Announcements

- Office hours are active
- Project Team creation due tonight at 11:59 PM
- All unassigned students will be randomly assigned to a group by tomorrow, 31st Aug- 9 AM
- Post with team members even if team is not complete
- GCP credits will be assigned today. Look out for an Ed post tonight.
- Assignment 1 out today- Project Planning

Assignment 1: Project Planning

- Discuss with your team-members
- Decide on the:
  - Project Lifecycle
    - Project process (Agile vs. Waterfall vs. Evolutionary Prototyping)
    - Possible deviations
    - Timeline
  - Roles and Responsibilities
  - Communication (meeting frequency, venue)
  - Risk management
  - Planning & Control
  - Technologies

Assignment 1 Grading Rubric

INTRODUCTION: (1/5)

MANAGEMENT STRUCTURE: (1/5)

RISK MANAGEMENT: (1/5)

PROJECT PLANNING: (1/5)

TECHNOLOGIES: (1/5)

CS3300 Introduction to Software Engineering

Lecture 03: Tools of the Trade #1

Java, Eclipse, Junit Testing, Maven

Nimisha Roy ▸ nroy9@gatech.edu

Contents

- Frontend *vs.* Backend
- Java
- Object Oriented Programming
- What is an IDE?
- Eclipse (Demo)
- Junit Testing (Demo)
- Maven (Demo)

Frontend *vs.* Backend

Front End development

- Visual representation of user's request within a browser
- Uses front end programming languages to create what the user sees and interacts with
- Backbone: **HTML** (Static Component); **CSS** (Styling Component); **JavaScript** (Interactive Component)
- JavaScript frameworks like React, Angular, Redux are now used to quickly develop user interfaces efficiently
- Angular framework, for example, lets developers build single-page web apps efficiently. **jQuery** simplifies tasks; **AJAX** adds XML, a markup language, to JavaScript to enable selective refreshing of sites.

Back End development

- Uses back-end programming languages to fulfill requests on the server side (logic that makes user requests a reality)
- Languages: **Java**, Python, C#, Ruby, PHP,
- Database management tools: SQL developer, MySQL
- Frameworks include **Spring**, Express, Django, Rails

