1. Learn all the primitive number types.

2. Learn how to concatenate strings.

3. Learn arithmetic operators and assignment operators.

4. Read Chapter 1 of Think Java

5. Spend 30 minutes trying to install Eclipse. Don't waste too much time if stuck.

**Vocab**

reserved type

string literal

primitive

Integer: 3, 0, -32

Floating point number – 4.567 x 10^13, 4.223 x 10^-3

! - bang

( - open parenthesis

) - closed parenthesis

{} - curly braces

[] - square braces

; - semicolon

/ - forward slash; division

\ - backslash; escape character

\n - newline! - bang

( - open parenthesis

) - closed parenthesis

{} - curly braces

[] - square braces

; - semicolon

/ - forward slash; division

\ - backslash; escape character

\n – newline

**problem-solving:**

The process of formulating a problem, finding a solution, and expressing the solution.

**high-level language:**

A programming language like **Java** that is designed to be easy for humans to read and write. **Scripting languages (python, basic) are also high-level.**

**low-level language:**

A programming language that is designed to be easy for a computer to run. Also called “machine language” or “assembly language.”

**formal language:**

Any of the languages people have designed for **specific purposes**, like representing mathematical ideas or computer programs. All **(current)** programming languages are formal languages.

**natural language:**

Any of the languages **people speak** that have evolved naturally.

**portability:**

**Can run on more that one OS.** A property of a program that can run on **more than one kind of computer**.

**interpret:**

To run a program in a high-level language by translating it one line at a time.

**compile:**

To translate a program in a high-level language into a low-level language, all at once, in preparation for later execution. **Usually into machine code.**

**source code:**

A program in **a high-level language**, before being compiled.

**object code:**

The output of the **Java compiler**, after translating the program.

**executable:**

Another name for **object code that is ready to run**. **Programming languages like C will create a .exe in windows.**

**byte code:**

A **special kind** of object code used for **Java programs**. Byte code is similar to **machine code**, but it is **portable**, like a high-level language.

**statement:**

**One line of code or one command.** A part of a program that specifies a computation.

**print statement:**

A statement that causes **output** to be displayed on the screen. **System.out.print(); or System.out.println();**

**comment:**

A part of a program that contains information about the program, but that has no effect when the program runs.

**End of line comment //**

**Multi-line comment /\* \*/**

**method:**

A named collection of statements. **A function attached to an object.**

**library:**

A collection of **class and method definitions. Ex. System Math**

**bug:**

An **error** in a program.

**syntax:**

The **structure** of a program.

**semantics:**

The **meaning** of a program.

**parse:**

**For the computer to read and find the “meaning” of your software.** To examine a program and analyze the syntactic structure.

**syntax error:**

An error in a program that makes it **hard/impossible to parse** (and therefore impossible to compile).

**exception:**

An error in a program that makes it fail at run-time. Also called a run-time error.

**logic error:**

An error in a program that makes it do something other than what the programmer intended.

**debugging:**

The process of finding and removing any of the three kinds of errors.