A Beginner's Guide to Project 1

11-711: Algorithms for NLP

Fall 2017

Table of Contents

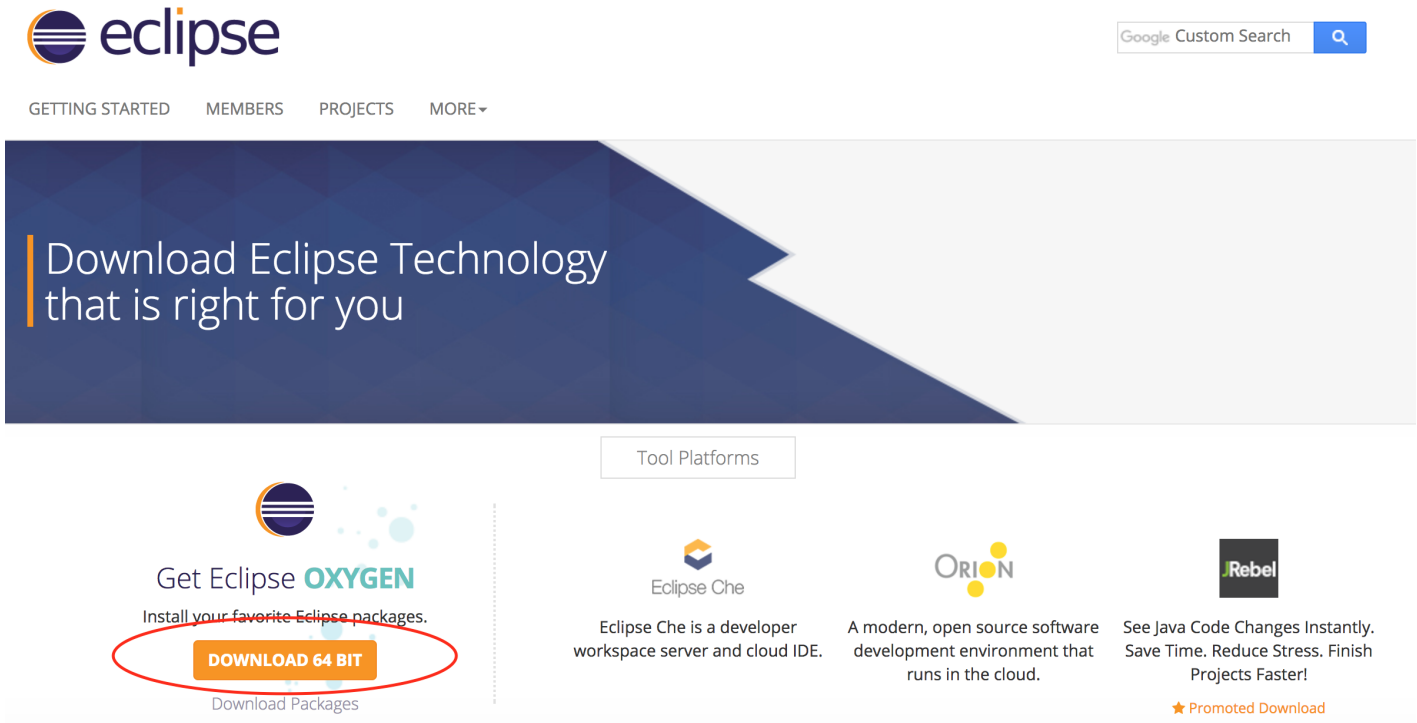
Carnegie Mellon

Outline

Install Eclipse

Carnegie Mellon

- Link: `https://www.eclipse.org/downloads/`



- Follow their instructions

Is Your Mac 64-bit?

- May create unexpected results with `HashMap`, `HashSet`, etc.
- What you see on your computers may be different from what we see when grading
- Do this in your Terminal

```
$ getconf LONG_BIT
```

- Most of your Mac computers are compatible with 64 bits.

Install Apache Ant

- On Mac

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
$ brew install ant
```

- On Linux

```
$ sudo apt-get update
$ sudo apt-get install ant
```

- On Windows
  - DIY: `https://www.mkyong.com/ant/how-to-install-apache-ant-on-windows/`

Outline

Download Project Code

- Starter code:

  `https://storage.googleapis.com/11711/code1.tar.gz`

- Data:

  `https://storage.googleapis.com/11711/data1.tar.gz`

- Unzip

```
$ tar -xzf code1.tar.gz
$ tar -xzf data1.tar.gz
```

Try Running the Starter Code

- Go to `code1`

```
$ ant -f build_assign1.xml
$ export DATA_PATH="/absolute/path/to/your/data/folder"
$ java -cp assign1.jar:assign1-submit.jar -server -mx500m \
  edu.berkeley.nlp.assignments.assign1.LanguageModelTester \
  -path $DATA_PATH -lmType UNIGRAM
```

Automatize Your Workflow

- Create a `run.sh` script that launches your experiment

```
$ vim run.sh
$ # copy the scripts from the previous slide
$ # changing UNIGRAM to TRIGRAM
$ chmod 777 run.sh
```

- Now you can run your experiment with 1 command

