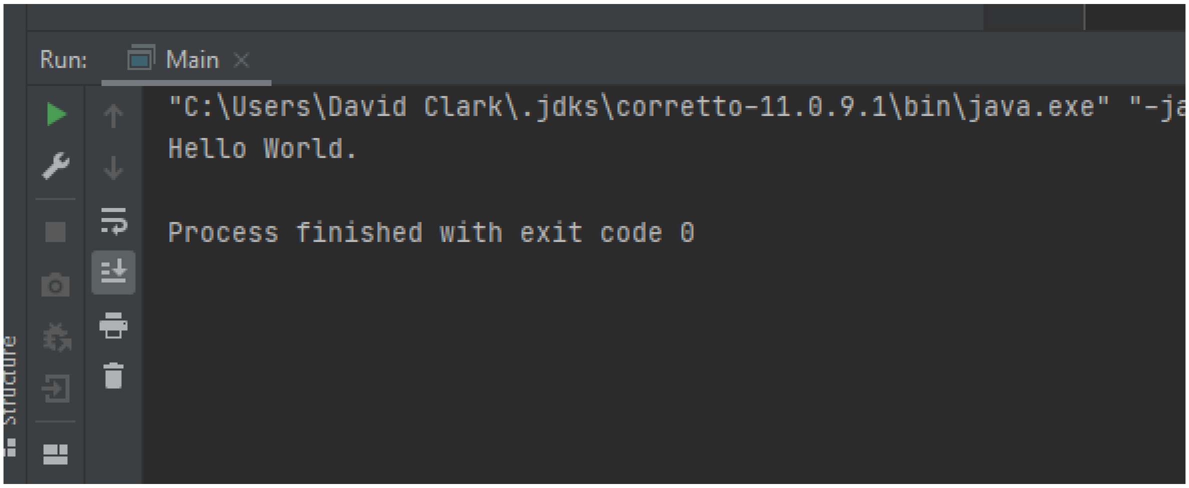


Project name: untitled

Project location: C:\Users\David Clark\IdeaProjects\untitled

Base package: com.company

5. IntelliJ will show you **`Main.java`**, and will start downloading some things in the background. Go ahead and add the following to the main function: **`System.out.println("Hello world.");`**
6. Run the program by clicking the green play icon in the upper right corner, or by choosing `Run -> Run 'Main'` from the menu, or by typing Shift-F10. Confirm that `Hello World.` is printed in the output window at the bottom before continuing.

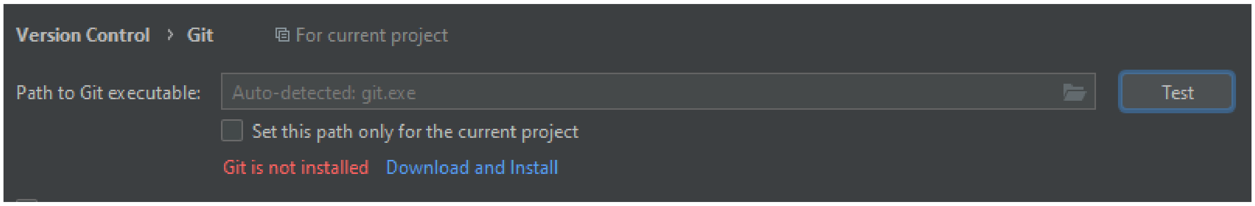


```
"C:\Users\David Clark\.jdks\corretto-11.0.9.1\bin\java.exe" "-ja
Hello World.

Process finished with exit code 0
```

Git

You may already have Git installed on your computer. Go to `**File -> Settings -> Version Control -> Git**`, and click the Test button on the right. If something like '**Git is not installed**' appears, click the '**Download and install**' text beside it, and install a version of git for your machine.

