Chapter 2

Using Objects

Somaiyah Sultani, 13-12-2022

Variables

- Place holders for some values.
- The values for a variable can be changed.
  -

- … greeting = “Hello World”;
- System.out.println(“Hello World”);
  - prints out Hello World in the console
- System.out.println(greeting);
  - It will also print out Hello World in the console .
- … greeting = “Hi everyone”;
- System.out.println(greeting);
  - It will also print out Hi everyone in the console .

- Naming conventions:
    1. It can contain any letter or number or underscore symbol _
    2. The variable always starts with a small letter; ex. container, log, circle.
    3. If it is made up of multiple words without any underscore separating them, the first letter of each consecutive words will be capital.
        1. ex. my_account
        2. ex. myAccount
    4. It should descriptive of the value it is representing/referencing
        1. … greeting = “Hello World”; (OK)
        2. … x = “Hello World”; (NOT OK)

variables conti. (Syntax Errors)

- Cannot use any Java Keywords for naming variable
  - ex. … class = …; Error
  - ex … public = …; Error
- The variable names cannot start with a number
  - ex, … 1stAccount = …; (Error)
  - ex, firstAccount, account1

List of Java Keywords

Types/Data Types

- It specifies the type of data the variable is referring to.
- It is used by the compiler to allocate memory for each variable
    - if the type of the variable is different from what it is referring to we will get a compilation error

Primitive Data Type

- it is the simplest form of data
-

String

- A list of characters define in ASCII form
- The size of a string depends on the number of characters included in that String
- Difference between a String and other primitive data types:
    1. all keywords for primitive data types start with a small letter. However, String start with capital S. (all complex data types start with capital letter)
        1. ex. int, float double vs. String
    2. The size of the memory allocated for primitive types are fixed (we always know what they are) but for String it varies based on the value of the variable

String

- It is a complex data type which is treated sometimes like a primitive type and other times it is a complex type.
- to define a String:
  - String variableName = variableValue
    - ex. String greeting = “Hello World!”;
  - literal form of a String is enclosed in double quotation,
    - ex, “Hello World!”, “This is a String”, “1- This is the first line. \n2- This is the second line.” , “Hi”, “x”,
  - Anything inside the double quotation marks will be represented as is. The double quotation marks are NOT part of the String.
    - ex. “HELLO World!” is different than “hello world” and they represent HELLO World! and hello world respectively.

Declaration

- it declares the type of the variable ex. int height;
- syntax form
  - dataType variableName;
- Each variable is declared ONLY once.
  - compilation error
    - if we included the dataType for a variable when calling the variable again, we will have an error: Duplicate local variable variableName

Assignment Operator

- is the = and it is used to assign a value to a variable
  - ex. weight = 45.3;
- Syntax form:
  - variableName assignmentOperator value;
  - variableName = value;
- if we are giving the variable a value for the first time, we call it initialization
- if we are changing the value of a variable to assigning a new value we call just call it assignment

Initialization

- It initializes the value of a variable; it assigns a value to a variable for the first time.

Create a variable

1. do it is two step
    1. Declare a variable, int weight;
    2. initialize the variable, weight = 672.8;
2. do it in one step
    1. declare and initialize at the same time; int weight = 672.8;

- General Syntax
    - first form
        - dataType variableName;
        - variableName = value;
    - second form
        - dataType variableName = value;