Object-Oriented Programming

Introduction to Java

Contents

- Brief history of Java
- Java platforms and applications
- Writing your first java program
- Compile and run your first Java program
- Code structure
- Basic data types and operators
- Loop control and decision making

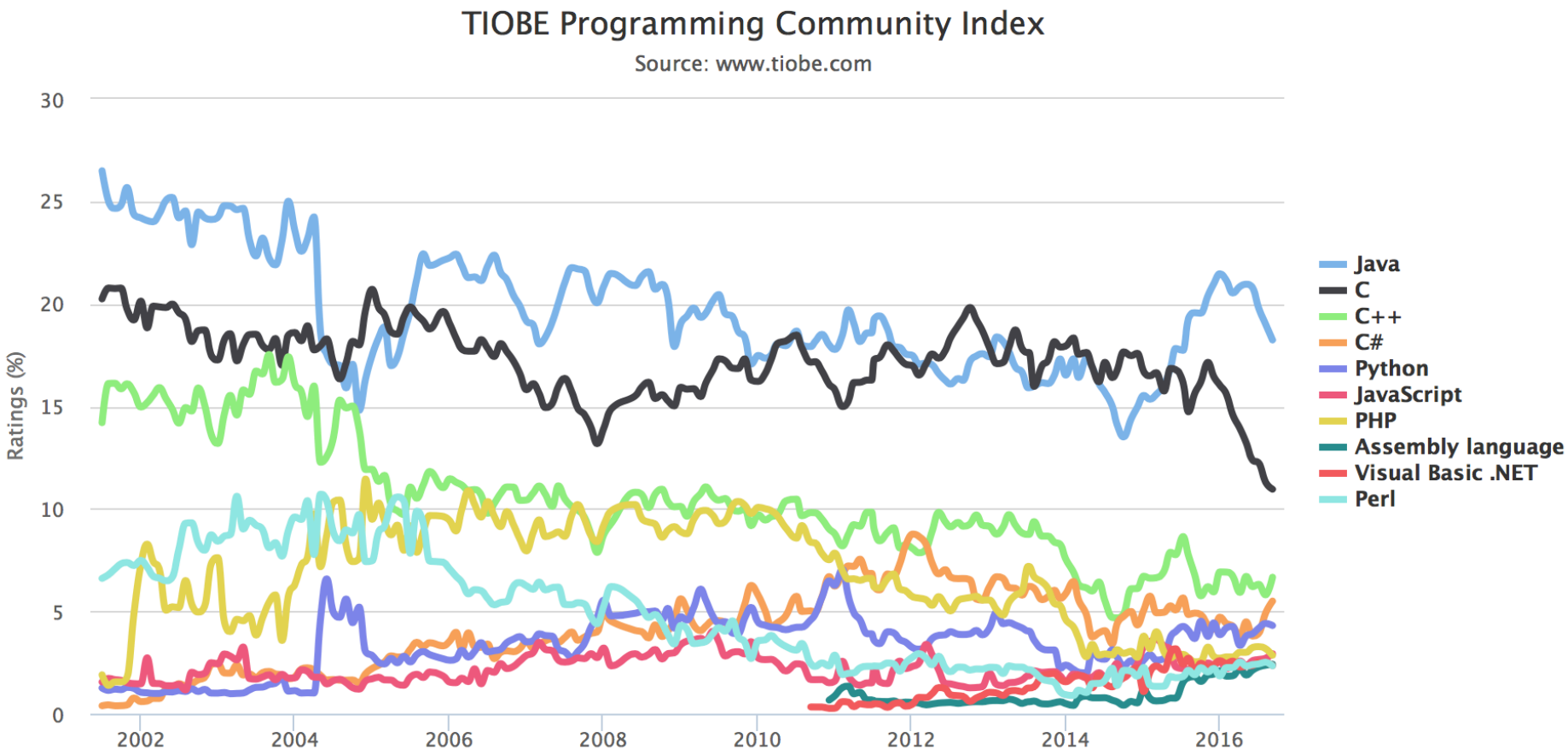
History of Java

- 1991: developed by Sun Microsystems as a small programming language for embedded household devices
  - initially called Oak
- Java 1.0.2 (1996), Java 1.1 (1997)
  - "Write Once, Run Anywhere"
  - very slow
  - became popular with Web pages running Applets
- Java 2 (versions 1.2 – 1.4 from 1998-2002)
  - much faster, powerful
  - 3 platforms: J2SE, J2EE, J2ME
- Java 5,6,7,8 (versions 1.5 – 1.8 in 2004, 2006, 2011, 2014)
  - more powerful

