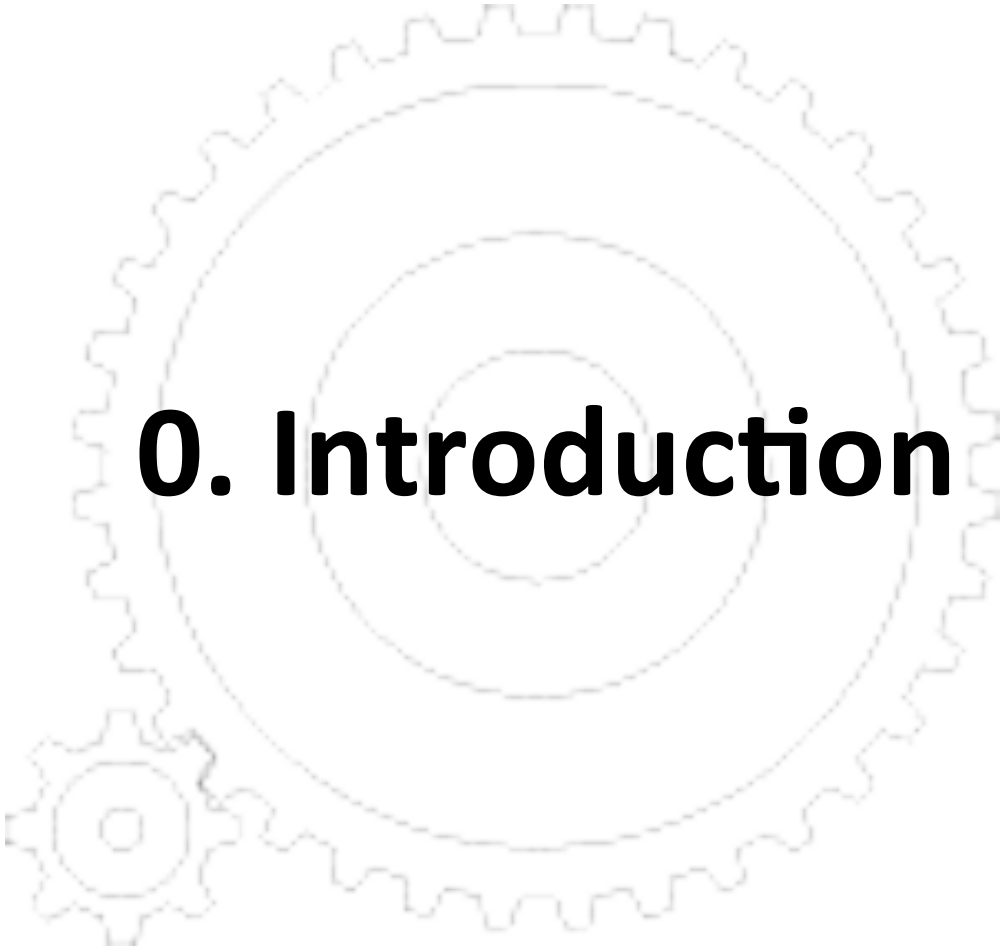
G51OOP



0. Introduction

Object Oriented Programming

Colin Higgins

Teaching Staff

- Lectures : Colin Higgins
- Lab/Coursework helper – Steve Nutbrown
- Lab Assistants - labs, problems/feedback classes
- Web site - http://moodle.nottingham.ac.uk/course/view.php?id=14430

Course Details

- 18 lectures
  - Monday 12:00 : LT3
  - Friday 12:00 : LT2
- Four coursework exercises. Up to 2 days late with standard penalty (5% per day) then 100% penalty
- Lab sessions
  - Tuesdays 14:00 to 16:00.
- Problems classes
  - To be arranged, Thursday or Friday
  - Four classes per week: attend one of them as necessary
  - ~Alternate weeks of feedback/problem solving
- Help desk
  - 14:00 to 16:00 on afternoon of coursework deadline

Course Details continued

- Prerequisites : G51PRG (and a little intelligence and perseverance ☺).
- Objectives
  - to practice and improve coding skills
  - to learn to problem solve
  - to think in an object-oriented manner
  - to express solutions as sound implementations in Java.