```
$ ./run.sh
```

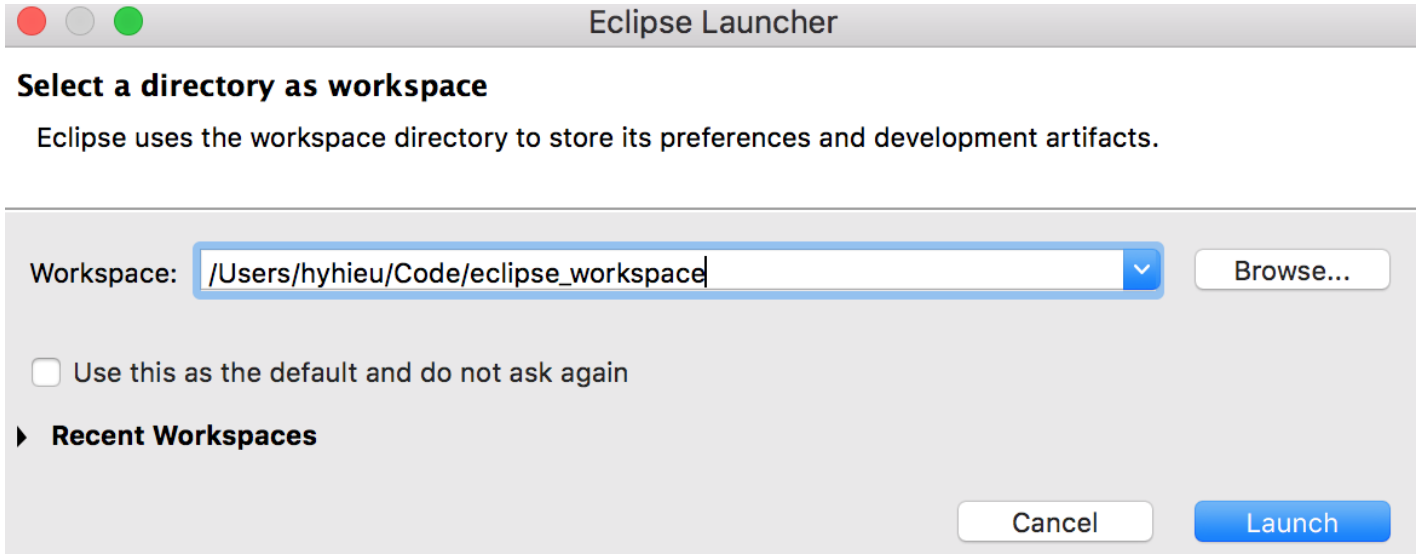
- Much less frustration when working on large projects

Outline

Launch Eclipse and Create your Workspace

Carnegie Mellon

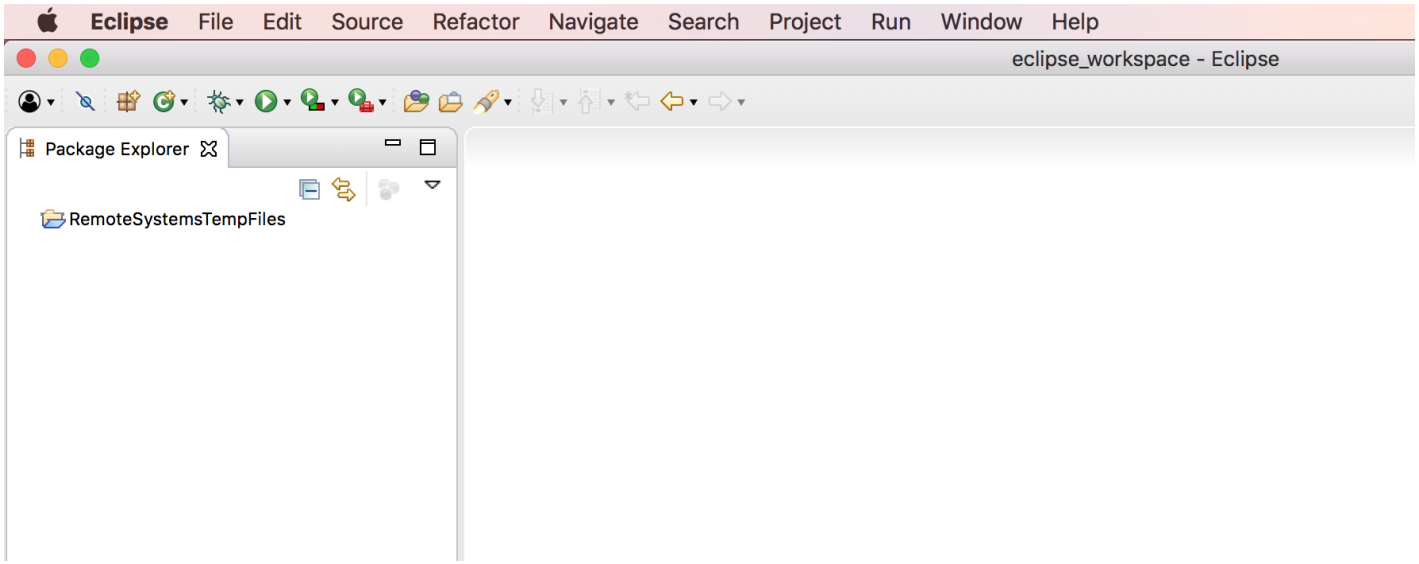