Java Platforms and Applications

- Desktop Applications - Java Standard Edition (J2SE)
  - Java Application: normal Java application running on desktops; console or GUI
  - Java Applet: embedded application running within Web browsers
- Server Applications - Java Enterprise Edition (J2EE)
  - Web Services, JavaServer Pages (JSP), Servlet
- Mobile Applications - Java Micro Edition (J2ME)

Why Java?


TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
30
25
20
15
10
5
0
2002
2004
2006
2008
2010
2012
2014
2016
Java
C
C++
C#
Python
JavaScript
PHP
Assembly language
Visual Basic .NET
Perl

Installing Java

- Download and install Java Development Kit (JDK) on Windows, Linux or Mac OS

  http://www.oracle.com/technetwork/java/javase/downloads/index.html

- Remember to set PATH

Java Development Kit (JDK)

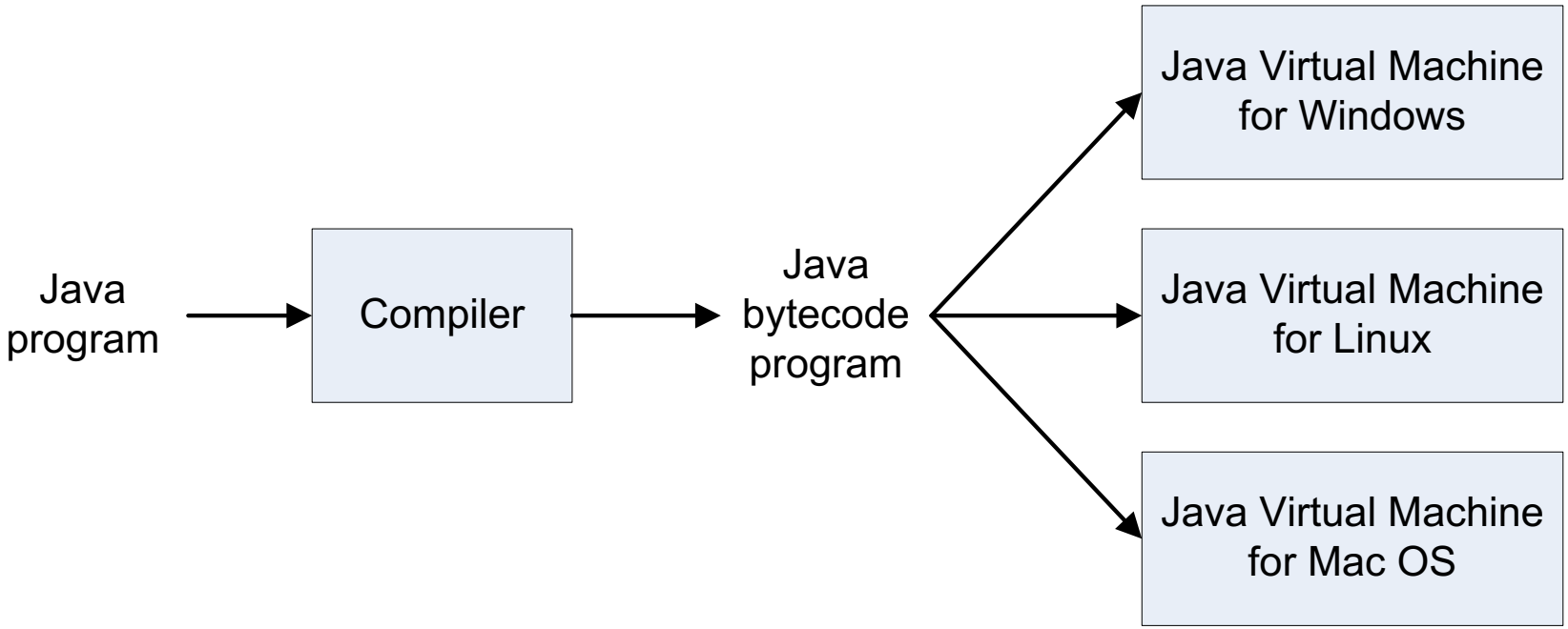
- Free development and run-time environment
- Main components:
  - **javac :** compiler, converts source code into Java bytecode
  - **java :** interpreter and application loader
  - **javadoc :** documentation generator, automatically generates documentation from source code comments
  - **jdb :** debugger