Important Notes

- Module is non-compensatable
- House Style! You will be expected to code to a standard layout (although there is some leeway) – see Eclipse IDE.

Assessment & Feedback

- Four lots of coursework that count towards assessment.
- Not all coursework units count equally towards the assessment:

  cw1 = 10%, cw2 = 20%, cw3 = 30%, cw4 = 40%
- Verbal help and feedback via lab assistants in labs/problems class.
- Submission via Moodle.
- Formal feedback via Moodle.
  - Resubmission by permission only.

Grades

- Grades for each coursework are:

A*	90-100
A+	80-89
A	70-79
B	60-69
C	50-59
D	40-49
E	30-39
U	0-29

Resources

- PCs (JDK and Eclipse is installed in all lab PCs).
- Eclipse *integrated development environment* (*IDE*).
- OLD Web Site root is at http://www.cs.nott.ac.uk/~cah/G51OOP/
  - Ignore exercises, dates etc if you look here
- WWW
- Textbooks

Reading

Recommended? (but don't buy unless needed):

- Java Gently by Judy Bishop (Addison-Wes).
- Java. How to Program by Deitel & Deitel (Prentice Hall).
- Developing Java Software by Winder & Roberts (Wiley).
- Java Software Solutions by Lewis and Loftus (Pearson/Addison Wesley).
- Thinking in Java (Bruce Eckel) (available on the web)

Other Sources of Information:

- USE A WEB SEARCH
- Sun Java Tutorial :
  - http://java.sun.com/docs/books/tutorial/getStarted/index.html
- WWW – general : search in www.google.com!
- Intense Java Course : http://www.cs.nott.ac.uk/~azt/java.htm

Programming

- "A program is a sequence of instructions. A recipe, a musical score, and a knitting pattern are all programs."

  P Grogono, *Programming in Pascal*.

- *The way to deal with an impossible task is to chop it down into a number of merely very difficult tasks, and break each one of them into a group of horribly hard tasks, and each one of them into tricky jobs, and each one of them...*

  (Terry Pratchet, Truckers)

Programming expectations

- You have been taught basic coding skills in G51PRG
- We will now progress towards “industrial strength” coding
- This means a working program is only a necessary condition for good marks, NOT sufficient.
- We will also consider quality issues as appropriate eg
  - Design/logic/structure
  - Layout
  - Conventions (eg naming)
  - (testing & documentation?)
  - Etc

Programming Languages

- There are four basic programming language levels:
  - machine language
  - assembly language
  - high-level language
  - fourth-generation language

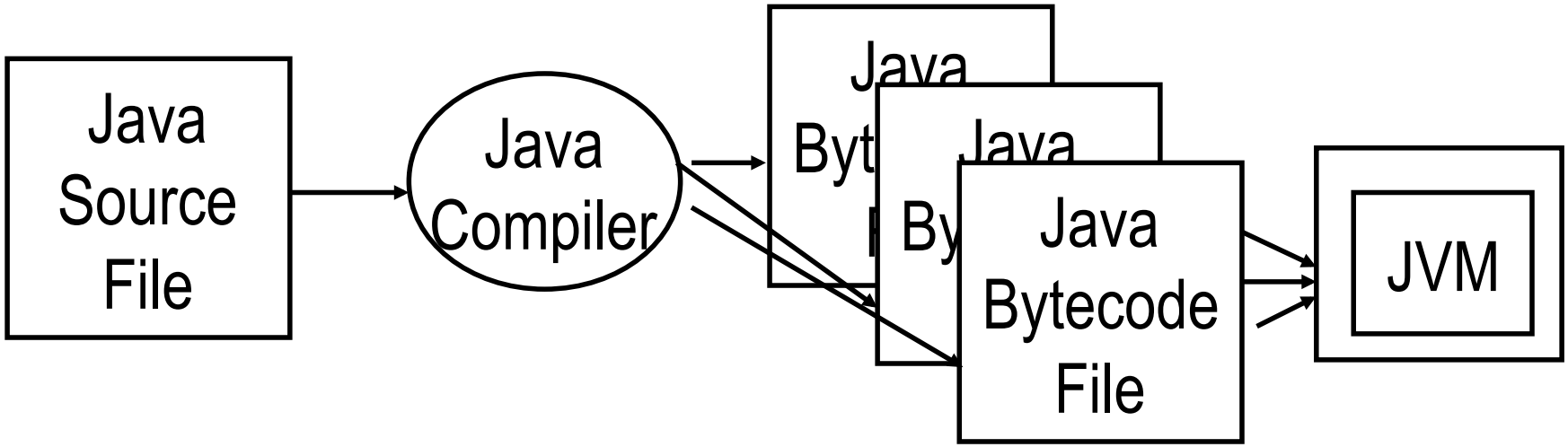
- Each CPU has its own specific *machine language*
- The other levels were created to make programming easier

