

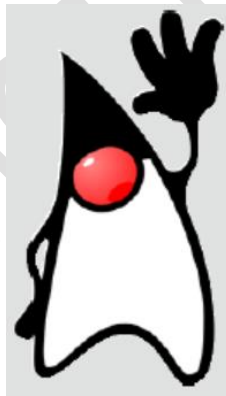


---

# JAVA FOUNDATIONS 1Z0-811

---

ORACLE ACADEMY



2 DE SEPTIEMBRE DE 2025

<https://academy.oracle.com/>

[HTTPS://GITHUB.COM/ISC-UPA/2025-3-TIID3C-POO](https://github.com/ISC-UPA/2025-3-TIID3C-POO)

## Contenido

1. Introduction .....	2
1.1. Technological Requirements: .....	2
1.2. Create Java Project: .....	3
1.3. Setting Up Java .....	4
2. Java Basics .....	6
2.1. The Software Development Process.....	6
2.2. What is my Program Doing? .....	7
2.3. Introduction to Object-Oriented Programming Concepts .....	7
3. Java Data Types.....	8
3.1. What is a Variable? .....	8
String x ="Sam"; .....	8
3.2. Numeric Data.....	8
Rules of Precedence.....	9
3.3. Textual Data.....	10
Primitives .....	10
Escape Sequence.....	11
3.4. Converting Between Data Types .....	11
3.5. Keyboard Input \$\$ .....	13
4. Java Methods and Library Classes.....	13
5. Decision Statements .....	13
6. Loop Constructs .....	13
7. Creating Classes .....	13
8. Arrays and Exceptions.....	13
9. JavaFX.....	13

→

# 1. Introduction

## 1.1. Technological Requirements:

Java JDK <https://www.oracle.com/java/technologies/downloads/>

VS Code <https://code.visualstudio.com/Download>

Extensions: **Extension Pack for Java**

jdk-8u202-windows-x64.exe

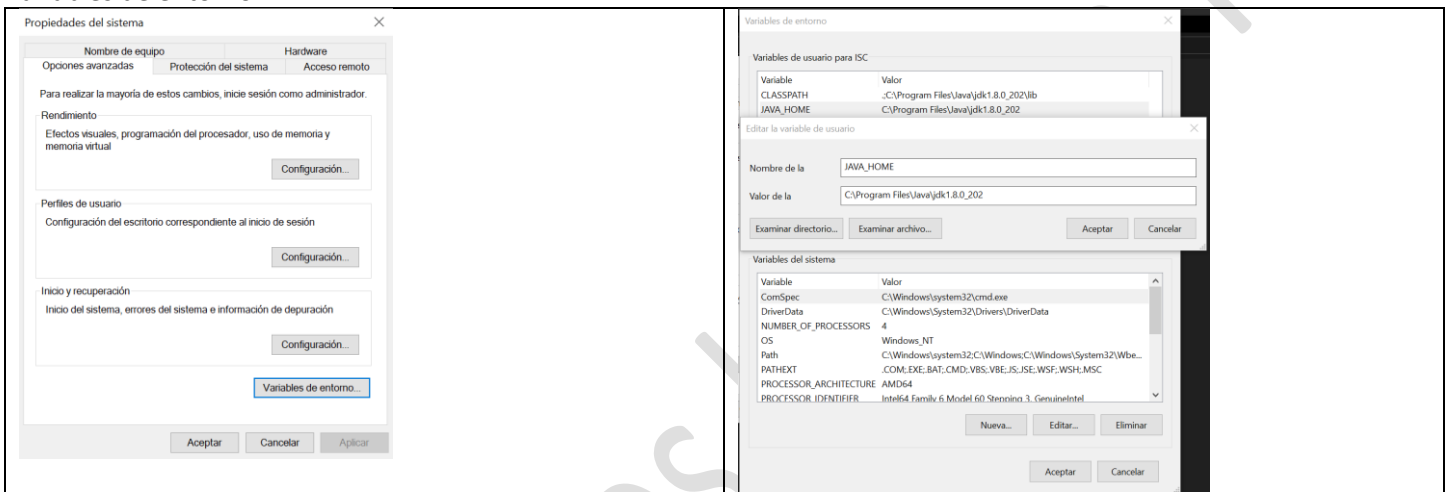
VSCodeSetup-x64-1.103.2.exe

Integrated Development Environment (IDE)