Java Editor/IDE

- Editor and Integrated Development Environment (IDE):
  - Notepad, EditPlus, Notepad++
  - Eclipse
  - NetBeans
  - IntelliJ Community Edition is recommended
- Build tool:
  - Gradle
  - Maven is recommended

Running Java Codes

- Java source code is compiled into bytecode
- Bytecode is executed in an interpreter environment, called Java Virtual Machine



Java Virtual Machine (JVM)

- Provide Java programs with run-time environments
- Normally provided as software:

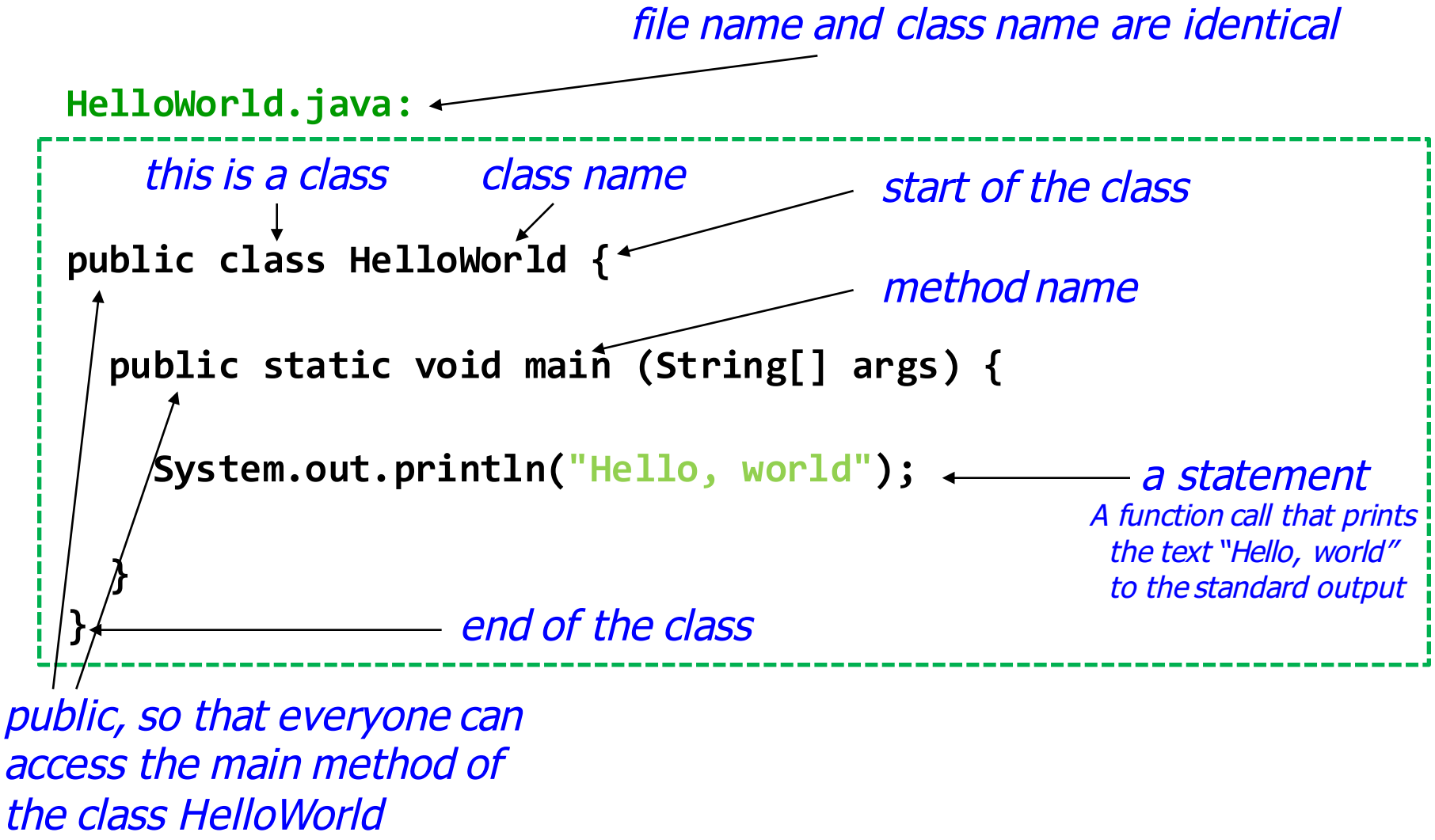
  JRE: Java Runtime Environment

- Depend on specific hardware and OS
- Java platform: JVM + APIs (Application Programming Interface)

Writing your first Java program

- In Java, everything goes in a **class**
- When you run a program, you run a class:
  - load the class then start executing the class's main() method
  - Each Java program MUST have a main() method

Writing your first Java program



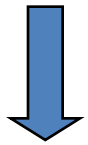
Compile and Run your first program

```
// Java source code is stored in the file HelloWorld.java
public class HelloWorld {
  public static void main (String[] args)
   {
     System.out.println("Hello, world");
   }
}
```