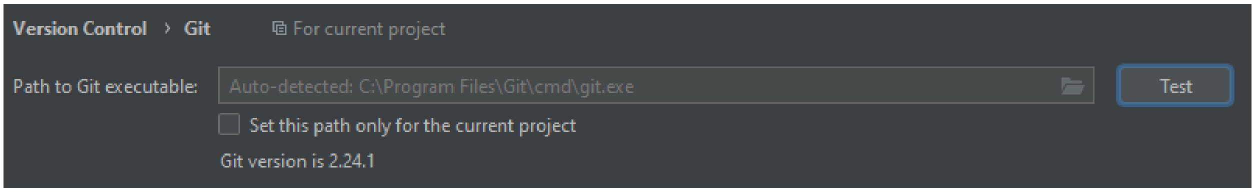


Path to Git executable: Auto-detected: git.exe

□ Set this path only for the current project

Git is not installed   Download and Install

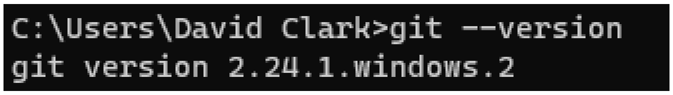
Test that Git works properly : type `**git --version**` at the command line or in a terminal. You should see something like `**git version is 2.24.1**` - the exact version isn't critical, as long as you're reasonably up-to-date.



Path to Git executable: Auto-detected: C:\Program Files\Git\cmd\git.exe

□ Set this path only for the current project

Git version is 2.24.1



```
C:\Users\David Clark>git --version
git version 2.24.1.windows.2
```

Your course repository is located at gitlab.csc.uvic.ca. You can find it by selecting '**Projects -> Your Projects**' from the upper left corner of the screen. Clone this repository to your computer, using the HTTPS link.

This course won't be covering the use of Git, which was one of the tools covered in SENG265. If you're having trouble remembering how to clone a repository, don't worry; some of the commands are a little obscure. Check the hands-on tutorial that GitLab provides. A more advanced understanding of Git will be quite helpful during the class; refer to the online book on the topic if you wish. You will most likely need to use the '**git config**' command to set up your git identity, and '**git clone**' to clone your repository.

IntelliJ has robust gui support for most of the basic Git functions. That said, you are encouraged to become familiar with the command line functionality provided by git. You should also feel free to try some of the visual gui git utilities that are out there - they may help you better visualize changes you make to your repository. Your teaching instructors will be using the command line exclusively, and may be unable to support you if you rely on an unfamiliar tool.

A few notes on Git best practices for this course:

- Make sure you commit frequently - at least once per exercise.
- Write clear commit messages - when writing software with others, these messages help communicate your intent to your co-workers. If in doubt, check out a guide on how to write clear commit messages.
- Push frequently to the GitLab remote, so that any automated builds and tests are triggered.
- Research and use branches. Git Branches are a technique for separating and organizing changes, and are necessary once large teams are working simultaneously on a program.
- Use merge requests as supported by GitLab, and avoid committing to master directly.

JPacman

This semester, we'll be writing our tests against a java program called JPacman, an incomplete implementation of the Pac-Man arcade game.

1. Clone JPacman using the git repository URL from gitlab.csc.uvic.ca.
2. Open JPacman in IntelliJ, using either the Open button on the Welcome to IntelliJ screen, or `**File -> New -> Project From Existing Sources**`. Note that the `**.gradle**` folder within the project is highlighted; IntelliJ recognizes JPacman as a Gradle project.
3. Try running the game! in `**src -> main -> java -> nl.tudelft.jpacman**`, you'll find a class called **Launcher**. Right-click it and choose `**'Run Launcher.main()'**`. After a few moments the JPacman window will appear. Click **Start,** and you can move Pac-Man around with the arrow keys. Try to avoid the ghosts. :)
4. To quit, close the JPacman window.

Gradle (optional)

