What is programming?

- The term programming means to create (or develop) software, which is also called a program. In basic terms, software contains the instructions that tell a computer or a computerized device what to do.

Types of software

Airplanes, cars, cell phones, Web browsers and e-mail.

Programming Languages

Computer programs, known as software, are instructions that tell a computer what to do. Computers do not understand human languages, so programs must be written in a language a computer can use. There are hundreds of programming languages, and they were developed to make the programming process easier for people. However, all programs must be converted into the instructions the computer can execute.

Types Programming Languages

- Machine Language

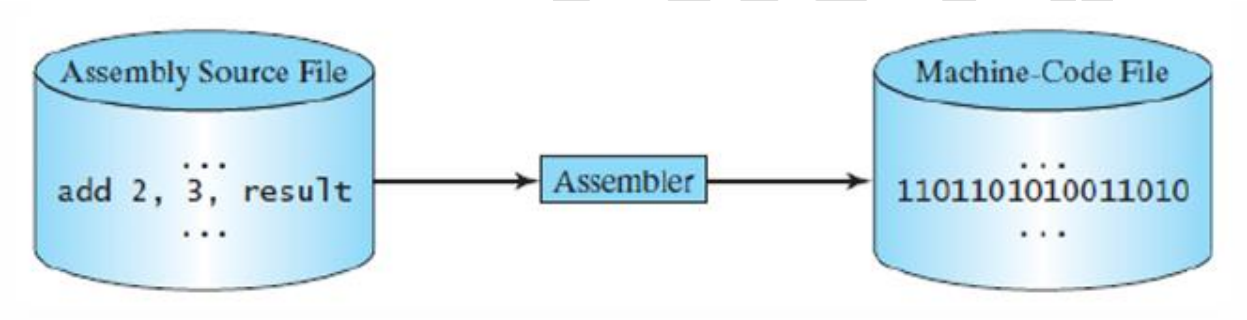
  A computer's native language, which differs among different types of computers, is its machine language a set of built in primitive instructions. These instructions are in the form of binary code, so if you want to give a computer an instruction in its native language, you have to enter the instruction as binary code. For example, to add two numbers, you might have to write an instruction in binary code, like this:

  1101011100111011

- Assembly Language

  programs written in machine language are very difficult to read and modify. For this reason, assembly language was created in the early days of computing as an alternative to machine languages. For example, the mnemonic add typically means to add numbers and sub means to subtract numbers. To add the numbers 2 and 3 and get the result, you might write an instruction in assembly code like this:

  add 2, 3, result



- High-Level Language
  - They are platform independent, which means that you can write a program in a high-level language and run it in different types of machines. High-level languages are English-like and easy to learn and use. The instructions in a high level programming language are called statements. Here, for example, is a high level language statement that computes the area of a circle with a radius of 5:

    area = 5 * 5 * 3.14159;

  - There are many high-level programming languages, and each was designed for • Basic, C, C++, C#, COBOL, Fortran, Java, Pascal, Python, Vistula Basic.

