2007 KARABÜK ÜNİVERSİTESİ

CPE207 Object Oriented Programming

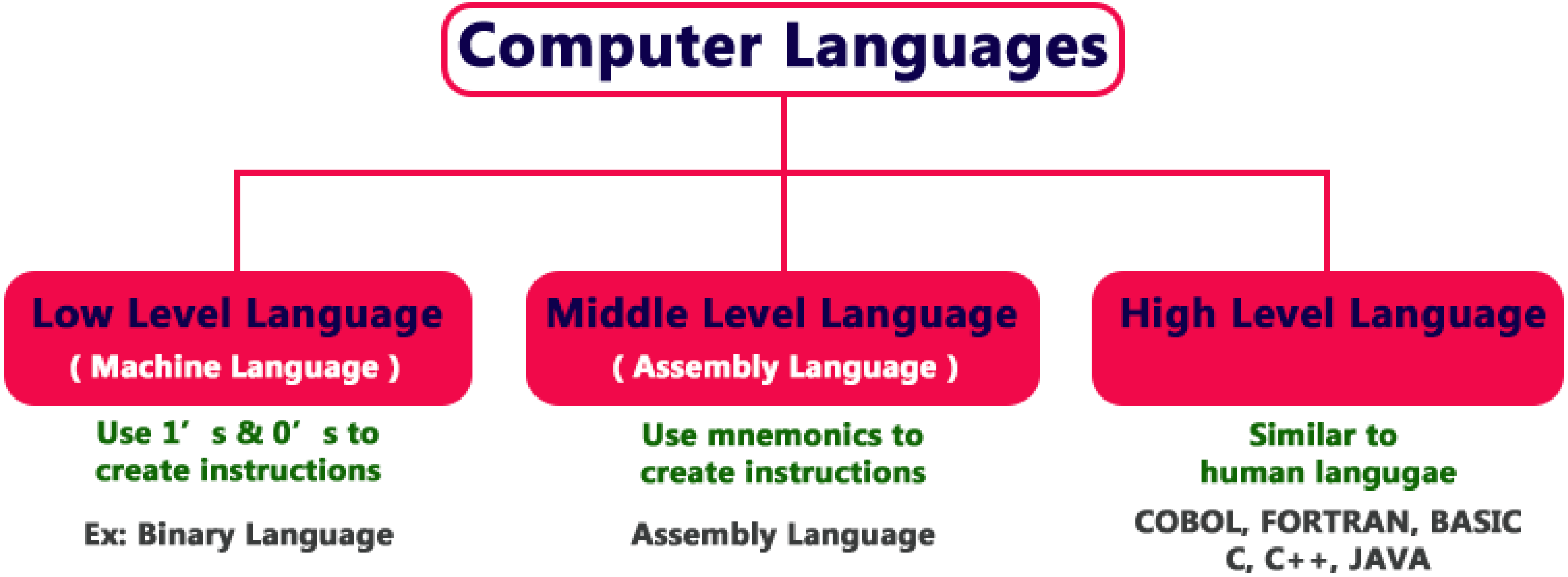
Week 1

Object-Oriented Programming (in Java)

Dr. Nehad Ramaha,
Computer Engineering Department
Karabük Universities

The class notes are a compilation and edition from many sources. The instructor does not claim intellectual property or ownership of the lecture notes.

Levels of Programming Languages



Computer Languages

Low Level Language

(Machine Language)

Use 1’ s & 0’ s to create instructions

Ex: Binary Language

Middle Level Language

(Assembly Language)

