Week 6 – Topics

- Debugging
- Unit Tests

Debugging

Debugging

Debugging can be described as diagnosing problems in code: finding what is causing a problem, and fixing those problems!

A bug in a program is something that causes the program to fail to do what it is designed to do.

The term debugging comes from taking bugs out of code, literally removing the bugs from a program.

The term debug is anecdotally attributed to U.S. Navy Rear Admiral Grace Hopper. The story is told that in 1945, she coined the term debug when she and her colleagues found a problem in Mark II computer, when they opened up the machine, they found and removed an actual moth. Although the term bug had been used since the 19th century to describe a mechanical malfunction, Grace Hopper was actually the first to use bug to talk about a computer problem, and to talk about debugging a computer.

https://github.com/ckiefriter1/Java-Code-Examples/blob/main/src/debugging/DebuggingExample.java

Using Debuggers like in Eclipse

Important vocabulary:
debugger: a computer program that helps find and correct errors in other computer programs.
breakpoint: a signal that informs the debugger to temporarily suspend execution of the program at a certain point in the code.

Integrated Development Environments (IDE) like Eclipse provide debuggers.

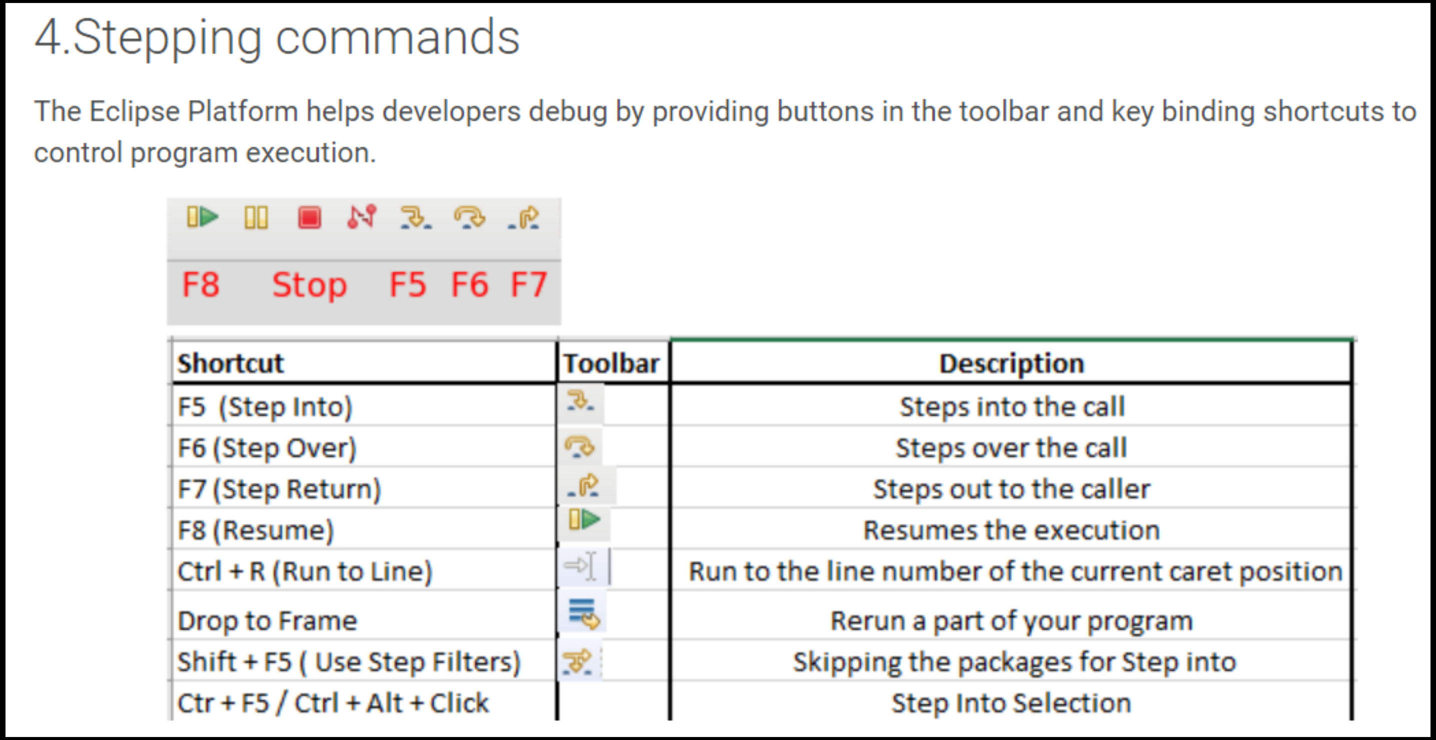
The following instructions will aid in the use of the Eclipse IDE debugger:

To define one or more breakpoints in your source code where you suspect the problem is happening. To set a breakpoint, right-click in the left margin when editing your code in Eclipse IDE, and select Toggle Breakpoint.

To run your Java code in debug-mode within the Eclipse IDE, choose:
Run --> Debug As --> Java Application

The Debug Perspective offers new views to allow better troubleshooting, including Debug View, Breakpoints View, Variables View, Expressions View, and more.
Definitely switch to the perspective to the Debug Perspective to take advantage of these different views.

Using Debuggers like in Eclipse



4.Stepping commands

The Eclipse Platform helps developers debug by providing buttons in the toolbar and key binding shortcuts to control program execution.

Shortcut	Toolbar	Description
F5 (Step Into)	↳	Steps into the call
F6 (Step Over)	↷	Steps over the call
F7 (Step Return)	↱	Steps out to the caller
F8 (Resume)	▶	Resumes the execution
Ctrl + R (Run to Line)	→	Run to the line number of the current caret position
Drop to Frame	≡	Rerun a part of your program
Shift + F5 (Use Step Filters)	↳	Skipping the packages for Step into
Ctr + F5 / Ctrl + Alt + Click		Step Into Selection

Unit Tests

Unit Tests

Unit Testing is code you write that runs your application code to make sure the output is as expected.

Junit is a common framework used to build unit tests for code.

https://junit.org/junit5/docs/current/user-guide/#overview

https://github.com/ckiefriter1/Java-Code-Examples/tree/main/src/UnitTesting

Appendix