Frontend *vs.* Backend

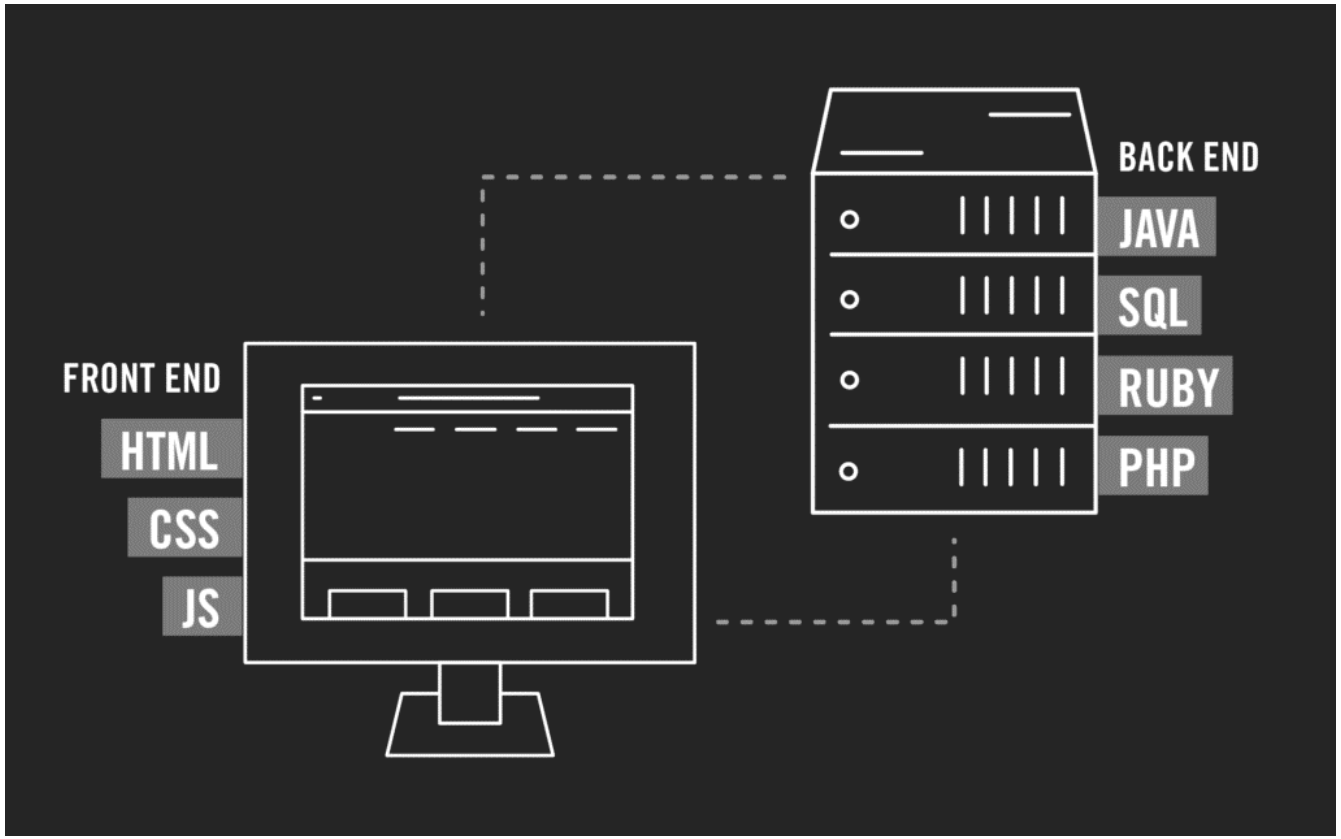


Image courtesy: https://flatironschool.com/blog/front-end-vs-back-end-development

Front End languages communicate requests to Back End languages. Every website has a server, database, and other applications that interact with the Front End through code created by a Back End developer.

Java

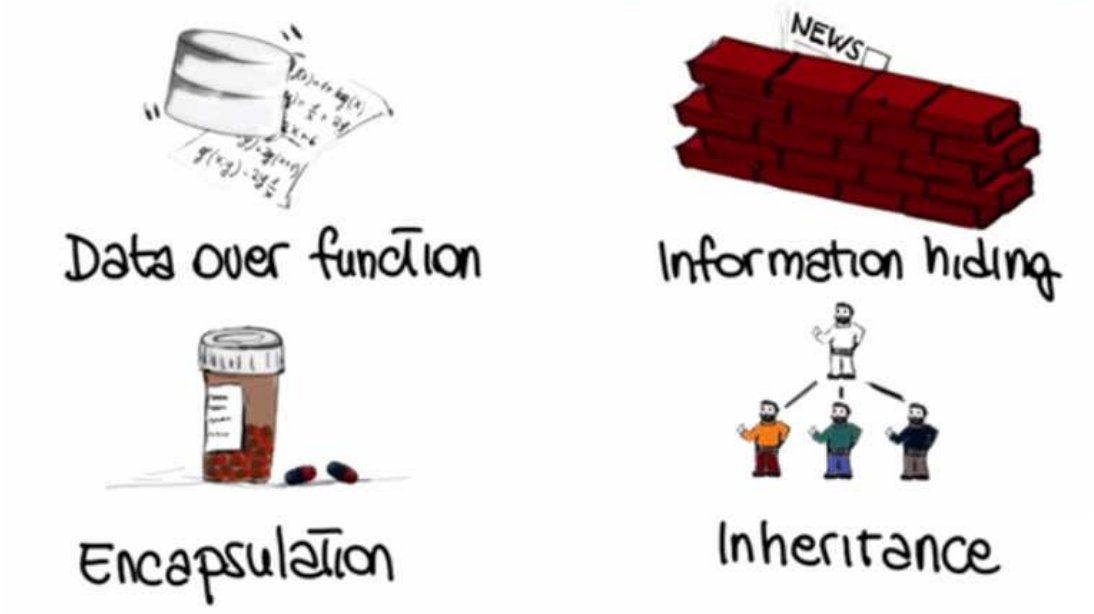
- Object-Oriented Programming (OOP) language and a Platform
  - Platform - Any hardware or software environment in which a program runs.
  - Java has its own runtime environment (JRE) and API- hence a platform.

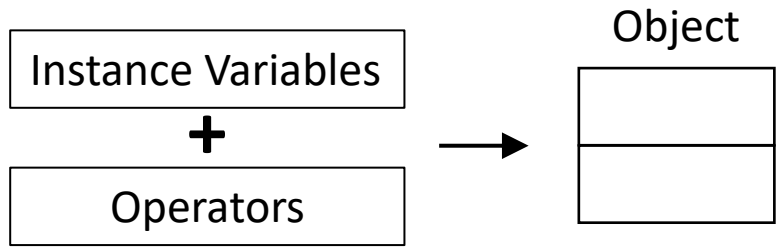
- Developed by James Gosling from Sun Microsystems in 1995.
  - One of the most popular programming languages today.
  - About 5.5 billion devices use Java today
  - Desktop & Web App, Mobile, Embedded system, Smart Card, Games, Robotics etc.

