

# Lesson 1.01 — Using Eclipse

*N.B. THIS LESSON IS OPTIONAL*

## Overview

**Objectives** — *Students will be able to...*

- **Describe** one or more careers related to computer science and technology.
- **Ask** intelligent questions about the field of computer science.
- **Identify** ‘next steps’ to learn more about computer science
- **List** the class expectations and what is required of them.

**Assessments** — *Students will...*

- **Demonstrate** Plug-In and Un-Plug procedures

**Homework** — *Students will...*

- **Visit** <http://www.pokemon.com> and play a few games or **play** Pokémon on a gaming system
- **Visit** <http://bulbapedia.bulbagarden.net> to familiarize yourself with the Pokémon franchise

## Materials & Prep

- **Projector and computer**
- **Student Computers** with Eclipse installed
- **Classroom copies** of WS 1.1.1 and WS 1.1.2
- **USB Drives or Network Drives** for each student (if applicable to school)
- **Handout/Slide Deck** demonstrating file submission procedure

If your school does not have a designated IT specialist available to install Eclipse on the classroom computers, **WS 1.1.1** offers step-by-step installation directions, with screenshots. If your students have computers at home, or if your school loans them laptops for homework and labs, you can print out copies of WS 1.1.1 for your students so they can install Eclipse on their home computers.

If your school distributes USB drives, model best practices by **wearing your USB drive** on your ID lanyard or keychain. Students will lose and/or forget their drives unless they are attached to another object they use daily!

Since all instructors have different preferences and requirements for file uploads/sharing, we have not included a procedure for file submission. You should prepare a handout or slide deck **demonstrating your procedure for submitting work**, and have students send you a sample file to assess understanding of your procedure. 10 minutes of this lesson have been reserved for you to teach these procedures.

---

**Emphasize with students...**

**Big Ideas - Tools and technologies can be adapted for specific purposes** As you begin your journey in this course, think carefully about the tools you use to program. A carpenter, a bricklayer, a painter... all of these people need tools in order to create and build. In much the same way, a computer programmer requires tools to create and build programs.

The Eclipse IDE is a tool that programmers can use to create computer software. Just like how a saw or a hammer or a paint brush can be used to create a wide variety of things, the Eclipse IDE can be used to create a wide variety of software applications.

---

## Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Classroom Introduction <i>Icebreakers &amp; Background</i>	20min
Plug-In & Un-Plug Procedure <i>Demonstration and practice</i>	10min
File Submission Procedure <i>Demonstration and practice</i>	10min

## Procedure

### Bell-work and Attendance [5 minutes]

### Classroom Introduction [20 minutes]

- Using the slide deck as a base (edit the deck to fit your needs):
  - Poll your class to learn their names, experience, and rationale behind taking the course.
  - Go over background information of the computer science field.
  - Go over class expectations and information.
  - Touch on school requirements (varies) *e.g.* syllabus.
- Class icebreakers (varies school to school).

### Plug-In & Un-Plug Procedure [10 minutes]

---

### Emphasize with students...

### Content - Collaboration Tools for programming

- Using WS 1.1.2, model the steps for:
  - Opening Eclipse and a workspace
  - Creating and saving a program
  - Ejecting the USB (if applicable)
- Have students demonstrate the Plug In and Un-Plug procedures for you.

If your classroom has a projector hooked up to the teacher's computer, project each step as you model it for the students.

Can you think of other ways that you might collaborate with others in this course as you create and share programs? What methods are used in the field of programming when there are several of programmers working on developing the same software application?

Wait until all students have completed a step before moving on to another step.

Expect this exercise to take 10 minutes or longer.

---



---

### Emphasize with students...

**Big Ideas - Personal design interests require the evaluation and refinement of skills** Throughout this course you will create a wide range of programs, and you will want to extend projects and add your own creative touch to each one. As you do this, it's important to carefully evaluate and build on the skills that you have. Resources can help you develop further as a programmer, which will allow you to create even more engaging software applications.

---

## **File Submission Procedure [10 minutes]**

1. Demonstrate your procedure for file submission. Use a handout or slide deck to illustrate this procedure.
2. Have students send a sample file to assess understanding of the procedure.

## **Accommodation and Differentiation**

Allow students to work in pairs if they are having trouble understanding the directions. Encourage pairs to model the correct procedure for each other. Student helpers should point to areas on the screen rather than typing or using the mouse to complete the action.

## **Teacher Prior CS Knowledge**

IDEs (Integrated Development Environments) are widely used in industry to develop software. Prior to IDEs, software engineers used simple text editors to write programs. While teachers are welcome to choose from a wide variety of IDEs (see <https://jaxenter.com/the-top-java-ides-114599.html> for a comparison) or even use text editors, TEALS supports Eclipse in the AP CS A curriculum. Whichever IDE/editor you choose, it's a good idea to run through a few IDE tutorials to familiarize yourself with the workflow before demonstrating to the class.

## **Teaching Tips**

- Tips for Volunteers: <http://csteachingtips.org/Tips-for-classroom-volunteers>
- Tips for Reducing Bias: <http://csteachingtips.org/tips-for-reducing-bias>
- Class Introduction: Students will end up in your computer science class for a variety of reasons, here a few that are common:
  - My math teacher recommended I take computer science
  - My friend is in the class
  - I like computers
  - It fit in my schedule
  - It showed up in my schedule
  - It's an AP class and would look good on my transcript
  - I build apps in my spare time
- By knowing the student's motivation, you can better understand where the student is coming from. I use this information to inform the pace and depth of the initial lessons. I also try to determine the level of programming experience of each student. This helps with group formation in the beginning lessons where you try to pair experienced students with less experienced students so the students can learn from their peers.

## **Misconceptions**

When introducing the Java programming language, many students will think Java is synonymous with JavaScript. Although both are programming languages and many of the constructs are transferable between the two languages, Java and JavaScript are two different languages. JavaScript is used today for both client side web browser scripting to web servers. Java is used to create platform independent apps meaning a program written in Java will run across a variety of operating systems.

## **Forum discussion**

Lesson 1.01 Using Eclipse (TEALS Discourse account required)