- This is **not** where your code lives



Launch Eclipse and Create your Workspace

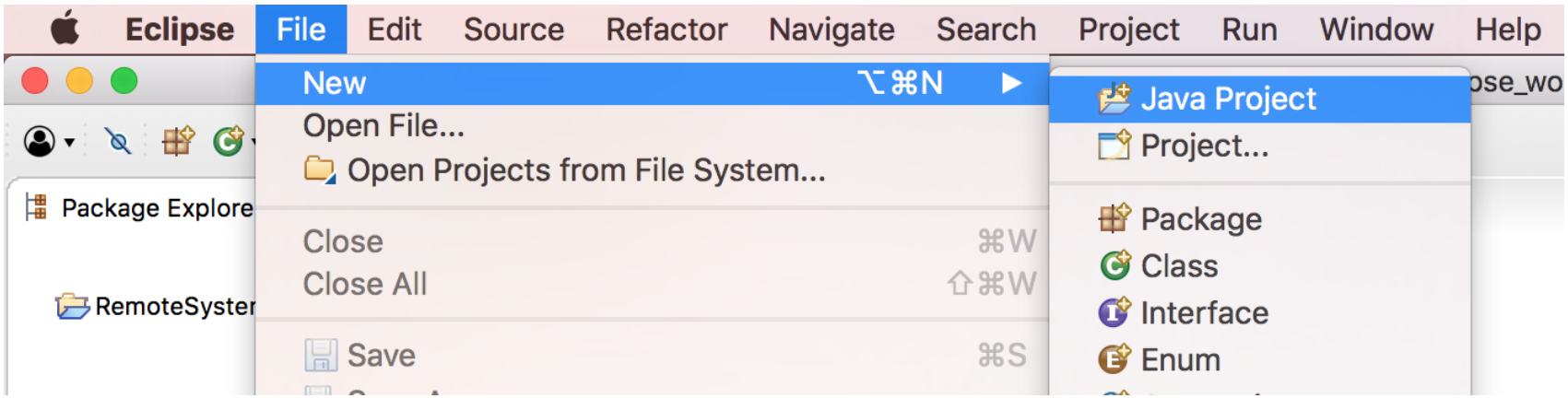
Carnegie Mellon

- Looks like this when done correctly



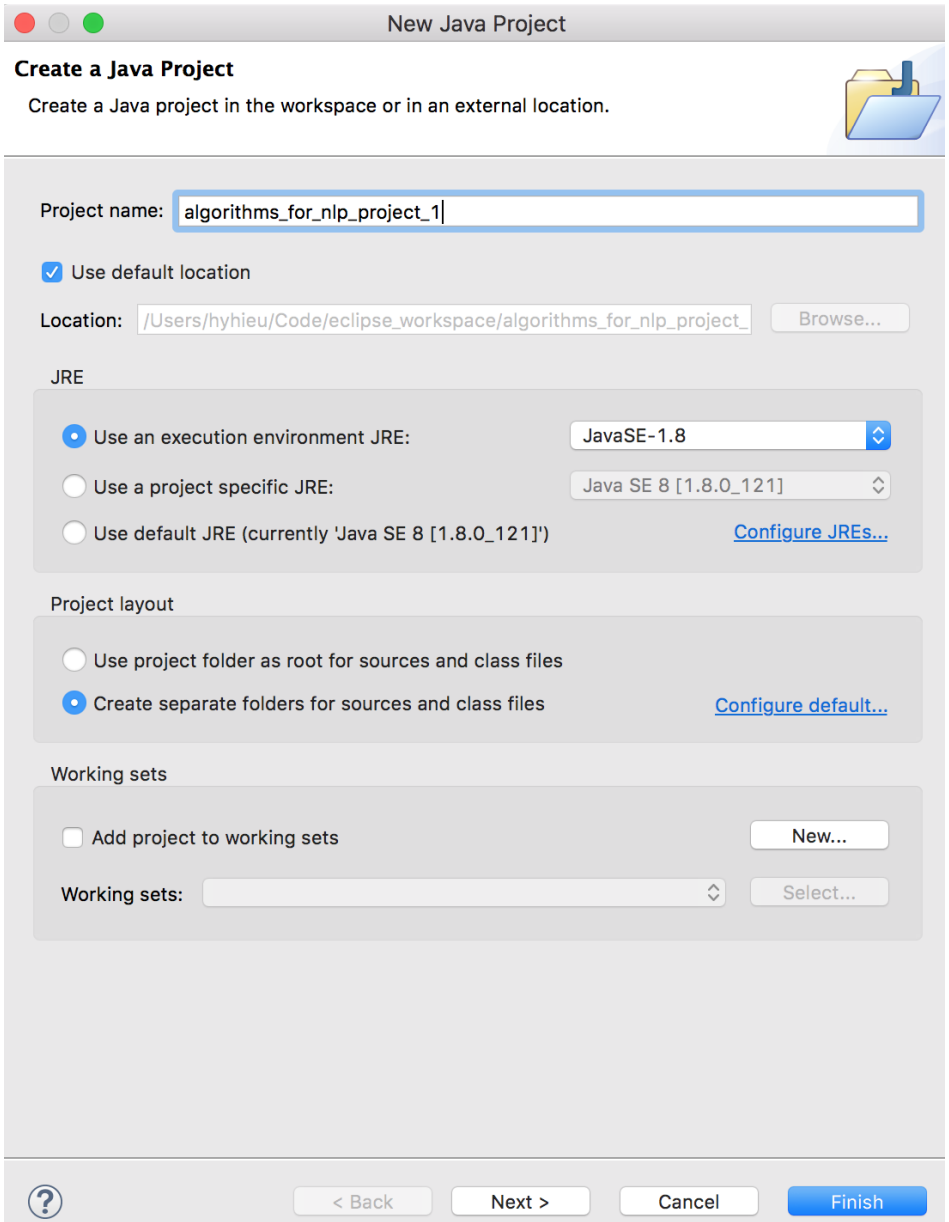