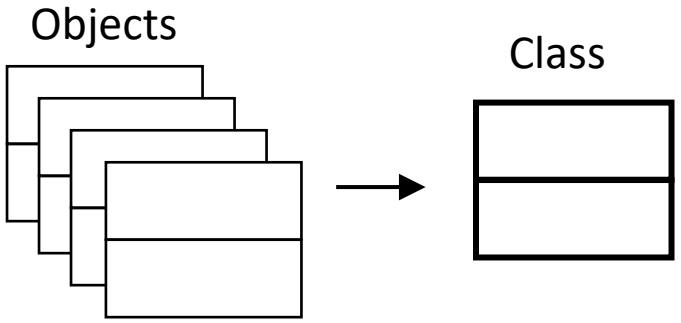
- Portable
  - Can be executed on any machine irrespective of the operating system, as long as it supports Java®.
  - Based on the Java® Virtual Machine (JVM) and the intermediate compilation into bytecode.
  - Source Code to Bytecode by Java compiler; Bytecode is non-executable and platform independent
  - Bytecode interpreted by the JVM.
  - JVM translates the bytecode instructions into machine instructions that your computer can understand and execute.
  - Implementation of JVM is JRE. Run a Java command – instance of JVM created

Object Oriented Programming

Principles of Object Orientation







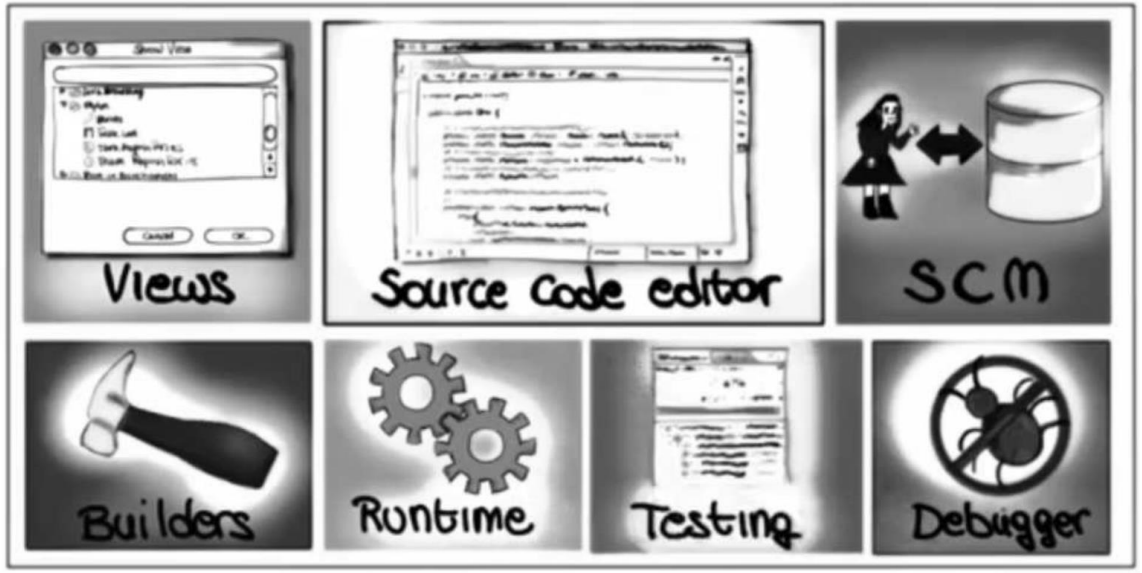
Why use OO?

- Reduce maintenance Costs – by limiting effects of changes (Encapsulation & Information Hiding)
- Improve development Process – Favoring Code & Design Reuse
- Enforce Good design Principles – high cohesion, low coupling

Slide adapted from Alessandro Orso

What is an IDE?

- Integrated Development Environments (IDEs) are software applications that support developers in many of their everyday tasks, such as writing, compiling, and debugging code.



- Some IDEs are designed to support only one programming language such as Java, while others can be used for various languages
- Plug-ins: additional functionality offering more features to IDEs, not available in core. Eg: Egit plug-in which adds support for Git Version Control
- Almost all IDEs work on Mac, Windows and Linux. Careful about version of IDE based on OS
- Most popular Java IDEs: IntelliJ IDEA, Eclipse, Netbeans

Slide adapted from Alessandro Orso

Eclipse

Eclipse is an open, extensible Java IDE that was initially created by IBM and is now managed by the Eclipse Foundation

DEMO TIME: Create Java Project in Eclipse

- How to Install JDK
- How to Install and setup Eclipse/Select Workspace
- How to create a project
- Perspectives & Layout
- How to create package and class within the project
- Run Configuration
- How to use Eclipse debugger.