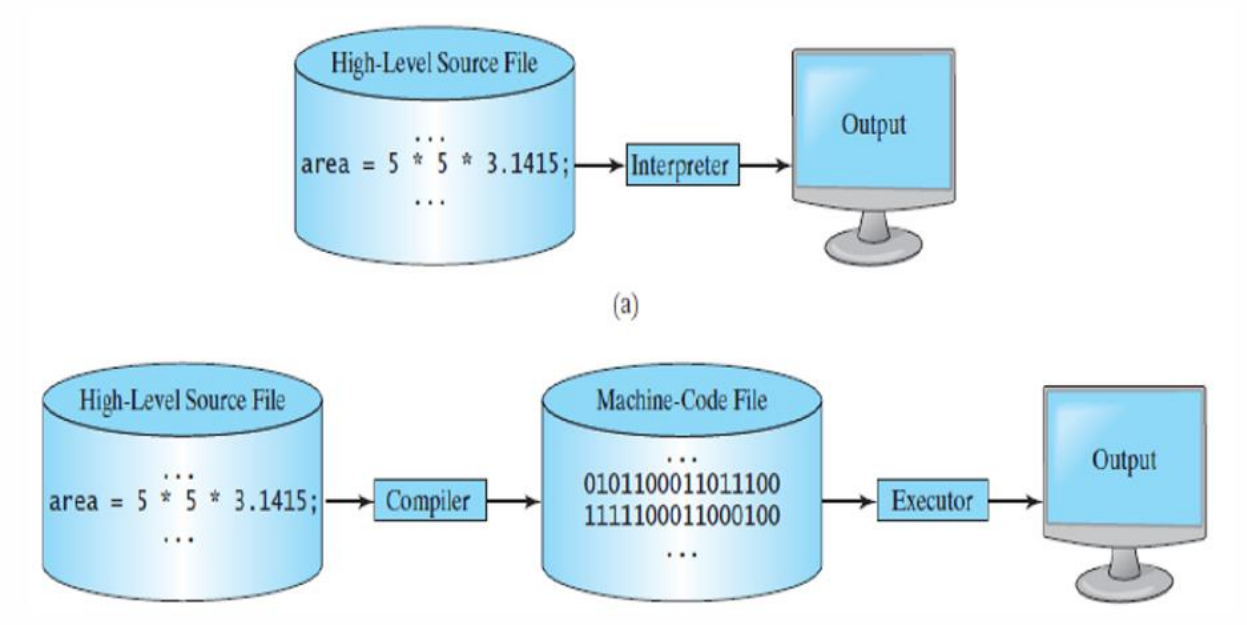
- A program written in a high-level language is called a source program or source code. Because a computer cannot execute a source program, a source program must be translated into machine code for execution. The translation can be done using another programming tool called an **interpreter** or **a compiler**.

An interpreter

Reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in Figure below. Note that a statement from the source code may be translated into several machine instructions.

A compiler

Translates the entire source code into a machine-code file, and the machine-code file is then executed.



Java, the World Wide Web, and Beyond

- First time Java was developed by a team led by James Gosling at Sun Microsystems.
- Sun Microsystems was purchased by Oracle in 2010. Originally called Oak
- 1995 renamed java it was redesigned for developing Web applications.

Java characteristics

Simple, object oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high performance, multithreaded, and dynamic.

The Java Language Specification, API, JDK, and IDE

Computer languages have strict rules of usage. If you do not follow the rules when writing a program, the computer will not be able to understand it. The Java language specification and the Java API define the Java standards.

- The Java language specification is a technical definition of the Java programming language's syntax and semantics.
- The application program interface (API), also known as library, contains predefined classes and interfaces for developing Java programs. The API is still expanding.

Java is a full-fledged and powerful language that can be used in many ways. It comes in three editions:

- Java Standard Edition (Java SE) to develop client-side applications. The applications can run standalone or as applets running from a Web browser.
- Java Enterprise Edition (Java EE) to develop server-side applications, such as Java servlets, Java Server Pages (JSP), and Java Server Faces (JSF).

- ❖ Java Micro Edition (Java ME) to develop applications for mobile devices, such as cell phones.

Java SE Java Standard Edition

- Java Development Toolkit (JDK)

  Consists of a set of separate programs, each invoked from a command line, for developing and testing Java programs.
- Java development tool

  (e.g. Eclipse, NetBeans and TextPad) software that provides an integrated development environment (IDE) for developing Java programs quickly. Editing, compiling, building, debugging, and online help are integrated in one graphical user interface.