Create Your Eclipse Project

Carnegie Mellon



Create Your Eclipse Project

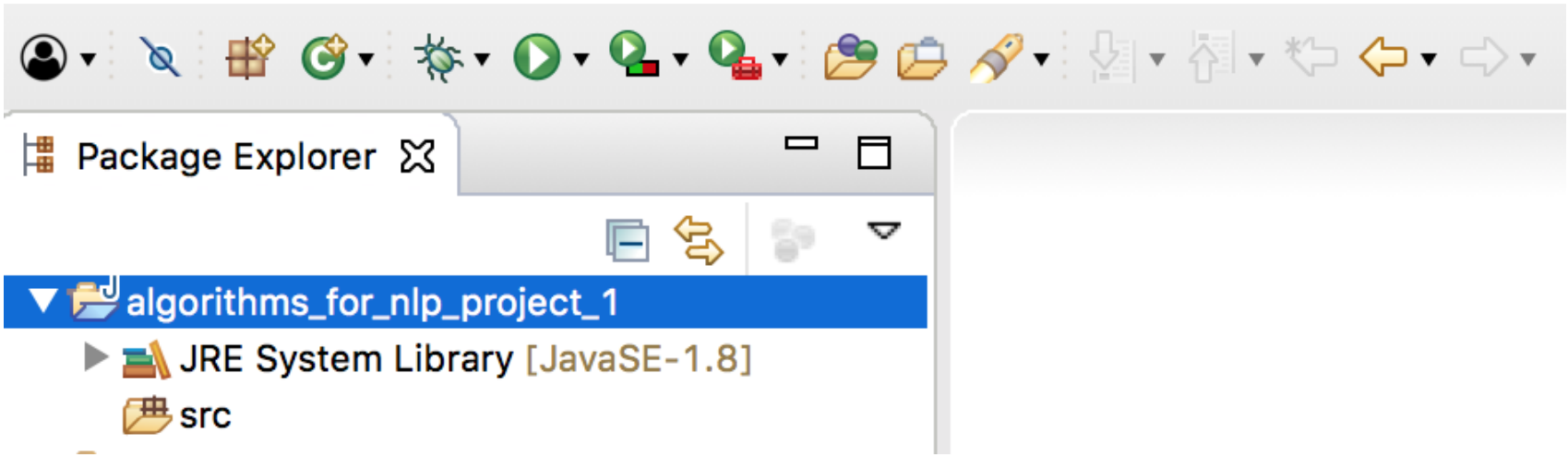
Carnegie Mellon



Create Your Eclipse Project

Carnegie Mellon

- The project folder lives in your workspace