Programming Languages

- A program must be translated into machine language before it can be executed on a particular type of CPU
- This can be accomplished in several ways
- A *compiler* is a software tool which translates source code into a specific target language
- Often, that target language is the machine language for a particular CPU type
- The Java approach is somewhat different

Java: interpreted – sort of

- The Java compiler translates Java source code into a special representation called *bytecode*
- Java bytecode is not the machine language for any traditional CPU
- Another software tool, called an *interpreter*, translates bytecode into machine language and executes it
- Therefore the Java compiler is not tied to any particular machine
- Java is considered to be *architecture-neutral*



Java Compiled and Interpreted

- Possibly!
- Compilation to Byte code
- Then one of:
  - Interpreted via JVM
  - JIT compiler
  - Ahead of time compiler

Java: What is it?

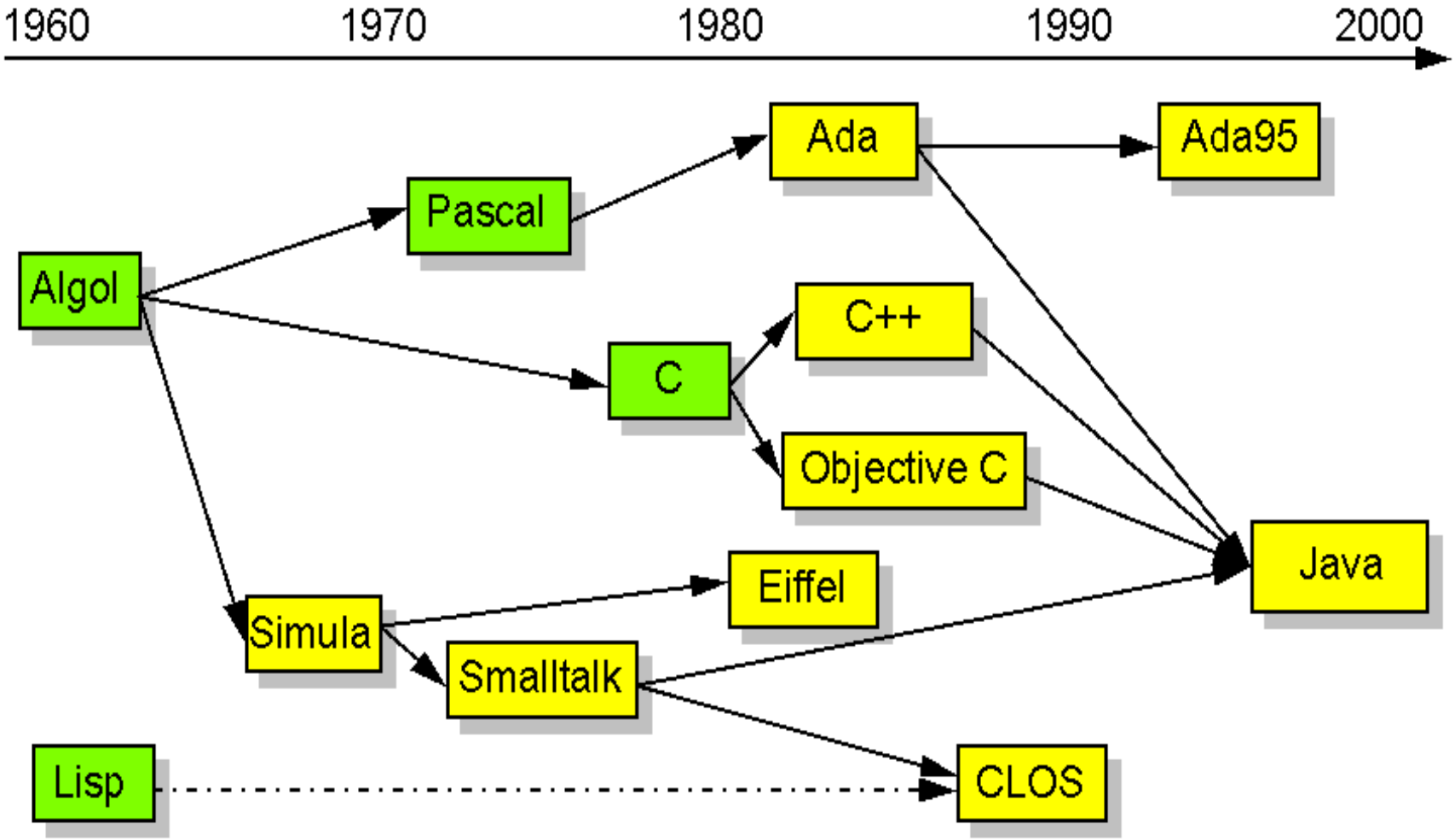
Sun's Description….
Java is a :

