Java Notes Chapter 02 – Using Data

Variables: Lower Camel Casing 🡪 string myName;

Methods: Lower Camel Casing 🡪 nextLine();

Objects: Lower Camel Casing 🡪 myHonda

Classes: Upper Camel Casing 🡪 public class MyFirstClass()

Named Constants: ALL CAPS with underscores separating words 🡪 final string MY\_NAME;

**CONSTANTS & VARIABLES:**

* Variable Declaration:
* Syntax: data type, identifier, & optional assigned value
* Assign a value to the variable using ‘=’
* Constant Declaration:
* Syntax: keyword ‘final’, data type, identifier, & assigned value
* Data Types: 8 primitive
* byte, short, int, long, float, double, char, Boolean.

\*\*Concatenating Strings with numbers makes the entire expression becomes a string

* Each primitive type has a corresponding **Wrapper Class** in the java.lang
* Can be used to convert string to other primitive types

**INTEGER:**

* Whole number w/out decimal
* int: no commas or periods; can use ‘\_’ in place of commas
* byte (byte-length), short, long (follow number with ‘L’)

**BOOLEAN:**

* Holds True or False
* Uses relational operators to compare two things

**FLOATING-POINT:**

* Contains decimal positions
* Float: can hold 6-7 digits; follow number with ‘F’
* Double: can hold 14-15 digits; requires more memory; default

\*\*Imprecision in floating-point numbers 🡪 Use techniques to round values

**CHAR:**

* Any single character; any letter, punctuation mark, or digit
* Place w/in single quotations
* Stores nonprinting characters (backspace, tab) using escape sequences

\*\*A character that’s a digit is represented in computer memory differently than a numeric value represented by the same digit

**STRING:** Built in class for storing & manipulating character strings

**Scanner Class:** used to accept user input

* System.in property: Standard input device (usually keyboard)
* Used to create interactive programs that accept input from user
* Not flexible, can only read bytes
* **Scanner Class:** makes System.in flexible
* Syntax:

Import java.util.Scanner; /\*import statement

Scanner inputDevice /\*declares object of type scanner & names it

= new Scanner(System.in); /\*creates scanner object & connects it to System.in

* Contains methods that retrieve values (tokens) from an input device

-next() nextInt() nextDouble() nextLine()

\*\*After any next(), nextInt, or nextDouble() call, add extra nextLine() call

**JOptionPane Class:** used to accept GUI input - 2 dialog boxes

* Syntax:
* Import javax.swing.JOptionPane; /\*import statement
* Input Dialog: prompts user for text input
* Uses the showInputDialog() method that returns string
* Four arguments:
* Parent component, prompt, title, icon
* Confirm Dialog: displays Yes, No, and Cancel buttons for answer
* Uses the showConfirmDialog() method that returns an integer
* Five arguments:
* Parent component, prompt, title, which option button(s) shown, icon

**ARITHMETIC:**

* Java supports two types of Division:
* Floating point division: 45.0 / 2 = 22.5
* Integer division: 45 / 2 = 22
* Remainder Operator: (%) 39 % 5 = 4
* Don’t need to perform a division operation first
* Operator Precedence:
* /, % 🡪 Higher
* +, -- 🡪 Lower
* Left to Right

**TYPE CONVERSION:**

* Performing mathematical operations on operands w/ unlike types
* Java chooses a unifying type based on:
* Short & byte automatically🡪 int🡪 long🡪 float🡪 double
* Purposefully override the unifying type imposed by Java using a type cast