To compile HelloWorld.java,
type javac HelloWorld.java

compile

HelloWorld.class

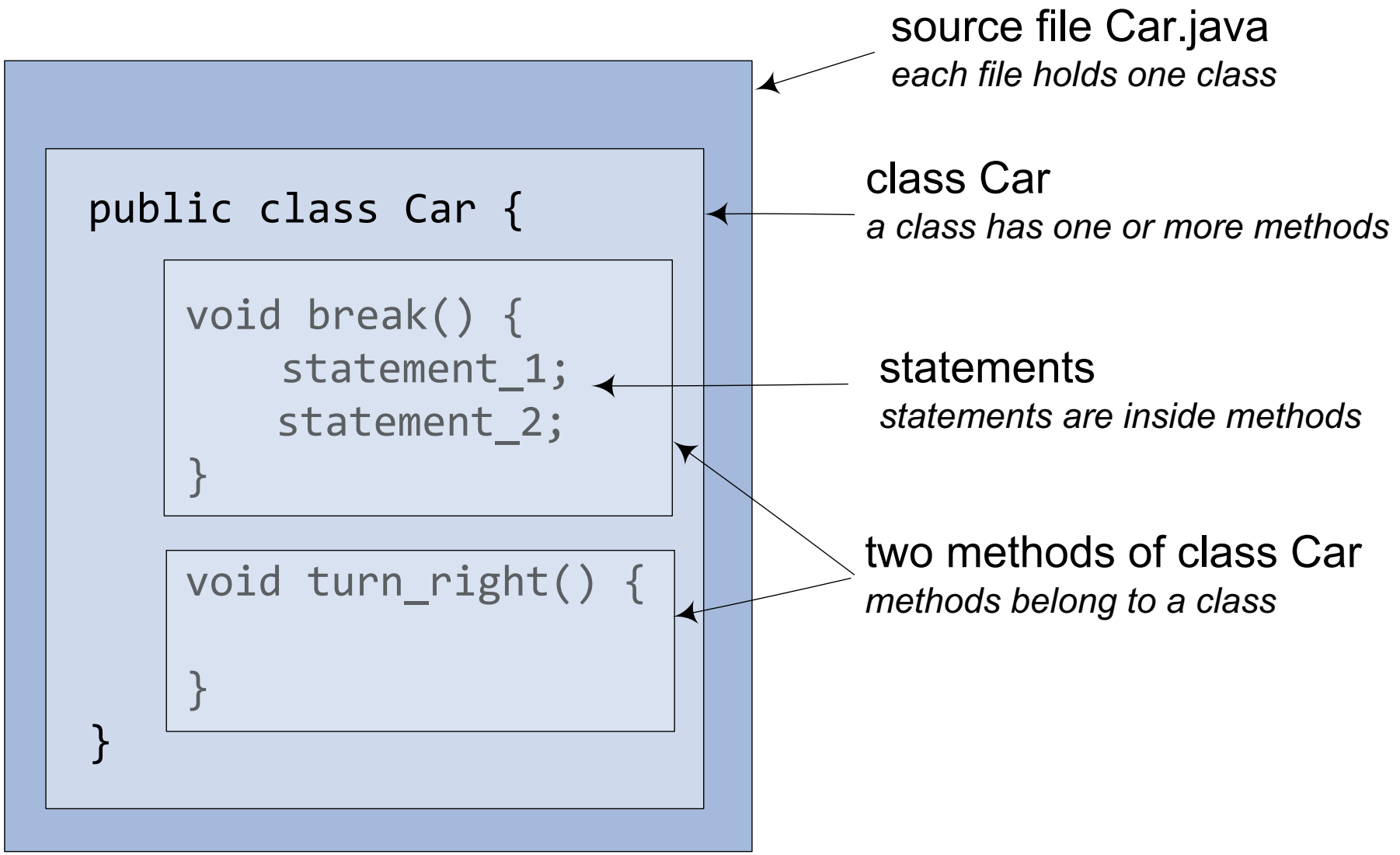


To run HelloWorld.main(),
type java HelloWorld

run

```
%> javac HelloWorld.java
%> java HelloWorld
Hello, world
```

Code Structure



Application with more than one class

Two classes are stored in two separated files:

TestGreeting.java:

```
public class TestGreeting {
  public static void main(String[] args) {
    Greeting gr = new Greeting();