- Simple
- Object Oriented
- Distributed
- Interpreted
- Robust
- Secure

- Architecture Neutral
- Portable
- High Performance
- Multithreaded and Dynamic

programming language.
It contains an extensive library of classes for graphics programming, input/output, string handling, maths functions, basic data structures, etc.

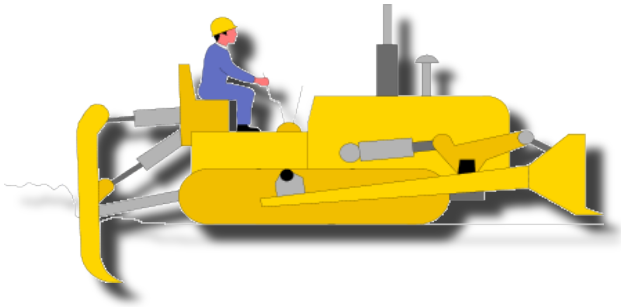
Java's Genealogy



Java : Types of Programs

- Applets :
  - Execute on HTML Browsers.
  - Have severe security restrictions.
- GUI Applications :
  - Are Interpreted and Executed.
  - Use the current's platform's GUI widgets.
  - Mostly use Java's AWT or JFC packages.
- Console Applications :
  - Simple Text Console.

JDK basic tools



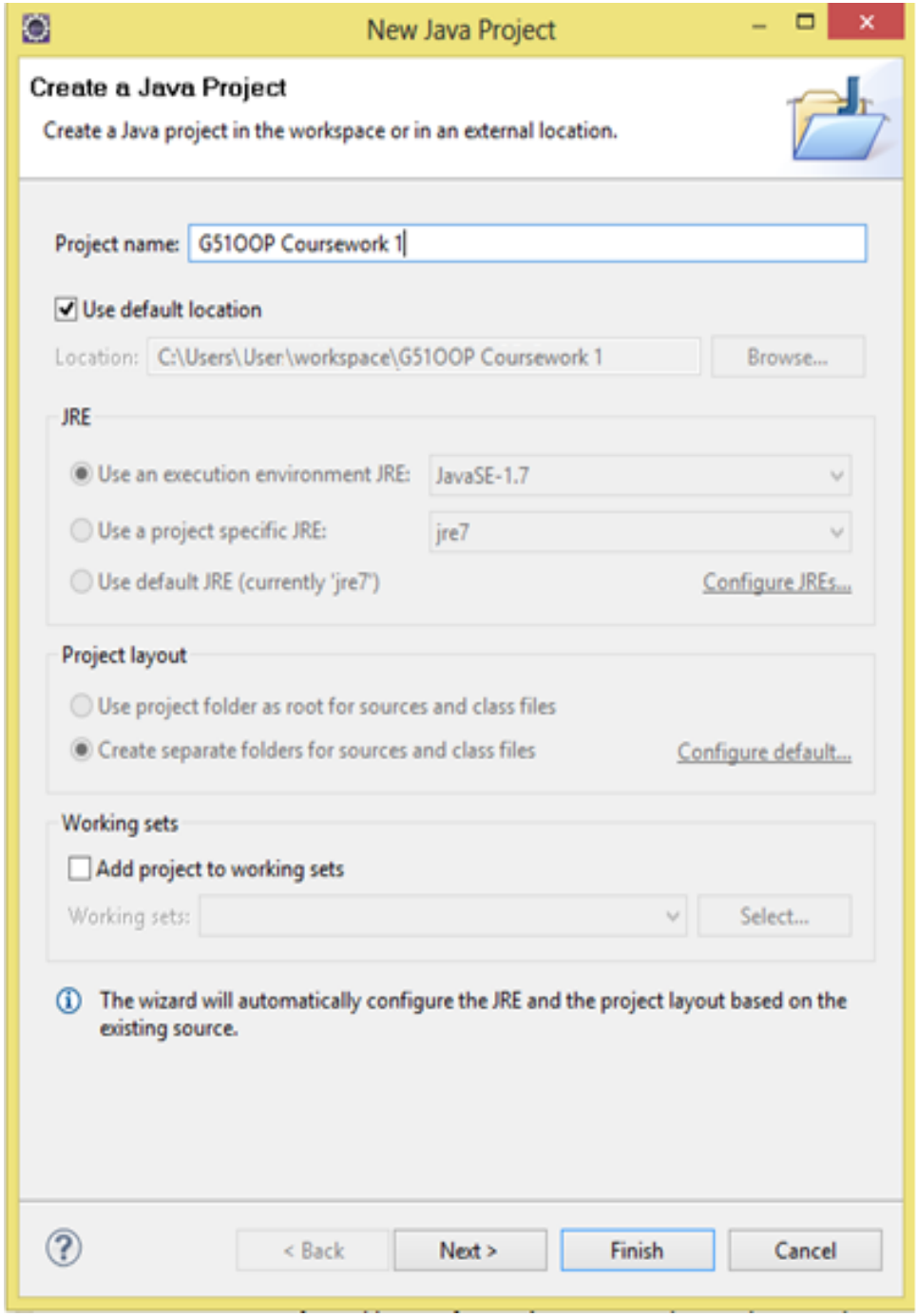
- javac (Compiler) *.java->.class*
  - The Java Language Compiler that you use to compile programs written in Java into bytecodes.
    Example : `javac HelloWorld.java`
- java (Interpreter) *executes .class*
  - The Interpreter that you use to run programs written in Java.
    Example : `java HelloWorld`
- javadoc (Doc. generator) *.java html*
  - Generates documentation in HTML format from Java source code.
    Example : javadoc HelloWorld.java