Use mnemonics to create instructions

Assembly Language

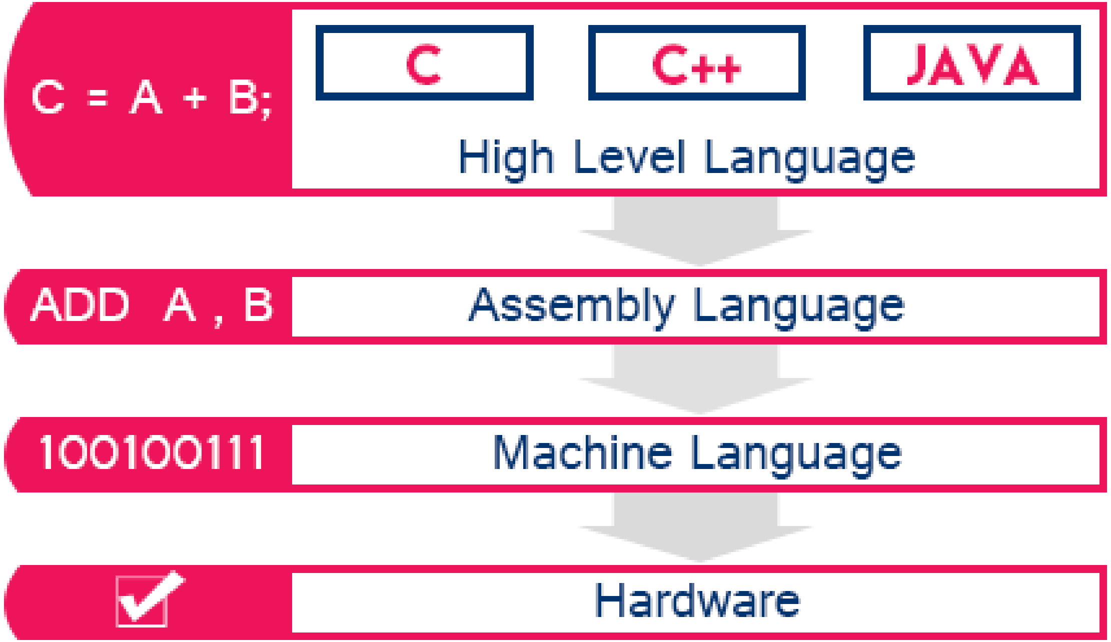
High Level Language

Similar to human langugae

COBOL, FORTRAN, BASIC
C, C++, JAVA

Computers understand only Machine Language

Levels of Programming Languages



C = A + B;