A Simple Java Program

```
1  // Fig. 2.1: Welcome1.java
2  // Text-printing program.
3
4  public class Welcome1
5  {
6     // main method begins execution of Java application
7     public static void main( String[] args )
8     {
9        System.out.println( "Welcome to Java Programming!" );
10    } // end method main
11 } // end class Welcome1
```

```
Welcome to Java Programming!
```

Note that

- ✓ **The line numbers** are for reference purposes only; they are **not part** of the program. So, **do not** type line numbers in your program.
- ✓ **Line 1** defines a **class**. Every Java program must have at least one class. Each class has a name. By convention, class names start with an uppercase letter. In this example, the class name is **Welcome**.
- ✓ **Line 2** defines the **main method**. The program is executed from the **main method**. A class may contain several methods. The **main method** is the entry point where the program begins execution.
- ✓ **A method** is a construct that contains statements. The **main method** in this program contains the **System.out.println** statement. This statement displays the string **Welcome to Java!** on the console (line 4).
- ✓ **String** is a programming term meaning a sequence of characters. A string must be enclosed in double quotation marks.
- ✓ Every statement in Java ends with a semicolon **(;)**, known as the **statement terminator**.
- ✓ **Reserved words**, or **keywords**, have a specific meaning to the compiler and cannot be used for other purposes in the program. For example, when the compiler sees the word **class**, it understands that the word after **class** is the name for the class. Other reserved words in this program are **public, static, and void**.
- ✓ **Line 3** is a **comment** that documents what the program is and how it is constructed. Comments help programmers to communicate and understand the program. They are not programming statements and thus are ignored by the compiler. In Java, comments are preceded by two slashes (//) on a line, called a **line comment**, or enclosed between /* and */ on one or several lines,

called a **block comment or paragraph comment**. When the compiler sees //, it ignores all text after // on the same line. When it sees /*, it scans for the next */ and ignores any text between /* and */. Here are examples of comments:

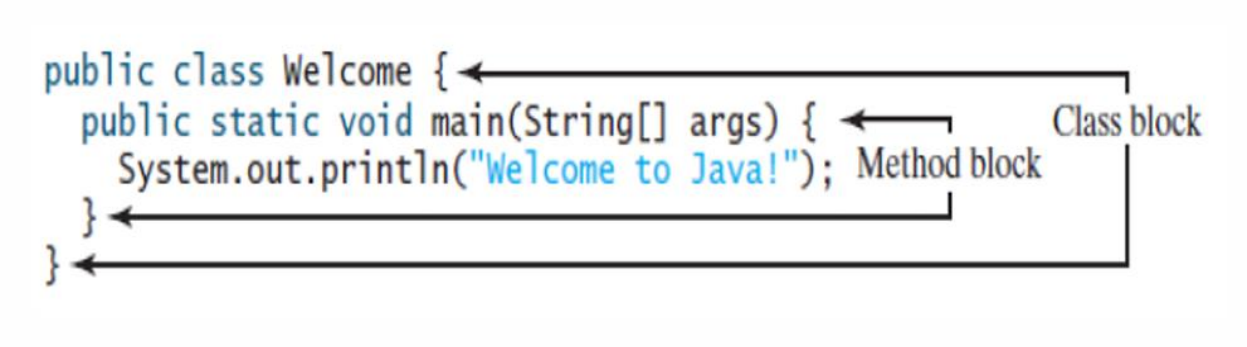
```
// This application program displays Welcome to Java!

/* This application program displays Welcome to Java! */

/* This application program

displays Welcome to Java! */
```

A pair of curly braces in a program forms a **block** that groups the program's components. In Java, each block begins with an opening brace ({) and ends with a closing brace (}). Every class has a **class block** that groups the data and methods of the class. Similarly, every method has a **method block** that groups the statements in the method. Blocks can be **nested**, meaning that one block can be placed within another, as shown in the following code.



Identifier

- An identifier is the name you give to a class, variable, method and so forth in a java program.
- It is a series of characters consisting of letters, digits, underscores (_) and dollar signs ($) that does not begin with a digit and does not contain spaces. Some valid identifiers are Welcome1, $value, _value, m_inputField1.
- By convention, class names begin with a capital letter and capitalize the first letter of each word they include (e.g., SampleClassName).
- Do not use **special characters** such as +, -, *, etc. or **reserve words** such as int, float, main, long, char, etc.

Variable

A variable is the name of a reserved area allocated in memory. In other words, it is a name of the memory location. It is a combination of "vary + able" which means its value can be changed.