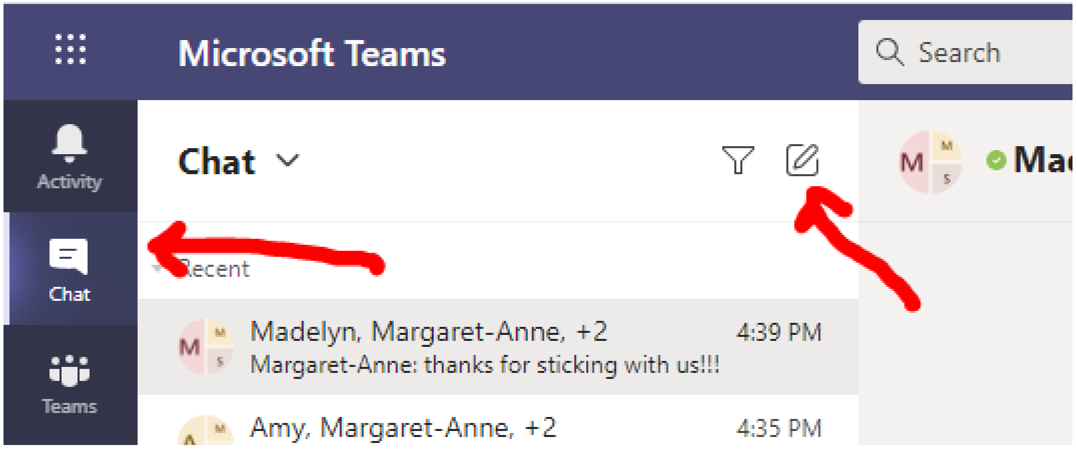
IntelliJ comes with version 5.3 of Gradle, which is fine for our purposes. However, you will occasionally see warning messages like `**Deprecated Gradle features were used in this build...**` in console output. The currently available version of Gradle is 6.7.1, and is available from gradle.org. You can complete everything in the course without installing gradle manually, and you will not be tested on your knowledge of gradle at the command line, but it is a commonly-used build management tool for Java, and familiarity with it may help you in your future career.

If you choose to investigate further, try running some of the built-in tests with gradle from the command line. From the JPacman project's main directory, try running `**gradlew check**`, or `**gradlew test**`, or `**gradlew staticAnalysis**`. Note the output, both on the console and in the reports generated in `**build/reports**`. Now check the `**build.gradle**` file, which is where these three commands are defined - can you see which tests were run for each?

What to submit

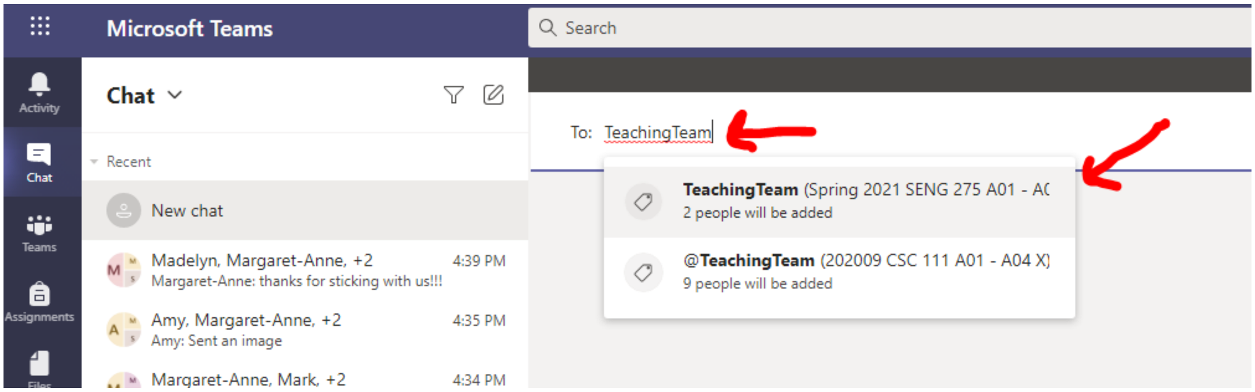
Your task this week is to prove that you have completed the above steps. In order to do this, take a screenshot of your machine running the JPacman game, as described above. The exact procedure for creating a screenshot will depend on your particular operating system - please check your systems help files or the internet for guidance. Only one screenshot is required, and any size or format is fine.

Submit this screenshot to us by logging into the SENG275 Team and sending a message to **TeachingTeam** with your screenshot as an attachment.



Madelyn, Margaret-Anne, +2 4:39 PM
Margaret-Anne: thanks for sticking with us!!!

Amy, Margaret-Anne, +2 4:35 PM



To: TeachingTeam

TeachingTeam (Spring 2021 SENG 275 A01 - A(
2 people will be added

@**TeachingTeam** (202009 CSC 111 A01 - A04 X)
9 people will be added

Madelyn, Margaret-Anne, +2 4:39 PM
Margaret-Anne: thanks for sticking with us!!!

Amy, Margaret-Anne, +2 4:35 PM
Amy: Sent an image

Margaret-Anne, Mark, +2 4:34 PM