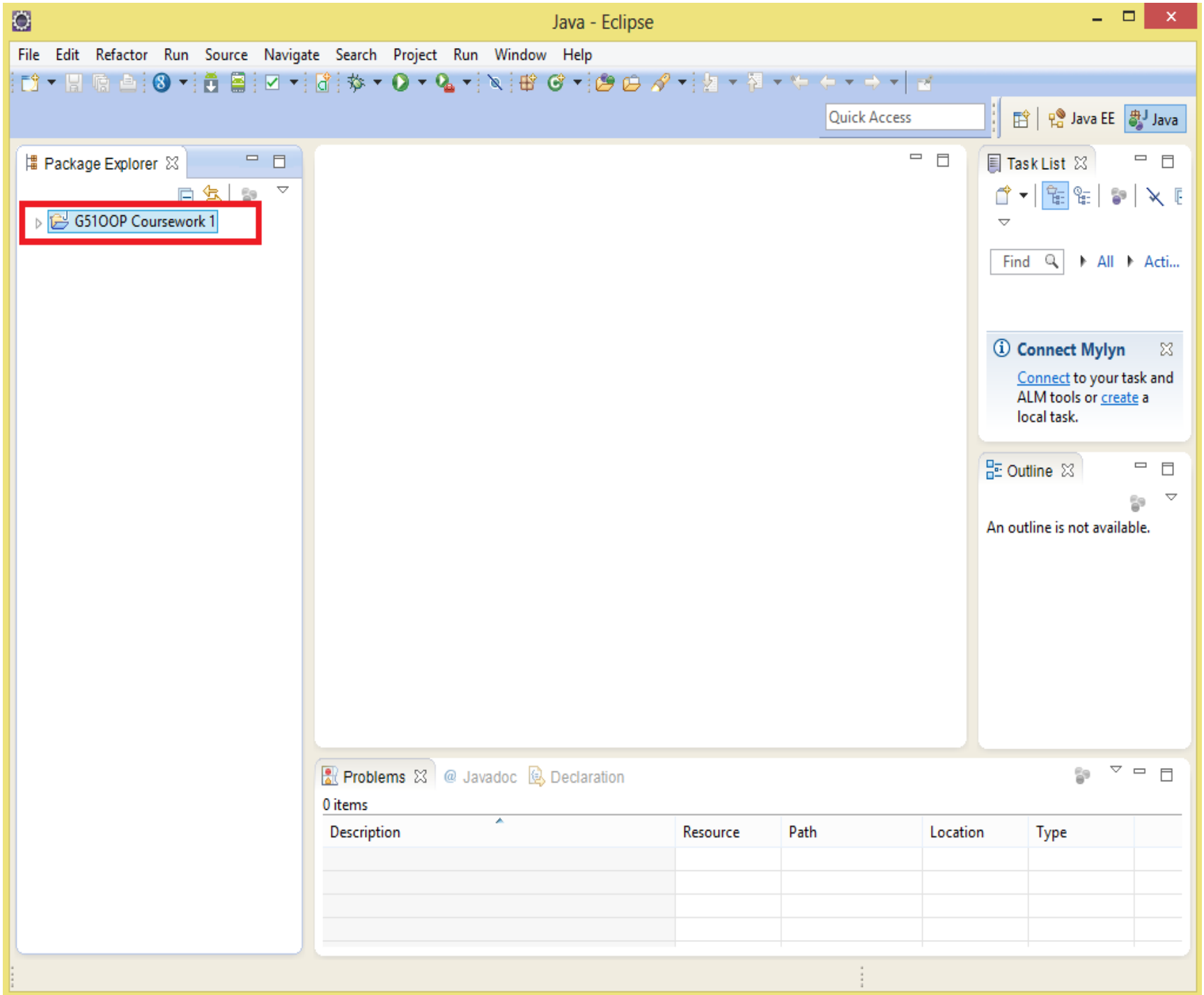
Getting started with Eclipse

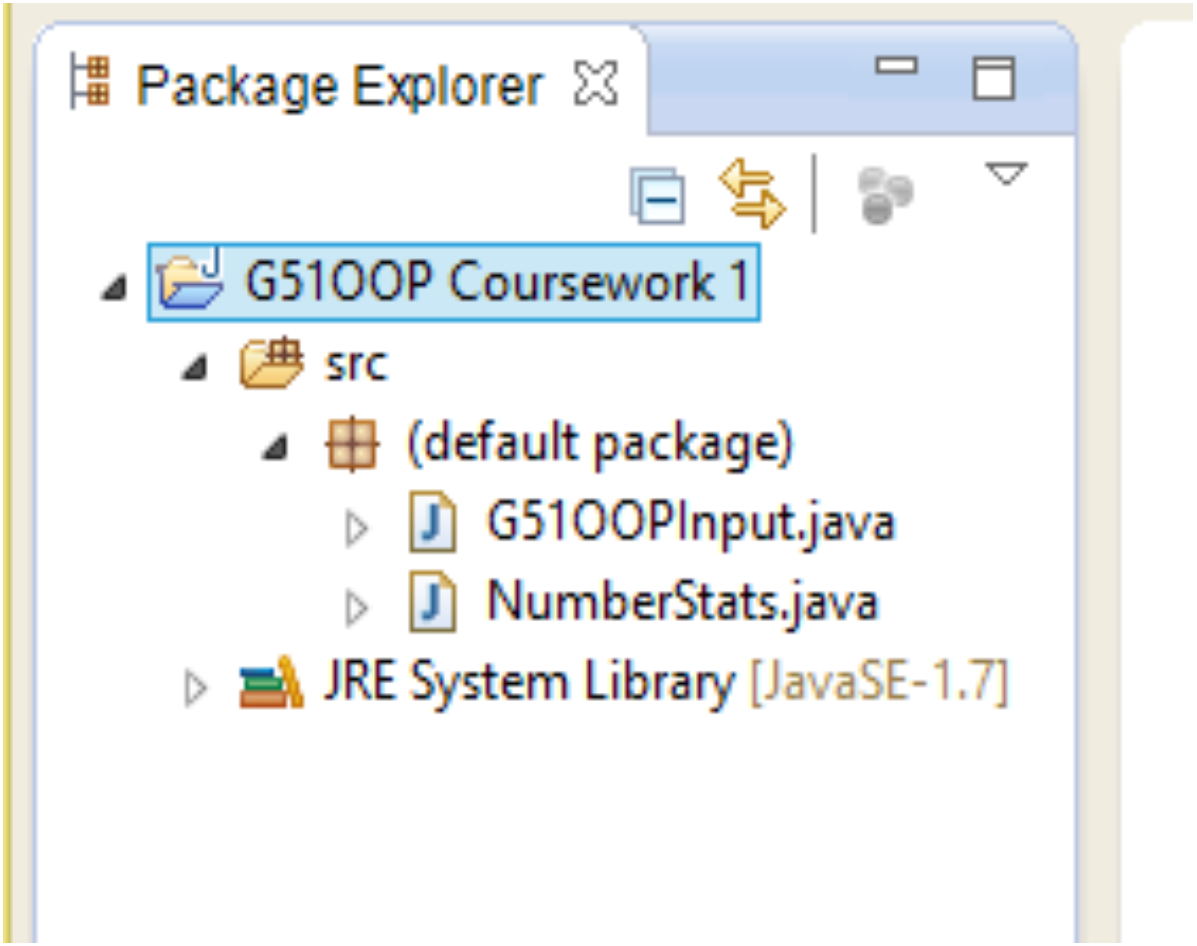
- Download and install the JDK from the link below, making sure to get the correct version for your system:
  - http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Download the latest version of Eclipse Classic from the Eclipse homepage using the link below
  - http://www.eclipse.org/downloads/
- Unzip Eclipse somewhere sensible
- Open eclipse