Eclipse

How to Install JDK

- Browse to the AdoptOpenJDK website.
- Download the **Latest release** JDK zip file (.gz file) for your OS and Java 18.

How to Install and setup Eclipse

- Browse to https://www.eclipse.org/downloads/packages/installer
- Select installer bundle for your platform and download it. Run self-extracting program for the installer.
- Choose “Eclipse IDE for Java Developers” and Click Install. (Note the installation folder and JDK version)
- Launch Eclipse
- Choose your workspace. Workspace is the directory where Eclipse will place all your projects.
- Verify that the JDK version you chose is the one Eclipse is directed to use. Window-- Preferences-- Java-- Installed JREs.

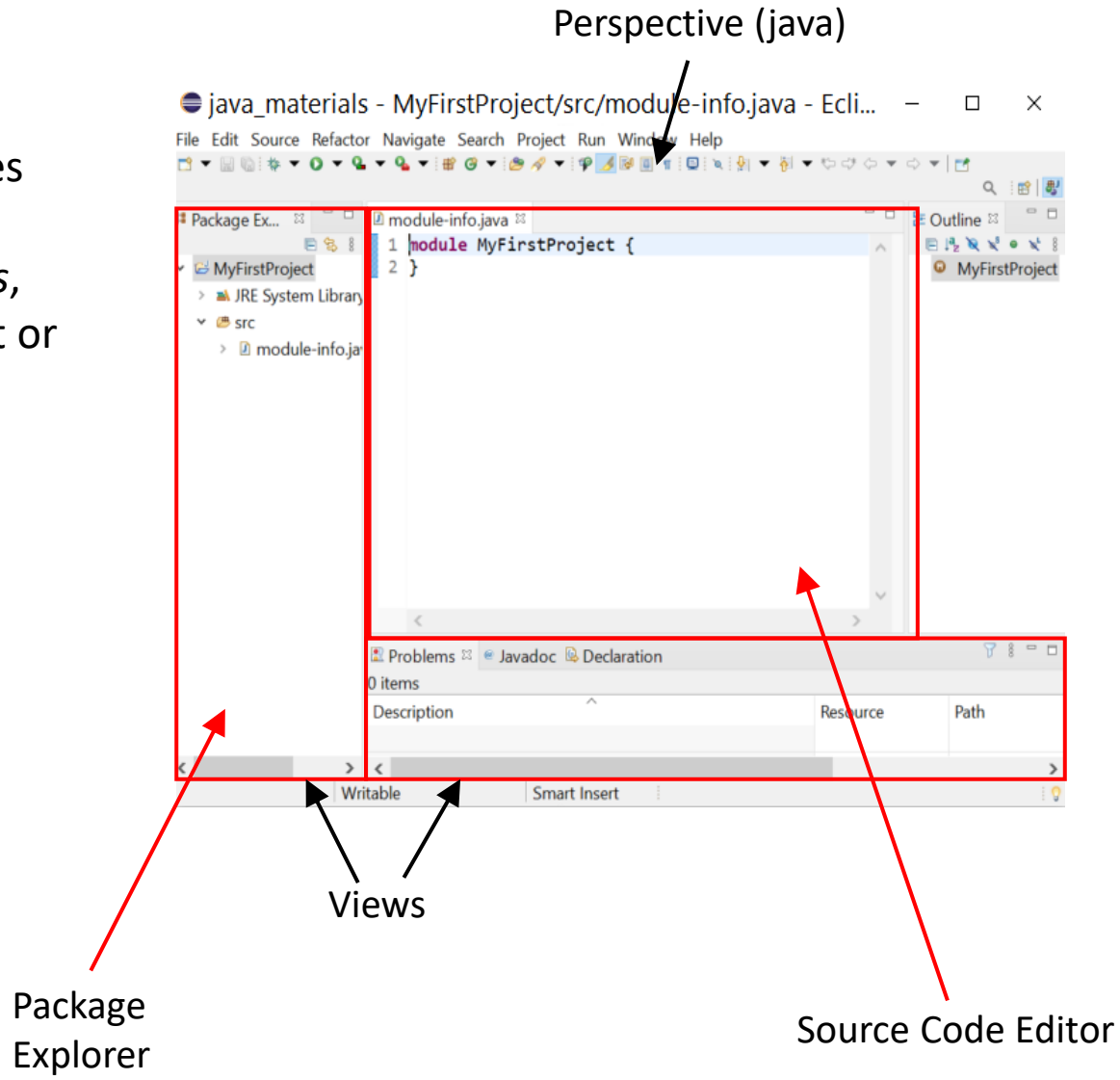
Eclipse

How to create a project

- File—New—Java Project. Name it *myFirstProject*. Eclipse saves the project in the default workspace.
- *Src* :contains the packages you create. Option to add *Libraries*, external *JAR* files. *Order and Export*: which part of the project or how project will be exported

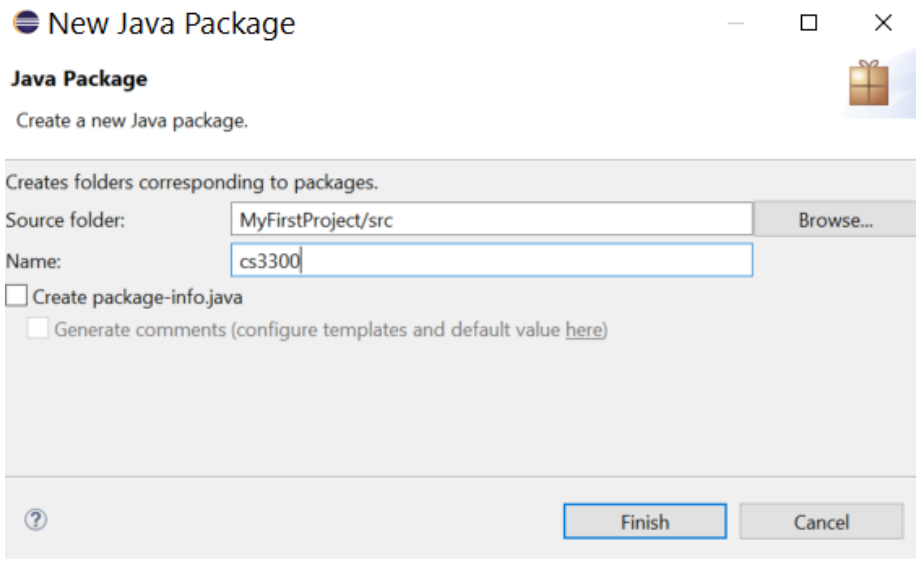
Perspective/Views

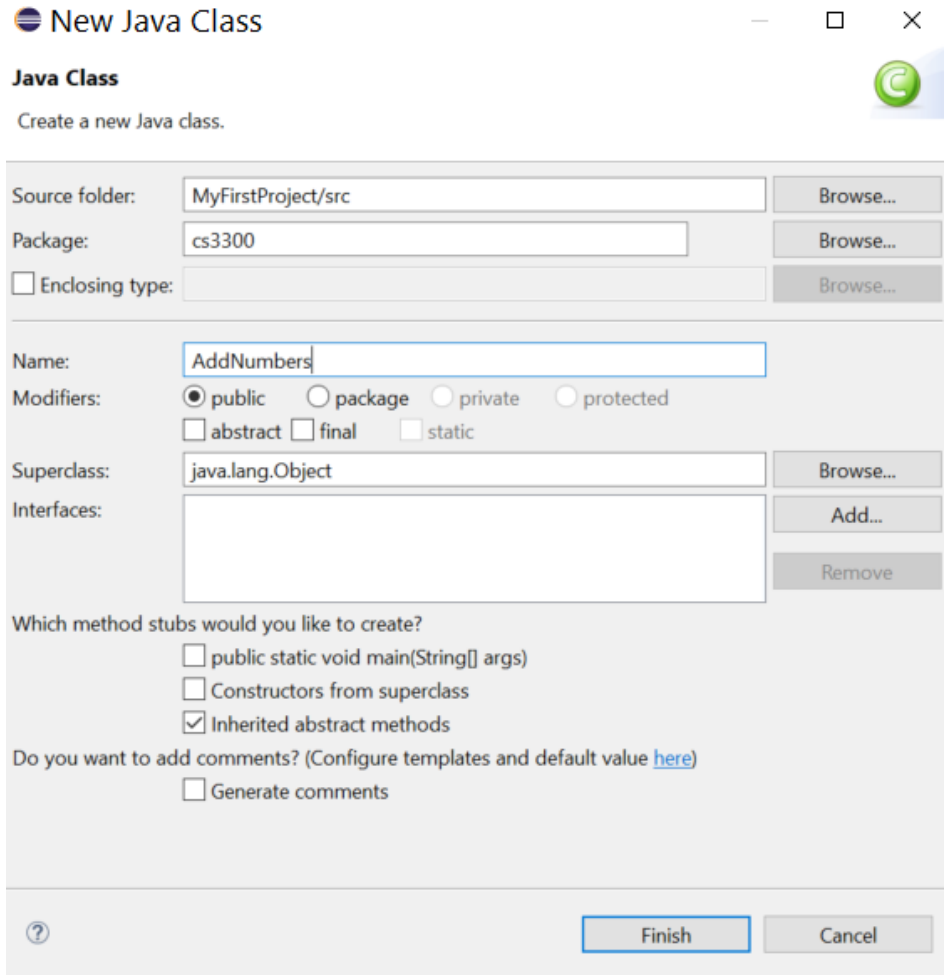
- Perspective: visual container for a set of views and content editors.
- Each perspective can have different Views. *Windows– Show View:* to add view components
- E.g. Java perspective: Package Explorer: organize classes in hierarchy
- *Problems, Javadoc, Declaration, Console*.



