

CS315 - October 28, 2015

Eclipse, Java, ANT, JavaDoc Lab

Java

- Cross platform
- Object Oriented
- Architecture neutral
- Portable
- Editions
 - *Java Standard Edition (J2SE)*
 - This is the one we're going to use.
 - Java Enterprise Edition (J2EE)
 - Java Micro Editions (J2ME)
- Start with a min call.
- The main method signature should look like this:

```
public class Example
{
    public static void main(String[] args)
    {
        System.out.println("Hello, world");
    }
}
```

- Classes can be marked
 - Abstract
 - Normal
- Attributes can be
 - Public (visible to everyone)

- Private (visible only to the current object)
- Protected (visible to current object and inheriting classes)
- Static (class level variables)

Eclipse

- Used in the industry
- Is a professional Integration Development Environment
- Can be used with different languages, specialized for java
- Utilizes a plug in system.
- Uses contextual views
 - Java view
 - Debug view
 - Git perspective
 - etc

Eclipse can be used to - Create classes and methods (normal and abstract) - Run applications (building/cleaning) - Debug (breakpoints, step in, over, out) - View the Watch List

ANT

- Another neat tool
- Similar to Make, which is used to automate builds
- Maintained by Apache
- Provides instructions on how to create, assemble and or deploy a project
- Written in XML
- Has tons of documentation and is very widely used.

The goal of ANT is **automation**.

If multiple builds are desired, you can create different versions, have different architecture targets and switch between and release builds.

JAVADOC

- JAVADOC is documentation provided in line for classes methods and fields.
- Notation is of the form:

```
/**
 * My class description
 *
 * @author Jacob Zarobsky
 * @version 1.0
 */
public class Example
{

}
```

Can be used to create HTML documentation for code.

Different parameters

- `@author` the author
- `@return` describes what is returned
- `@param [parameter name]` describes a parameter