C

C++

JAVA

High Level Language

ADD A , B

Assembly Language

100100111

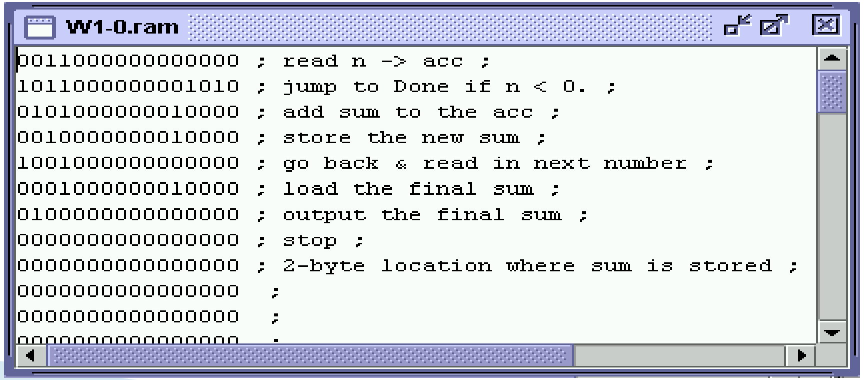
Machine Language

☑

Hardware

Machine Language

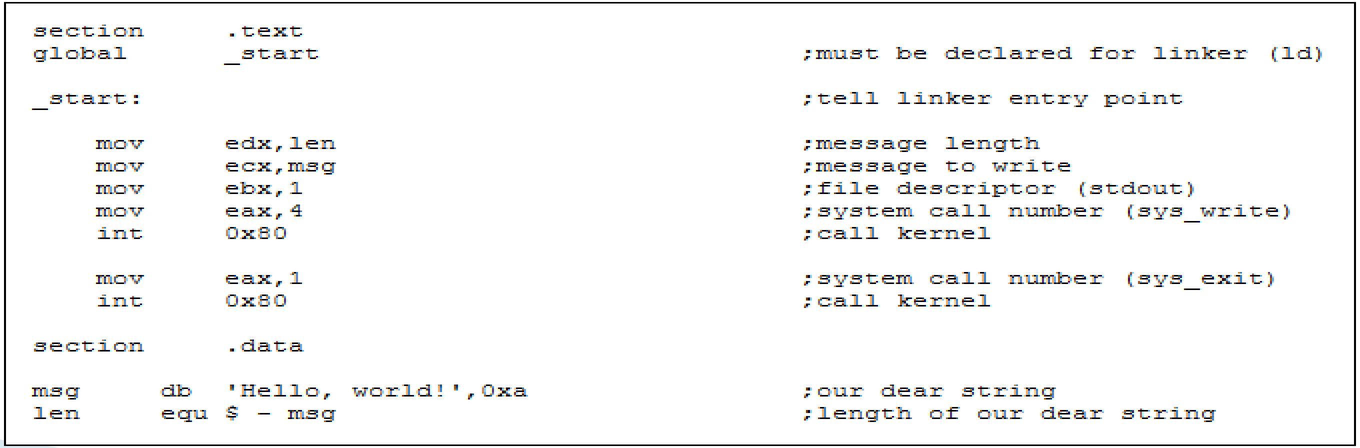
- The fundamental language of the computer's processor, also called Low Level Language.
- All programs are converted into machine language before they can be executed.
- Consists of combination of 0's and 1's that represent high and low electrical voltage.



```
0011000000000000 ; read n -> acc ;
1011000000001010 ; jump to Done if n < 0. ;
0101000000010000 ; add sum to the acc ;
0010000000010000 ; store the new sum ;
1001000000000000 ; go back & read in next number ;
0001000000010000 ; load the final sum ;
0100000000000000 ; output the final sum ;
0000000000000000 ; stop ;
0000000000000000 ; 2-byte location where sum is stored ;
0000000000000000  ;
0000000000000000  ;
0000000000000000  ;
```

Assembly Language

- A low level language that is similar to machine language.
- Uses symbolic operation code to represent the machine operation code.



```
section     .text
global      _start                                          ;must be declared for linker (ld)

_start:                                                     ;tell linker entry point

    mov     edx,len                                         ;message length
    mov     ecx,msg                                         ;message to write
    mov     ebx,1                                           ;file descriptor (stdout)
    mov     eax,4                                           ;system call number (sys_write)
    int     0x80                                            ;call kernel

    mov     eax,1                                           ;system call number (sys_exit)
    int     0x80                                            ;call kernel

section     .data

msg     db  'Hello, world!',0xa                             ;our dear string
len     equ $ - msg                                         ;length of our dear string
```

High level Language

- Computer (programming) languages that are easier to learn.
- Uses English like statements.
- there are hundreds of high level programming languages, following are a few of them:
  - Java
  - C
  - C++
  - Python
  - PHP
  - Perl
  - Ruby

High level Language (2)

- You eventually need to convert your program into machine language so that the computer can understand it.
- There are two ways to do this:
  - Compile the program.
  - Interpret the program.

compiler

- Compile is to transform a program written in a high–level programming language from source code into object code.
- This can be done by using a tool called compiler.
- A compiler reads the whole source code and translates it into a complete machine code program to perform the required tasks which is output as a new file.

Interpreter

- Interpreter is a program that executes instructions written in a high-level language.
- An interpreter reads the source code one instruction or line at a time, converts this line into machine code and executes it.

What is Java?

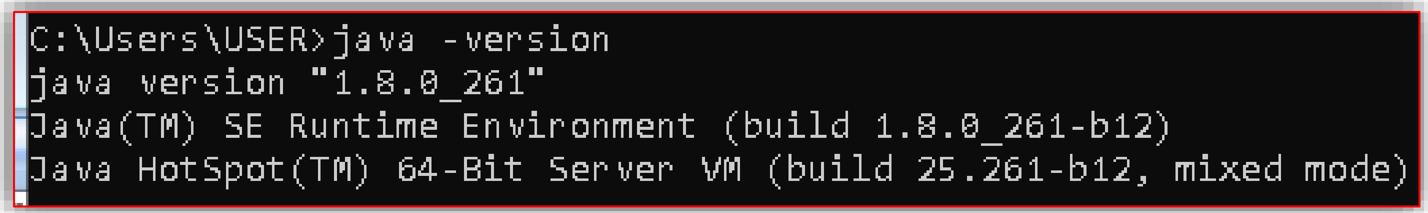
- Java is a popular programming language, created in 1995.
- It is owned by Oracle, and more than 3 billion devices run Java.
- It is used for:
  - Mobile applications (specially Android apps)
  - Desktop applications
  - Web applications
  - Web servers and application servers
  - Games
  - Database connection
  - And much, much more!

Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

Java Install

- To check if you have Java installed on a Windows PC, search in the start bar for Java or type the following in Command Prompt (cmd.exe):
  - java –version



```
C:\Users\USER>java -version
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)
```

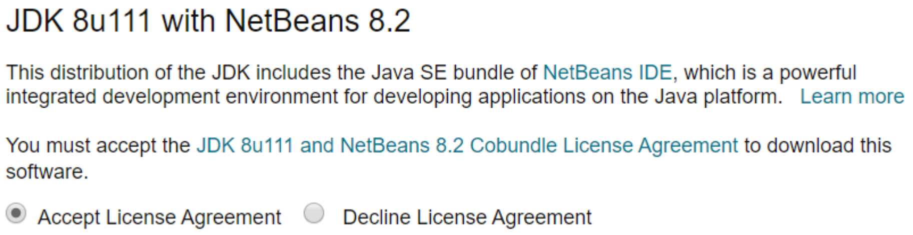
- If you do not have Java installed on your computer, you can download it for free at oracle.com.

Installing IDE for Java (NetBeans)

- Install JDK 8u111 with NetBeans 8.2
  - Use the following link (https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html)

- Or simply Google it!

- Choose Windows x64 and install

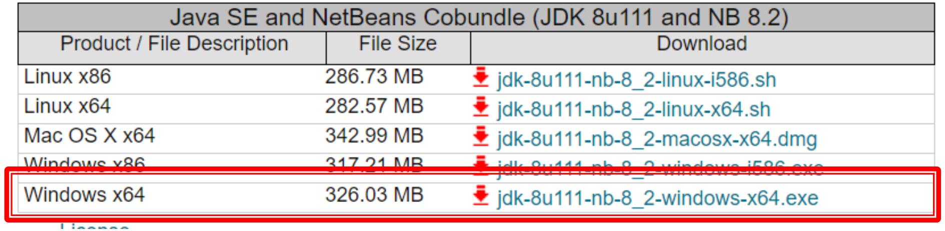


JDK 8u111 with NetBeans 8.2

This distribution of the JDK includes the Java SE bundle of NetBeans IDE, which is a powerful integrated development environment for developing applications on the Java platform. Learn more

You must accept the JDK 8u111 and NetBeans 8.2 Cobundle License Agreement to download this software.

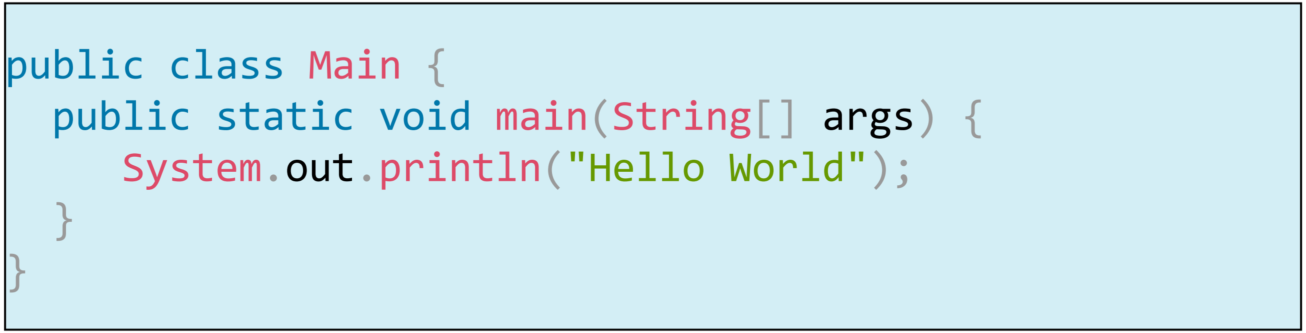
● Accept License Agreement ○ Decline License Agreement



Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	286.73 MB	jdk-8u111-nb-8_2-linux-i586.sh
Linux x64	282.57 MB	jdk-8u111-nb-8_2-linux-x64.sh
Mac OS X x64	342.99 MB	jdk-8u111-nb-8_2-macosx-x64.dmg
Windows x86	317.21 MB	jdk-8u111-nb-8_2-windows-i586.exe
Windows x64	326.03 MB	jdk-8u111-nb-8_2-windows-x64.exe

First program in Java

- In Java, every application begins with a class name, and that class must match the filename.
- Let's create our first Java file, called Main.java, which can be done in any text editor (like Notepad).



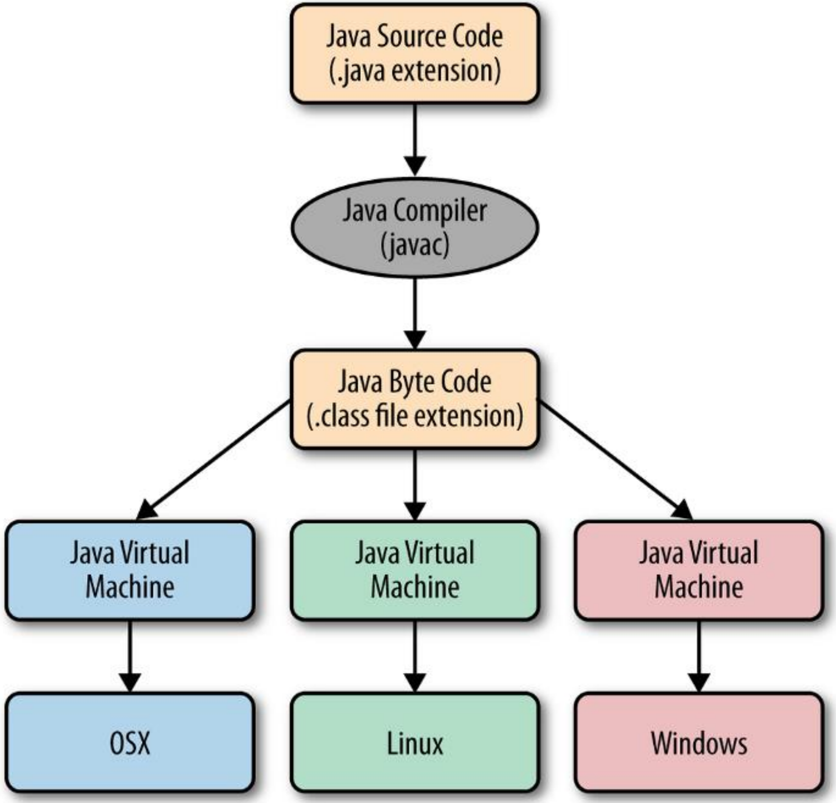
```java
public class Main {
  public static void main(String[] args) {
     System.out.println("Hello World");
  }
}
```

Save the code in Notepad as "Main.java".

Compile and Run Java program

- Open Command Prompt (cmd.exe), navigate to the directory where you saved your file, and type:
  - javac Main.java
- This will compile your code. If there are no errors in the code, the command prompt will take you to the next line. Now, to run the file type:
  - java Main

The JVM takes the byte code and generates machine code.



Java Editions

- Java Standard Edition (SE) contains the capabilities needed to develop desktop and server applications.
- The Java Enterprise Edition (Java EE) is geared toward developing large-scale, distributed networking applications and web-based applications.
- Java Micro Edition (Java ME) a subset of Java SE. geared toward developing applications for resource-constrained embedded devices, such as:
  - Smart watches
  - MP3 players
  - television set-top boxes
  - smart meters (for monitoring electric energy usage)
  - and more.