Eclipse

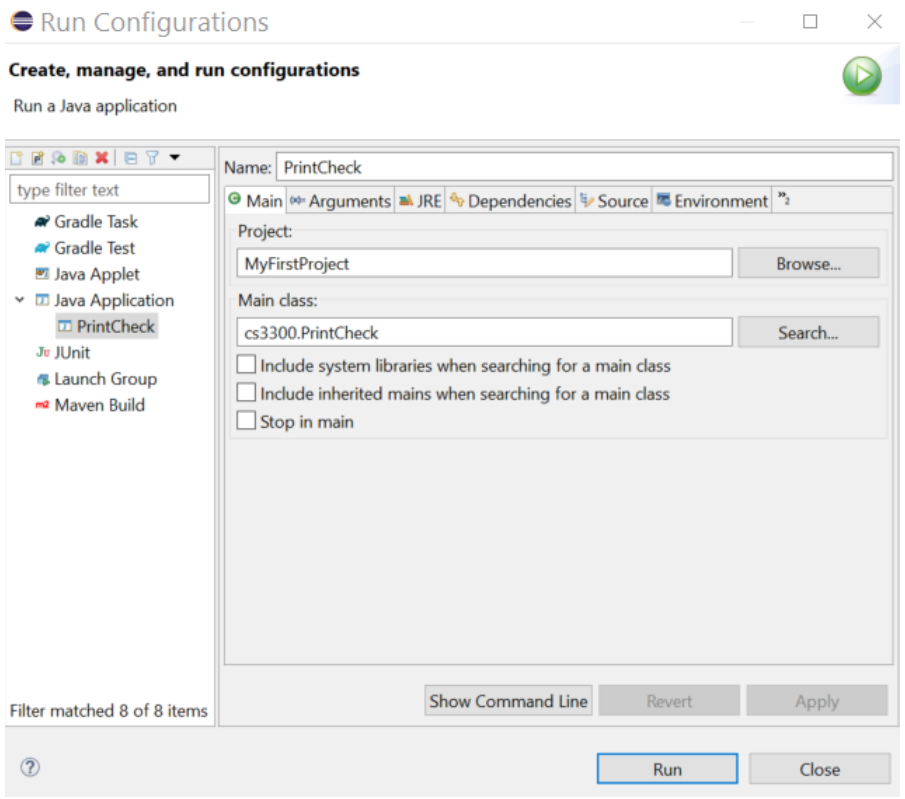
How to create package and Class



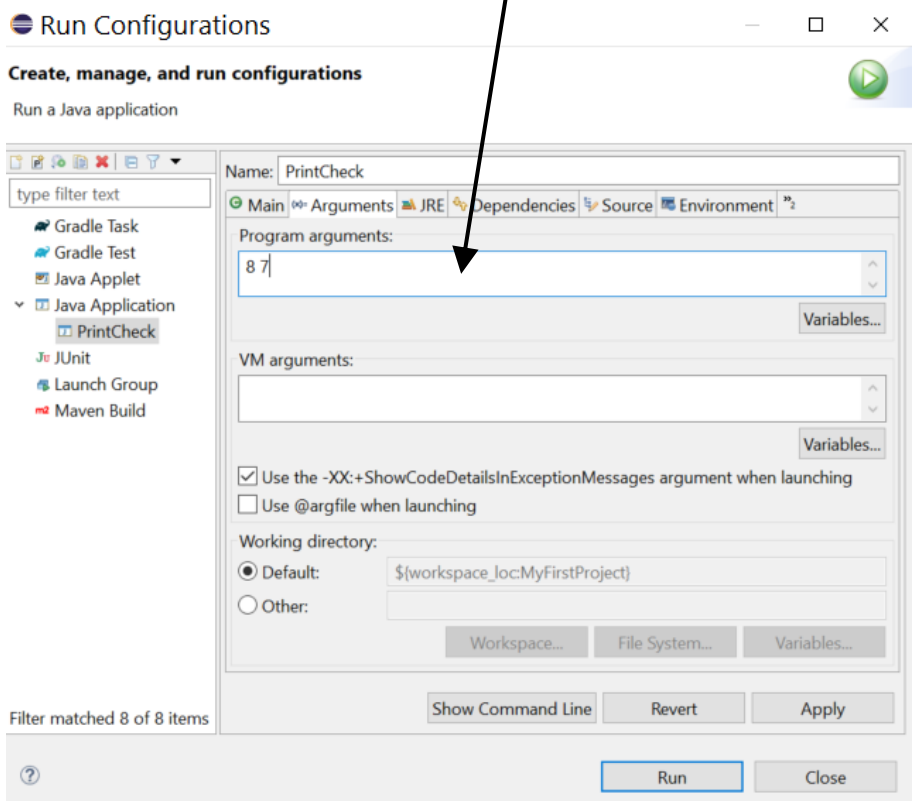


Eclipse

Run Configurations



Pass arguments to function in class



Debug

Switch to debug perspective. Add breakpoints, monitor value of variables

Junit Testing

- Lightweight framework for creating repeatable tests for your application
- Unit testing in Java
  - Imposes developers' discipline
  - Provides incremental specification
  - Avoids regression errors
  - Allows for changing with confidence
- Very helpful in Test driven development

DEMO TIME: Create a Junit test in Eclipse

- How to setup Junit Test case
- How to run a simple Junit test Case
- How to run a Junit test Suite

Maven

- Powerful build management tool that can be used for building and managing any Java based project.
  - Easy building of project
  - Generate source code, generate documentation from source code (log document, dependency list, unit test reports etc)
  - Getting right dependencies for project, Compiling source code
  - Packaging compiled codes into JAR, WAR files without scripting
  - Installing packaged code in local, server or central repositories
- Based on POM (Project Object Model)
  - POM files are XML files that contain information related to the project and configuration information such as dependencies, source directory, plugin, goals etc. used by Maven to build the project.