    gr.greet();
  }
}
```

Greeting.java:

```
public class Greeting {
  public void greet() {
    System.out.print("Hi there!");
  }
}
```

Compile and Run

- Compile

  `javac TestGreeting.java`

  - `Greeting.java` is automatically compiled
- Run

  `java TestGreeting`

```
%> javac TestGreeting.java
%> java TestGreeting
Hi there!
```

Basic Data Types

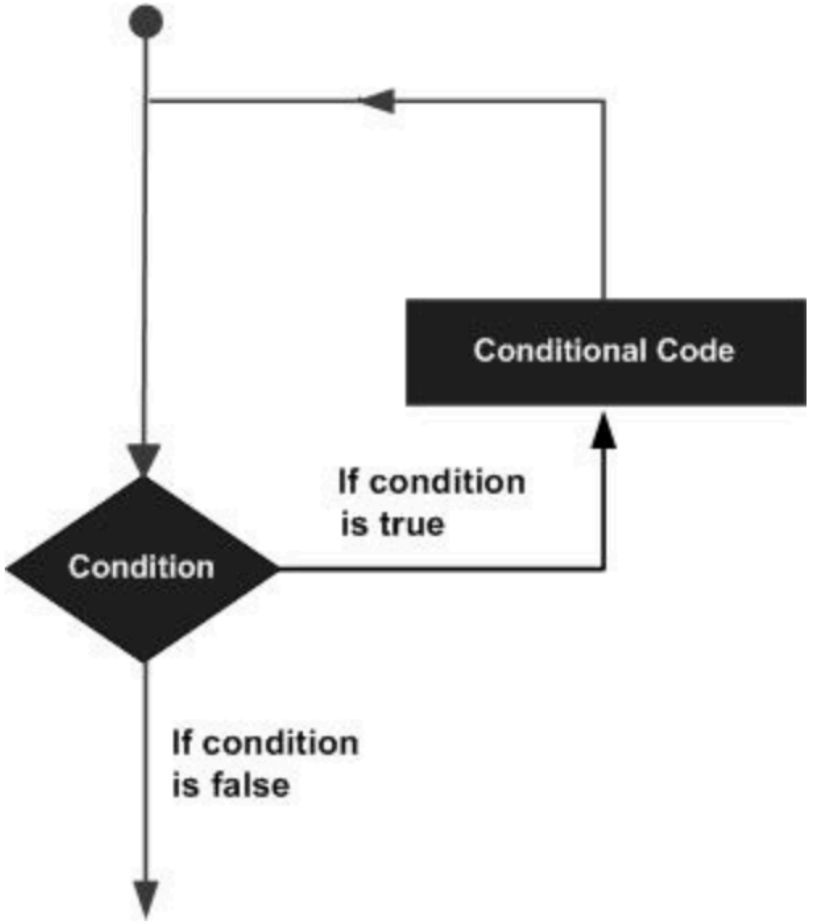
Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Basic Operators