Need for OOP

Procedure-oriented Programming(POP) vs Object-oriented programming(OOP)

POP

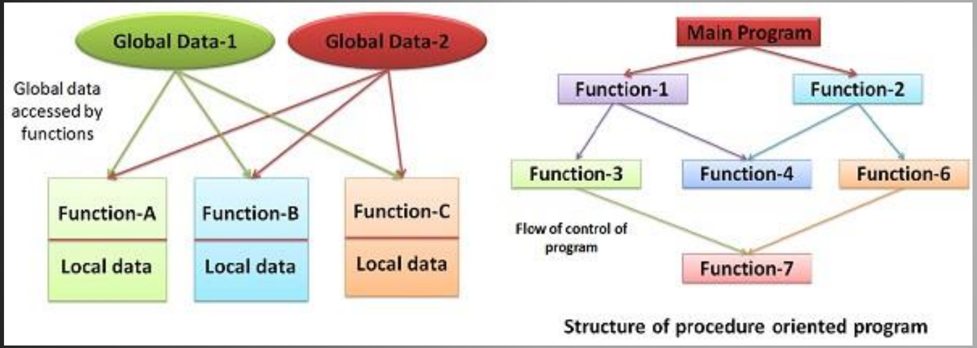
- Main focus is on **"how to get the task done"** i.e. on the procedure or structure of a program
- Large program is divided into functions(methods)
- **C,** VB, FORTRAN, Pascal

OOP

- Main focus is on 'data security'. Hence, only objects are permitted to access the entities of a class.
- Entire program is divided into objects.
- C++, JAVA, C#, Objective-C, phyton

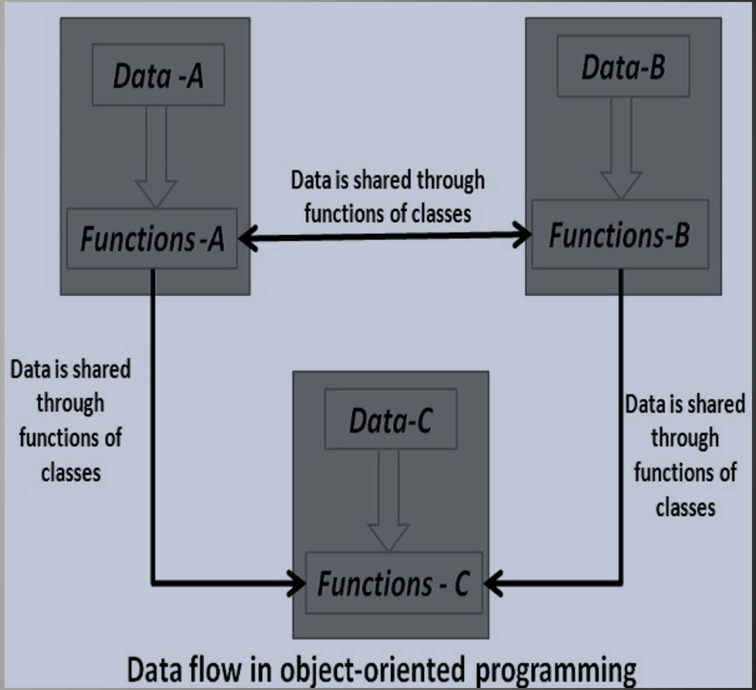
function = method

POP



Structure of procedure oriented program

OOP



Data -A
Functions -A
Data-B
Functions-B
Data is shared through functions of classes
Data is shared through functions of classes
Data-C
Functions - C
Data is shared through functions of classes

Data flow in object-oriented programming

Introduction to Object Technology

- Objects (comes from classes) are *reusable*.
  - Date, time, audio, video, automobile, people objects, etc.
  - Almost any *noun* can be represented as an **object** in terms of
    - *attributes* (e.g., name, color and size) and
    - *behaviors* (e.g., calculating, moving and communicating).