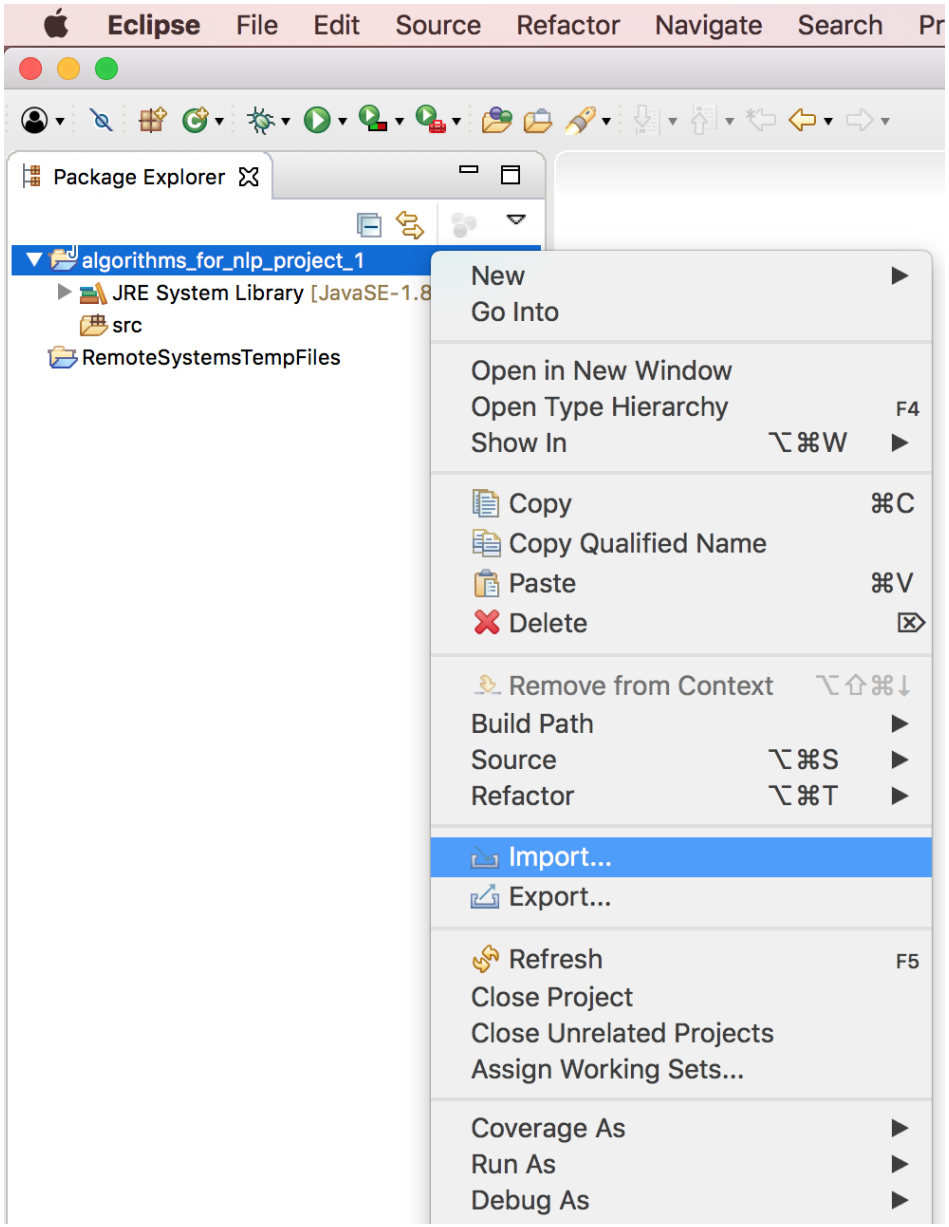


- The code **does not.**

Outline

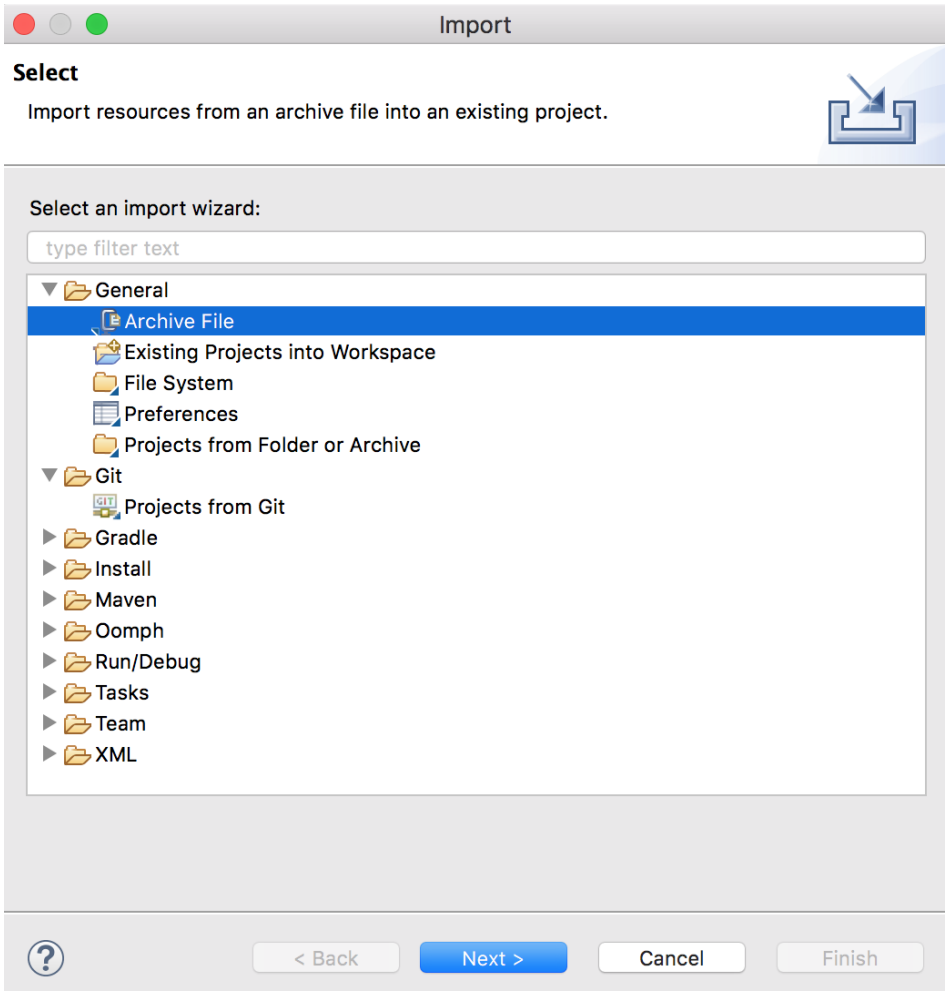
Import Code

Carnegie Mellon



Import Code

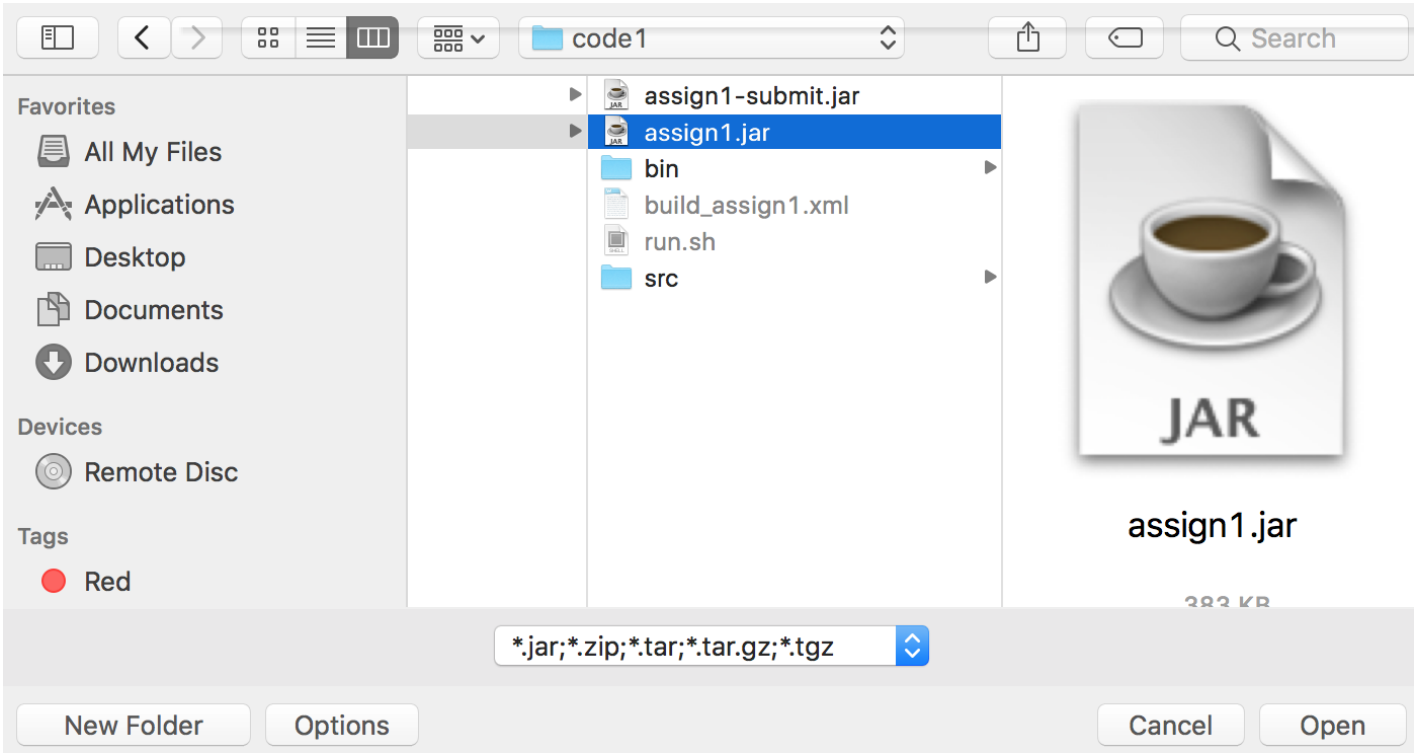
Carnegie Mellon



Import Code

Carnegie Mellon