Syntax

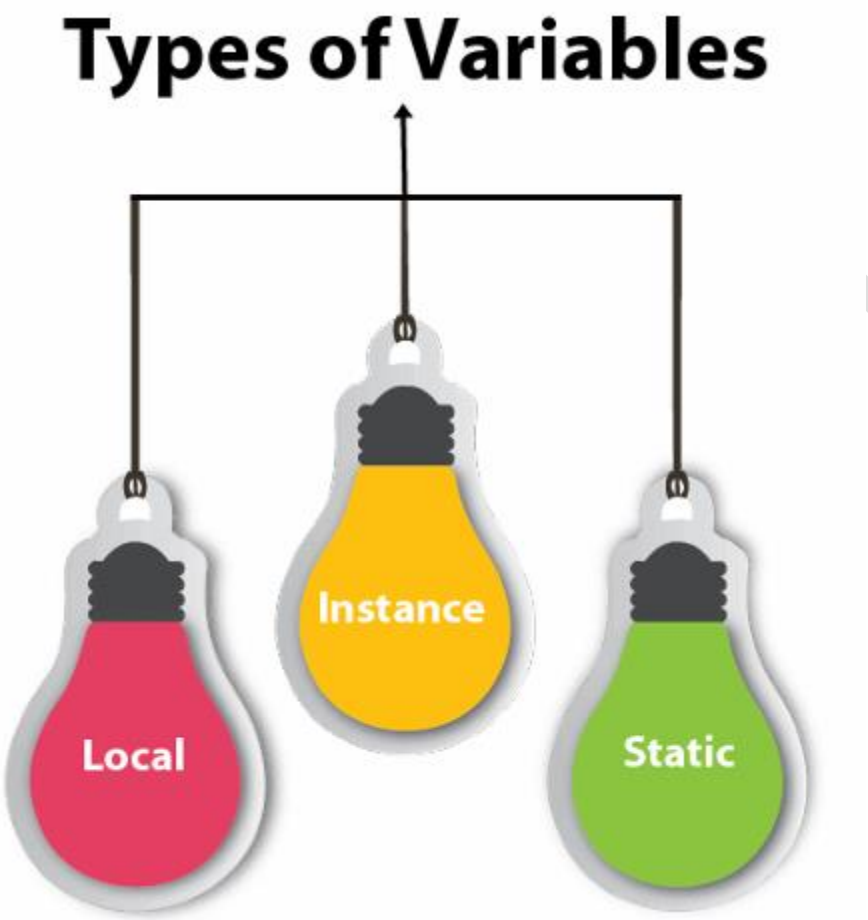
variable datatype variable_name = variable_value;

- A variable is a location in the computer's memory where a value can be stored for use later in a program.
- Every variable must be declared before using it.
- Variables have unique names.
- You can initialize the variable when it declares.
- The type of the variable determines the size of variable that stored in computer memory.

Example:

```
int data=50; //Here data is variable
```

Types of Variables



Local Variable

- A variable declared inside the body of the method is called **local** variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.
- A **local** variable cannot be defined with "static" keyword.

Instance Variable

- A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as **static**.
- It is called an **instance** variable because its value is instance-specific and is not shared among instances.

Static variable

A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

Case sensitive

Java is case sensitive.

Example:

rate RATE Rate

R not like r , a like a

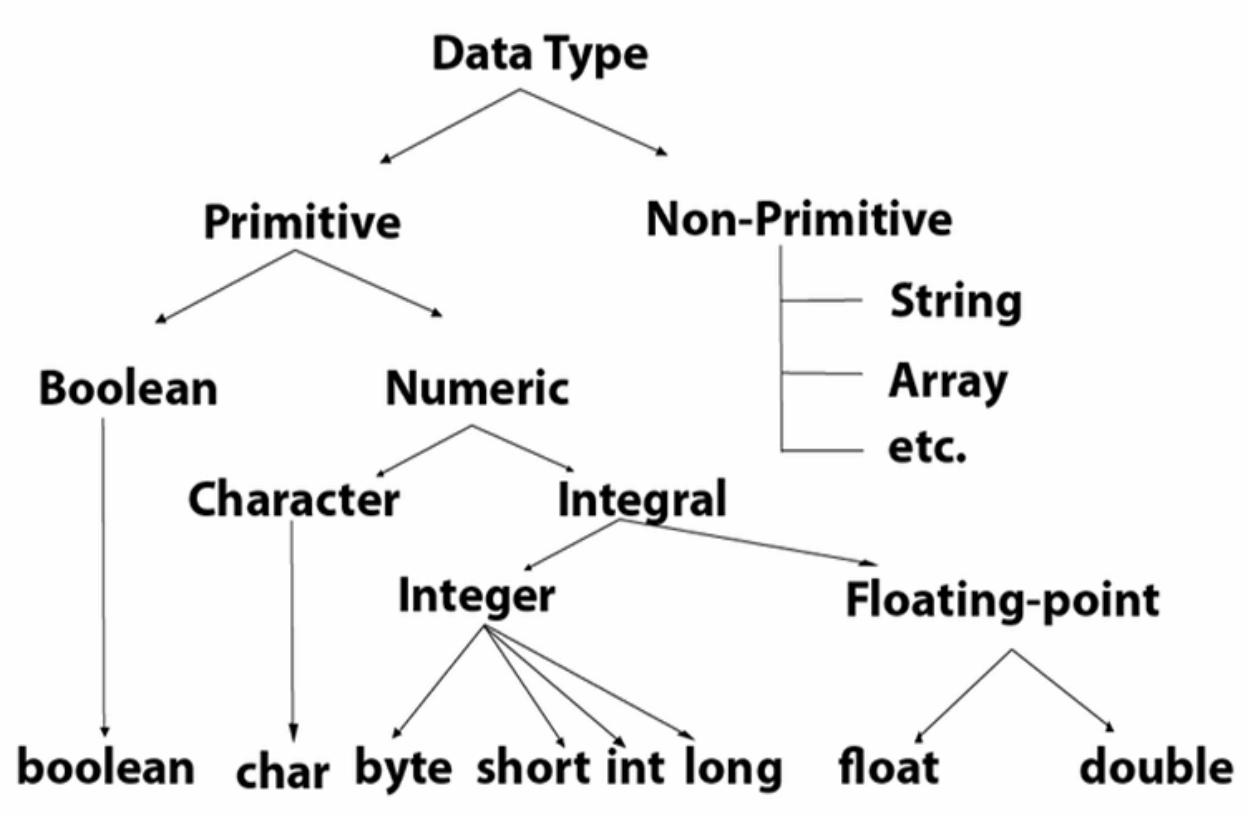
Data Types in Java

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

1. Primitive data types: The primitive data types include Boolean, char, byte, short, int, long, float and double.
2. Non-primitive data types: The non-primitive data types include **Classes**, **Interfaces**, and **Arrays**.

Java Primitive Data Types

- In Java language, primitive data types are the building blocks of data manipulation. These are the most basic data types available in Java language.
- There are 8 types of primitive data types:
- Boolean data type.
- Byte data type.
- Char data type.
- Short data type.
- Int data type.
- Long data type.
- Float data type.
- Double data type.


Data Type
Primitive
Non-Primitive
String
Array
etc.
Boolean
Numeric
Character
Integral
Integer
Floating-point
boolean
char
byte
short
int
long
float
double

Data Type	Default Value	Default size	Minimum	Maximum
Boolean	false	1 bit	____	___
Char	'\u0000'	2 byte	Unicode 0	Unicode 2^{16}- 1
Byte	0	1 byte	-128	+127
short	0	2 byte	-2^{15}	$+2^{15}$-1
int	0	4 byte	-2^{31}	$+2^{31}$-1
long	0L	8 byte	-2^{63}	$+2^{63}$-1
float	0.0f	4 byte	IEEE754	IEEE754
double	0.0d	8 byte	IEEE754	IEEE754

Thank You

د. محمد رضا الكعبي

Mobile: 07801582655

E-mail: mridha.hamoodi@uokufa.edu.iq

" ليس كل سقوط نهاية فسقوط المطر أجمل بداية "