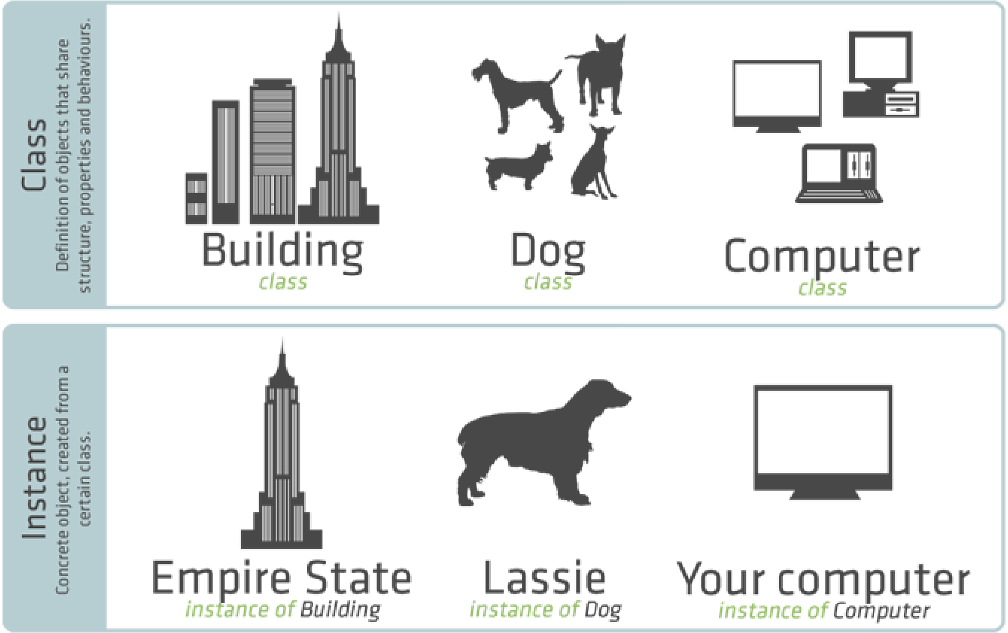
- Object-oriented design approach is much more productive than with earlier popular techniques like “structured programming”
- Object-oriented programs are often **easier to understand, correct and modify.**

Objects and Classes

- **Class** – A *class* is a blueprint or template or set of instructions to build a specific type of object

- **Object–** An object is a component that contains attributes and behaviors needed to make a certain type of data useful.

- **Instance–** An *instance* is a specific object built from a specific class



Class members

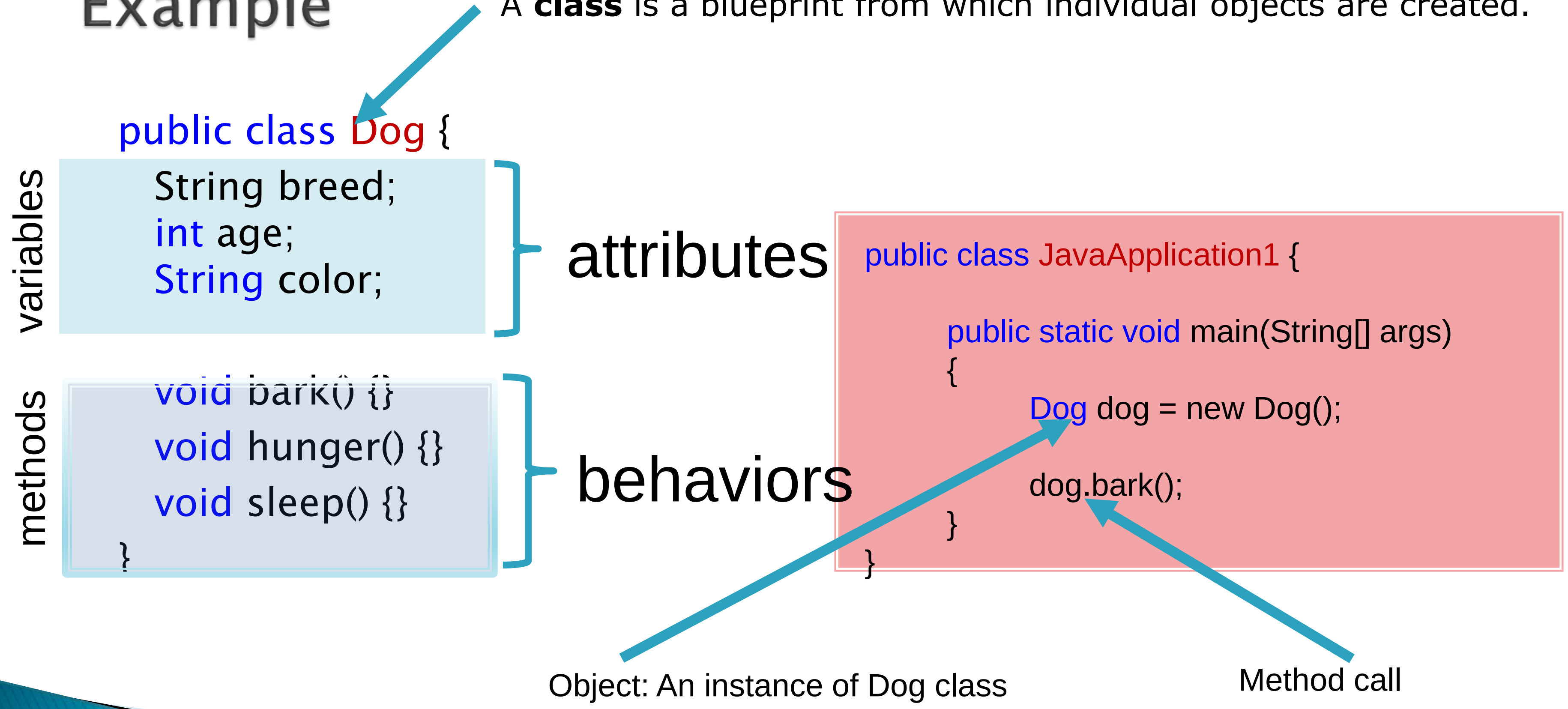
- Objects have
  - *attributes* (e.g., name, color and size) and (variables)
  - *behaviors* (e.g., calculating, moving and communicating). (methods)
- A car has *attributes*
  - Color, its number of doors, the amount of gas in its tank, its current speed and its record of total miles driven (i.e., its odometer reading).
  - The car's attributes are represented as part of its design in its engineering diagrams.
- Every car maintains its *own* attributes.
- *methods* are used to perform some tasks of the objects.