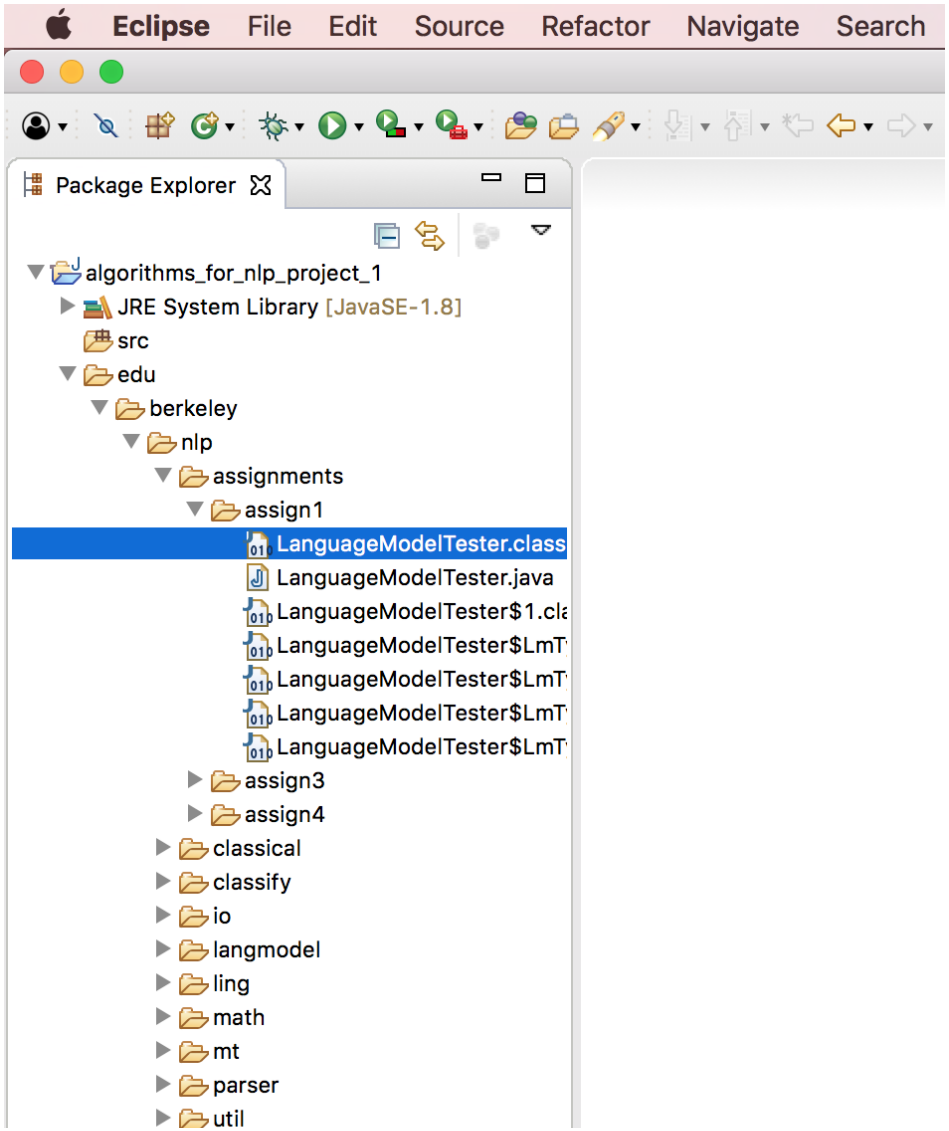
- Point to where you downloaded the code



Import Code

Carnegie Mellon

- Looks like this if done correctly

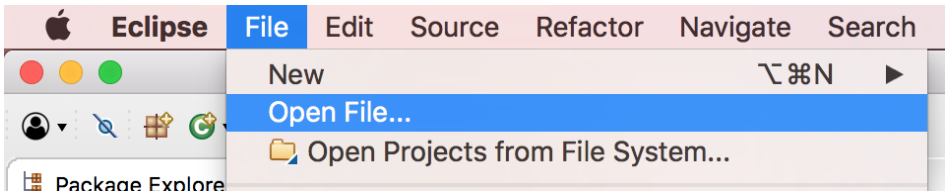


Outline

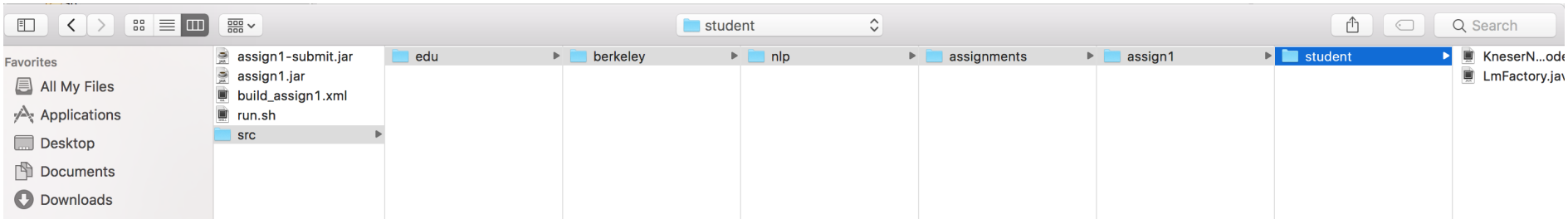