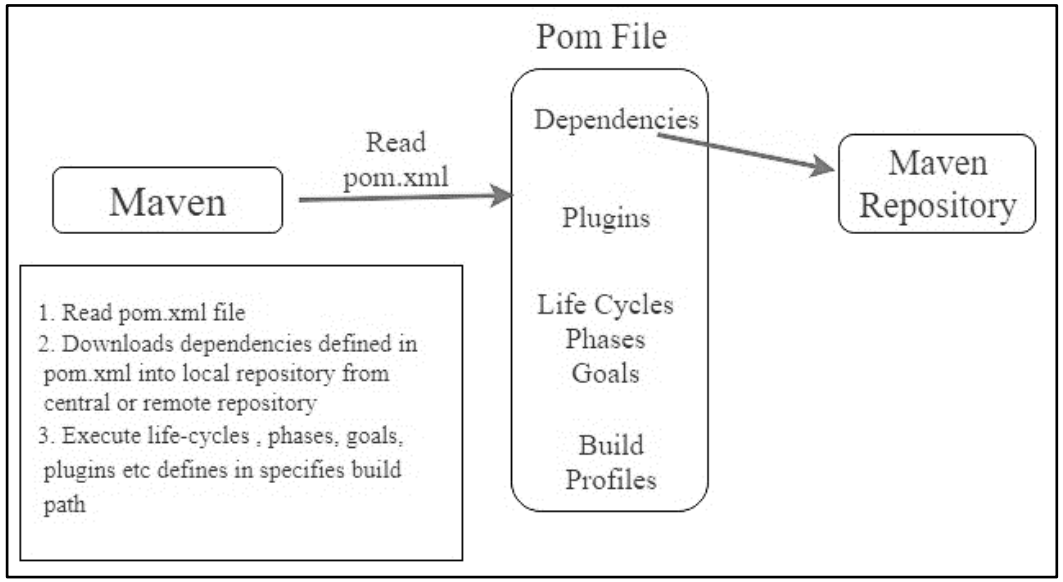


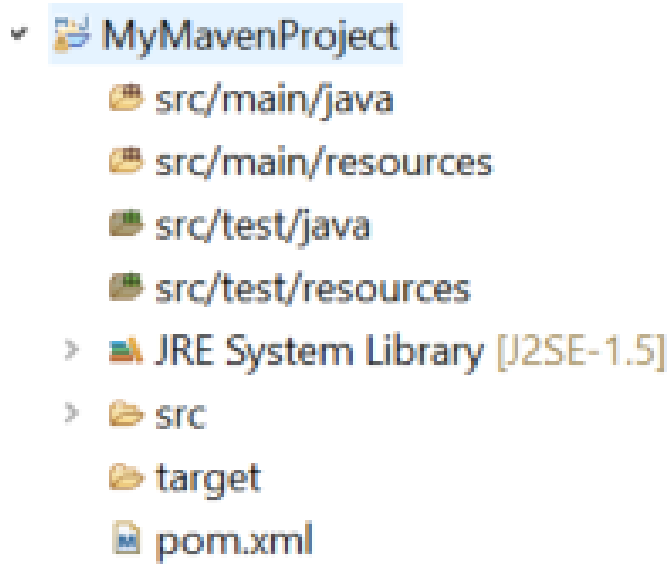
Image courtesy: https://www.geeksforgeeks.org/introduction-apache-maven-build-automation-tool-java-projects/

Maven

DEMO TIME: Create Maven Project using Eclipse IDE

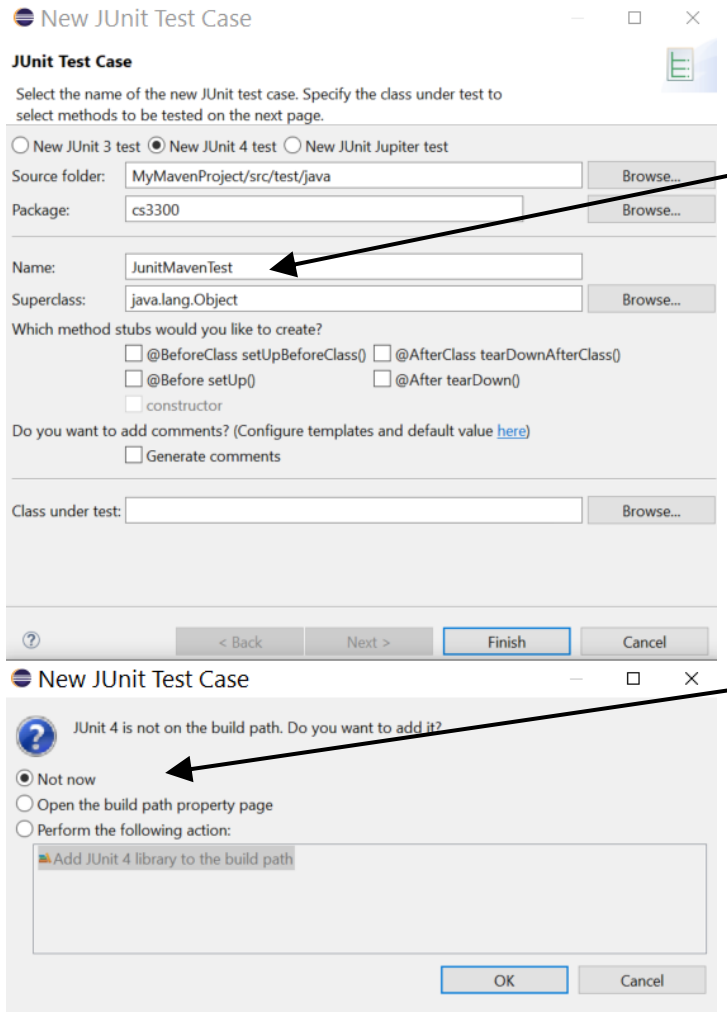
- How to use Maven to create a project
- How to edit pom.xml to add dependencies

Create a Project



Maven

Edit pom.xml to add Junit to build path



Create a Junit test class

Do not add Junit 4 to build path

Add Junit dependency in pom.xml

Takeaway

- Backend vs. Frontend development
- Backend Softwares
- Object Oriented Programming, IDE
- How to write basic Java programs on Eclipse, Junit testing, Use build management tools like Maven

Next Class

- Version Control Systems/ GIT
- Create a Github Student Account: student account instructions
- Download and Install GIT: Follow instructions on https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- Bring your Laptops !!