Instantiation

- Just as someone has to *build* a car from its engineering drawings before you can actually drive a car, you must *build an object* of a class before a program can perform the tasks that the class's methods define.
- An object is then referred to as an instance of its class.

Example

A **class** is a blueprint from which individual objects are created.



```
public class Dog {
    String breed;
    int age;
    String color;

    void bark() {}
    void hunger() {}
    void sleep() {}
}
```

variables

attributes

methods

behaviors

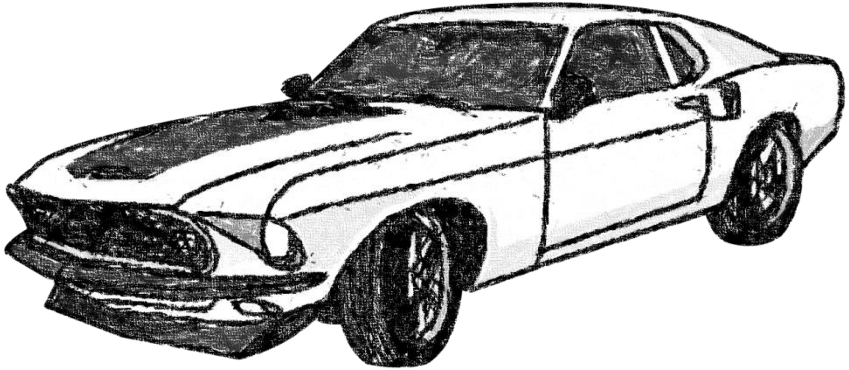
```
public class JavaApplication1 {

    public static void main(String[] args)
    {
        Dog dog = new Dog();

        dog.bark();
    }
}
```

Object: An instance of Dog class

Method call

Another Example: Car class

Attributes
Model
Color
Year
Price



Behaviours
StartEngine
Drive
Stop

This Car class can be *reused* many times to build many cars, you can reuse a class many times to build many objects.

Reuse of existing classes when building new classes and programs saves time and effort.

Online source

- https://sites.google.com/view/oopinjava

Thanks ☺