Write Your Code

Carnegie Mellon

- Open the file `LmFactory.java`



- It's tricky



Write Your Code

Carnegie Mellon

- Looks like this if done correctly

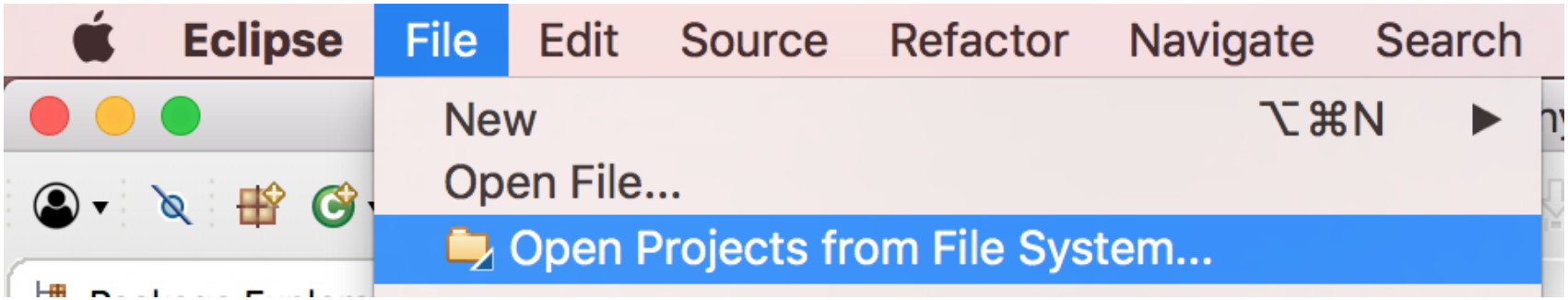
LmFactory.java