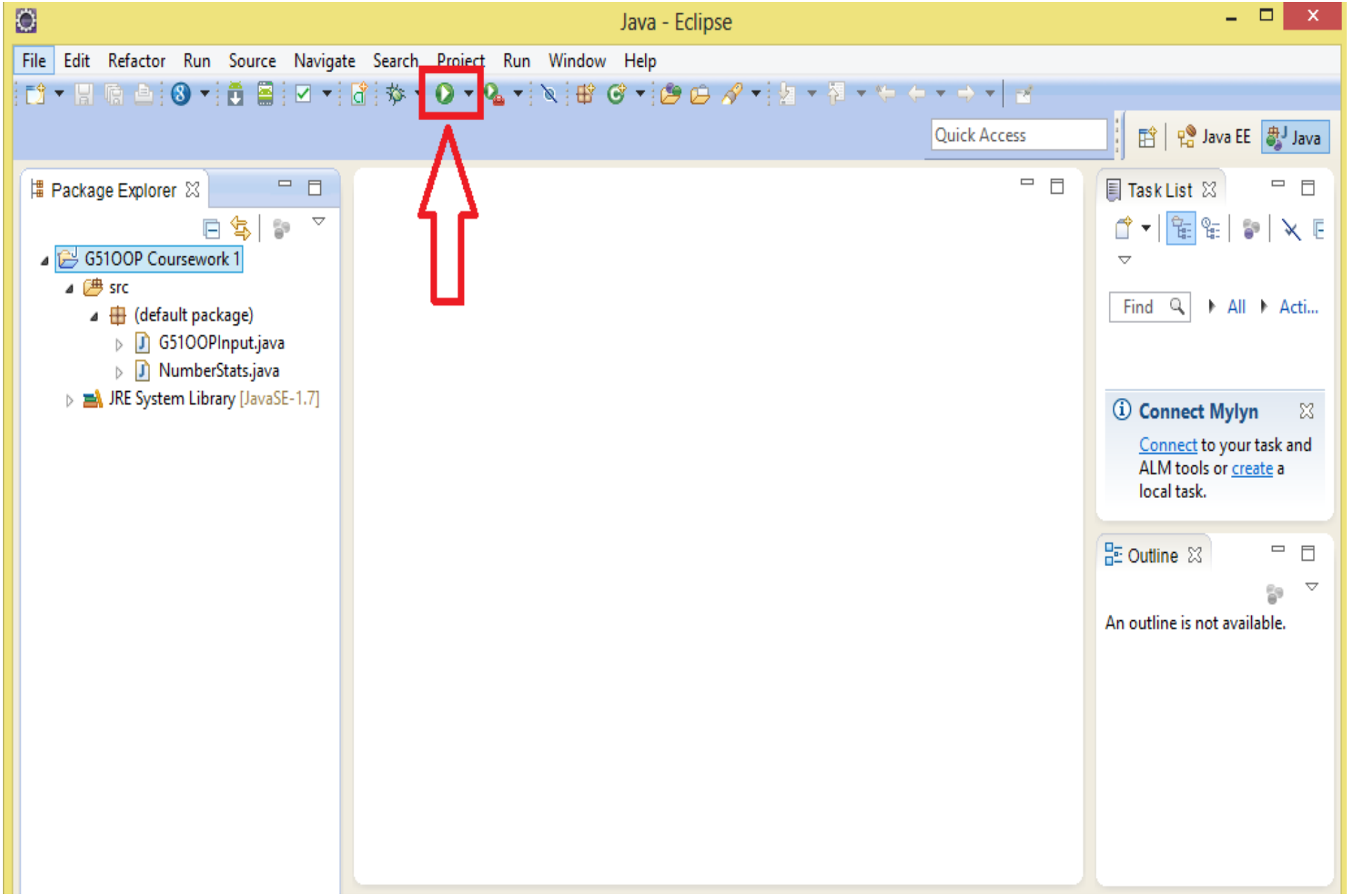
- Create a new project
  - Do this for each coursework
- File > New > Java Project
- Use a meaningful name

  eg G51OOPcw1




Java - Eclipse
File Edit Refactor Run Source Navigate Search Project Run Window Help
Quick Access
Java EE
Java
Package Explorer
G51OOP Coursework 1
Task List
Find
All
Acti...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
An outline is not available.
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
Type


Package Explorer
G51OOP Coursework 1
src
(default package)
G51OOPInput.java
NumberStats.java
JRE System Library [JavaSE-1.7]


Java - Eclipse
File
Edit
Refactor
Run
Source
Navigate
Search
Project
Run
Window
Help
Quick Access
Java EE
Java
Package Explorer
G51OOP Coursework 1
src
(default package)
G51OOPInput.java
NumberStats.java
JRE System Library [JavaSE-1.7]
Task List
Find
All
Acti...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
An outline is not available.

The First Java Program

Filename: HelloWorld.java

```
public class HelloWorld {

      public static void main(String argv[] ) {
          System.out.println("Hello World!!!");
      }
}
```