	Operator	Type
unary operator →	+ +, - -	Unary operator
Binary operator	+, -, *, /, %	Arithmetic perator
	<, <=, >, >=, ==, !=	Relational operator
	&&, | |, !	Logical operator
	&, |, <<, >>, ~, ^	Bitwise operator
	=, +=, - =, *=, /=, %=	Assignment operator
Ternary operator →	?:	Ternary or conditional operator

Loop Control Statements

- while loop
- for loop
- do…while loop



Example of While Loop

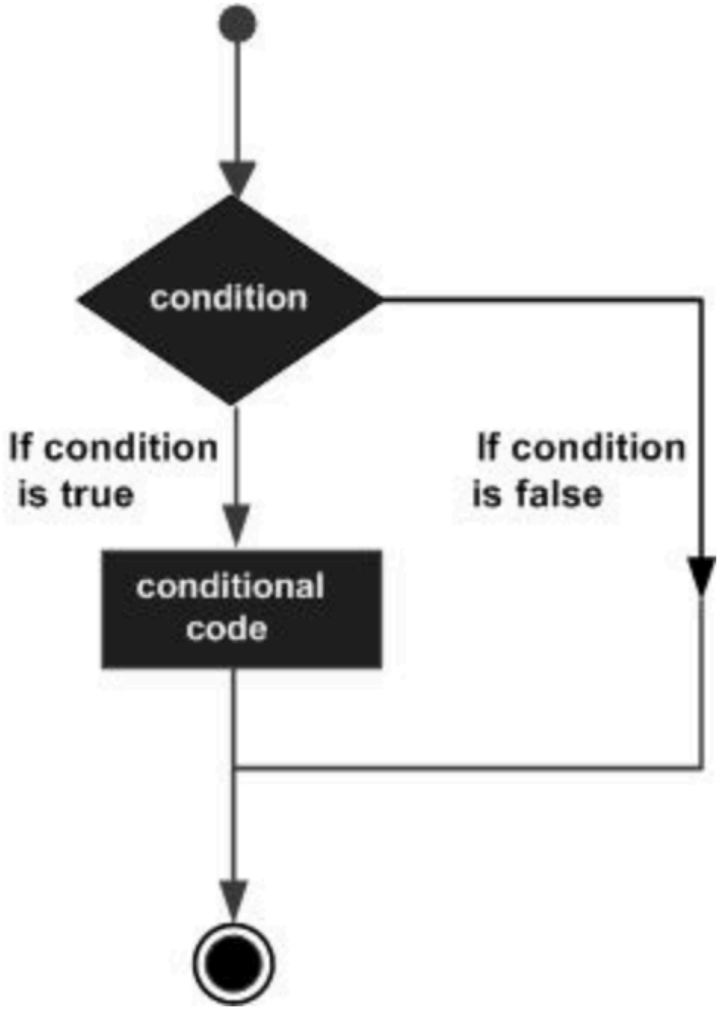
```
class WhileLoopExample {
    public static void main(String args[]){
        int i=10
        while(i>1){
            System.out.println(i);
            i--;
        }
    }
}
```

Example of For Loop

```
class ForLoopExample {
    public static void main(String args[]){
        for(int i=10; i>1; i--){
            System.out.println("The value of i is: "+i);
        }
    }
}
```

Decision Making Statements

- if statement
- if…else statement
- switch statement



Example of If Statement

```
public class IfStatementExample {
    public static void main(String args[]){

        int x = 5;

        if (x == 2) {
          System.out.println("x must be 2");
        } else {
          System.out.println("x is not 2");
        }
    }
}
```

What else can we do?

- do-while?
- switch?
- int, long, float, double, boolean,…?
- other Java basics?

Read the text books!