```
1  package edu.berkeley.nlp.assignments.assign1.student;
2
3  import java.util.List;
4
5  import edu.berkeley.nlp.langmodel.LanguageModelFactory;
6  import edu.berkeley.nlp.langmodel.NgramLanguageModel;
7  import edu.berkeley.nlp.assignments.assign1.student.KneserNeyLanguageModel;
8
9  public class LmFactory implements LanguageModelFactory
10 {
11
12   /**
13    * Returns a new NgramLanguageModel; this should be an instance of a class that you implement.
14    * Please see edu.berkeley.nlp.langmodel.NgramLanguageModel for the interface specification.
15    *
16    * @param trainingData
17    */
18    public NgramLanguageModel newLanguageModel(Iterable<List<String>> trainingData) {
19        return new KneserNeyLanguageModel(trainingData);
20    }
21
22 }
```

Open `code1` from Eclipse

Carnegie Mellon

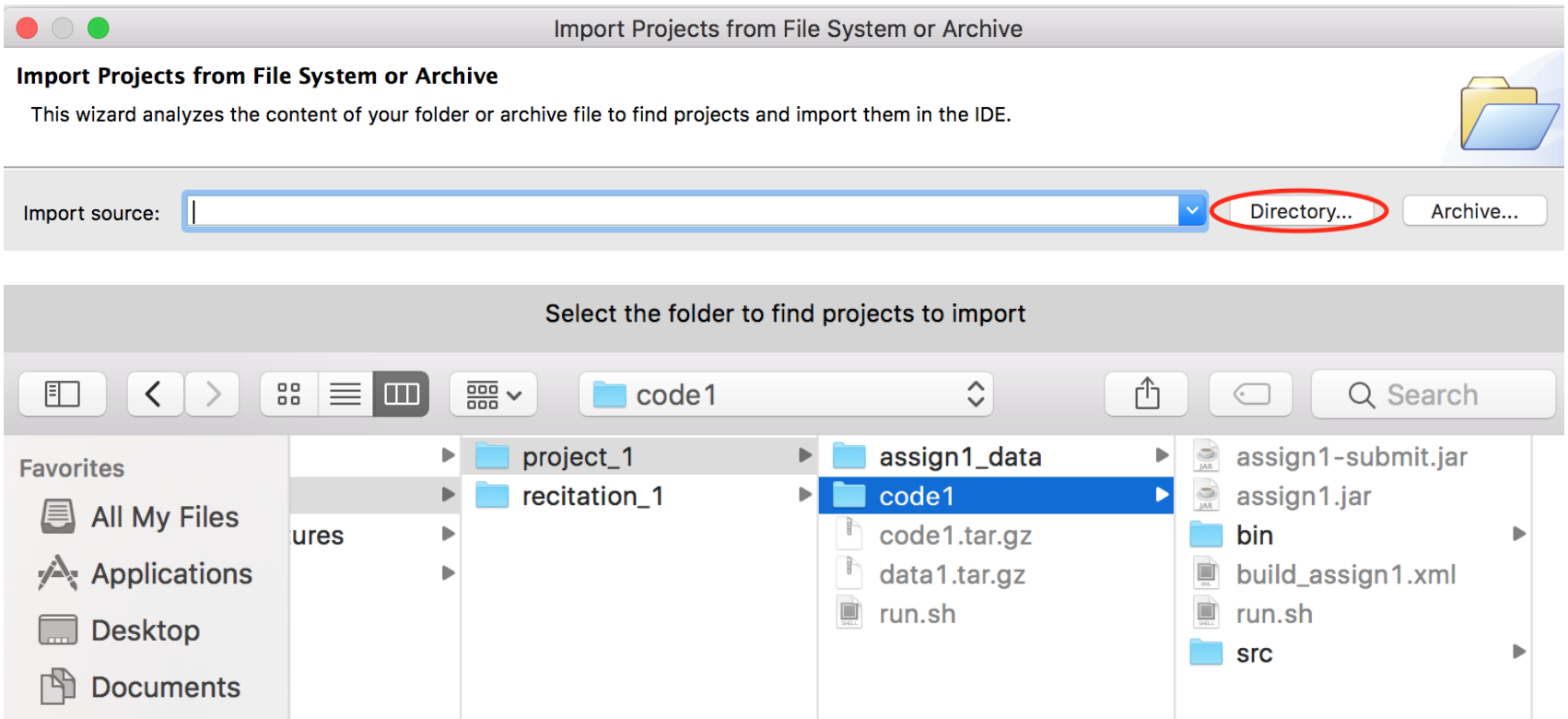
- Easier to navigate and organize your files



Open `code1` from Eclipse

Carnegie Mellon

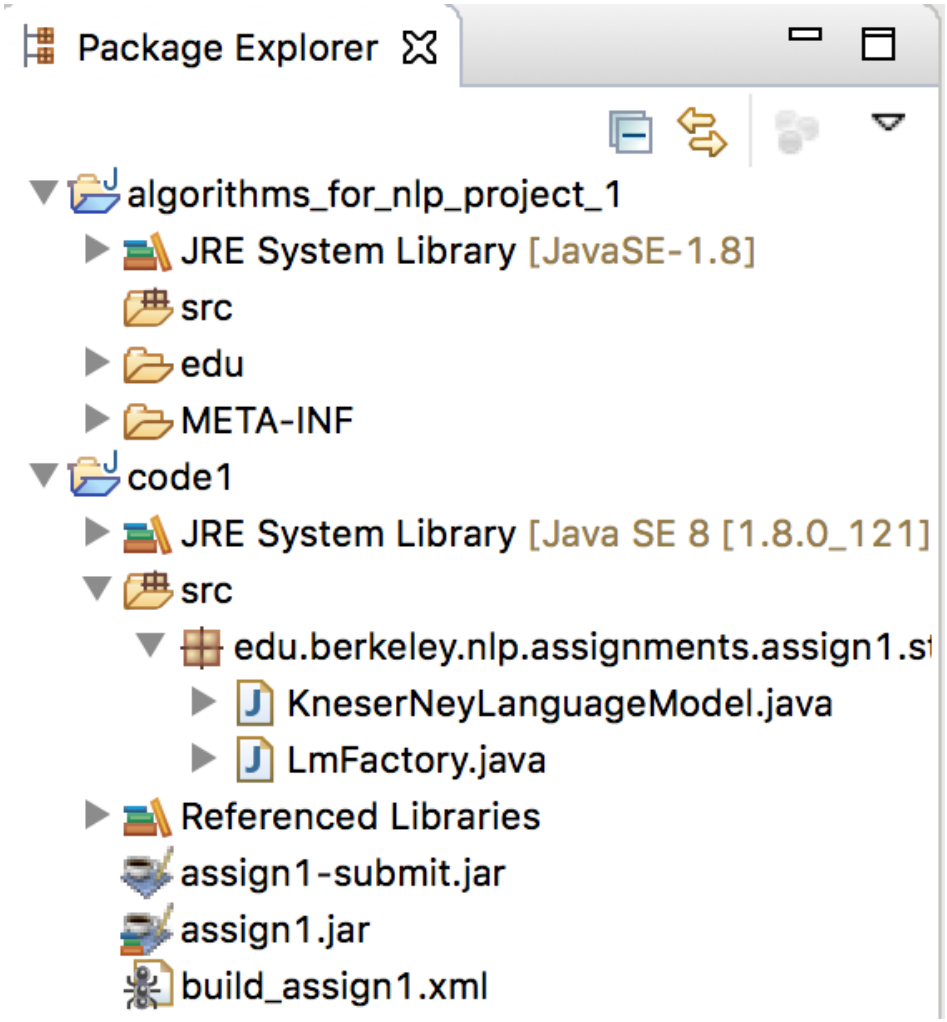
- Select the folder with `build_assign1.xml`



Open `code1` from Eclipse

Carnegie Mellon

- Looks like this if done correctly



Outline

Java Basics

- 15-110: Principles of Computing
- `http://www.cs.cmu.edu/~jxc/100.html`

Questions?