Eclipse IDE: <https://www.eclipse.org/downloads/packages/>

NetBeans IDE <https://netbeans.apache.org/download/index.html>

### Variables de entorno



Panel de control -> Sistema -> Configuración avanzada del sistema

Opciones avanzadas -> Variables de entorno -> Variables de Usuario

JAVA_HOME C:\Program Files\Java\jdk1.8.0_202	PATH %JAVA_HOME%\BIN
CLASSPATH . ; %JAVA_HOME%\LIB	Probar Instalación desde CMD C:\>java -version (correr) C:\>javac -version (compilar)

C:\dev>java -version java version "1.8.0_202"  C:\dev>javac -version javac 1.8.0_202  C:\dev\poo>javac Hola.java  C:\dev\poo>java Hola Hello World!	public class Hola {  public static void main(String[] args) { System.out.println("Hello World!"); } }
--	--

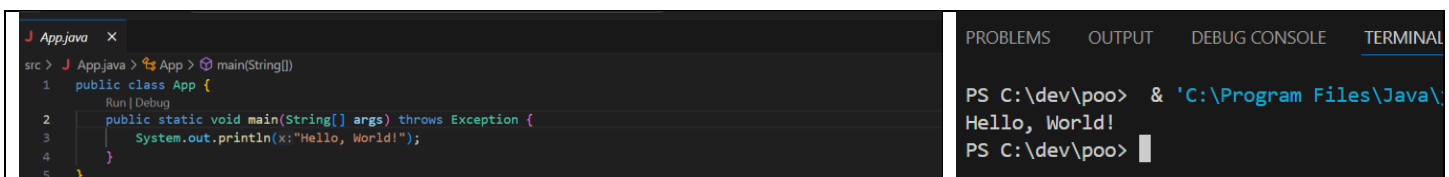
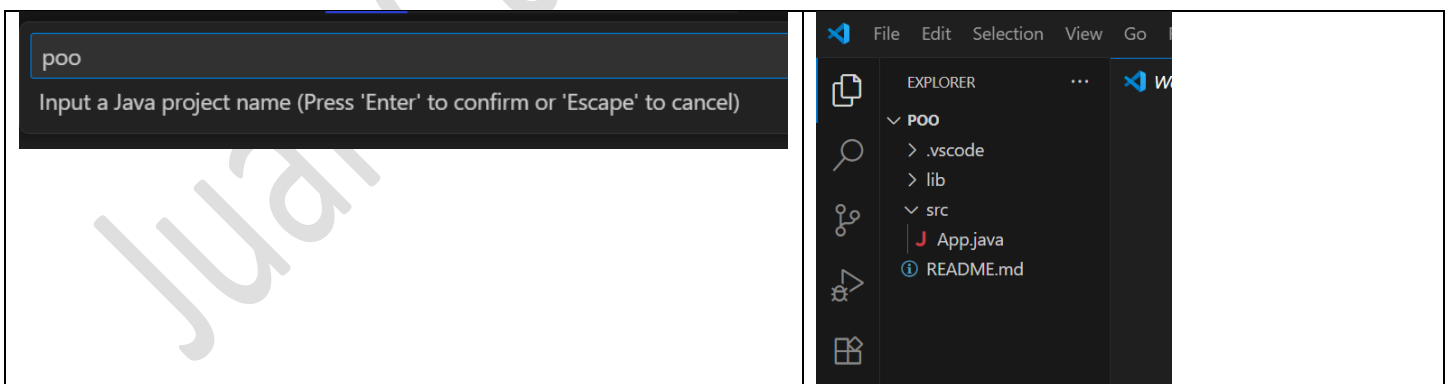
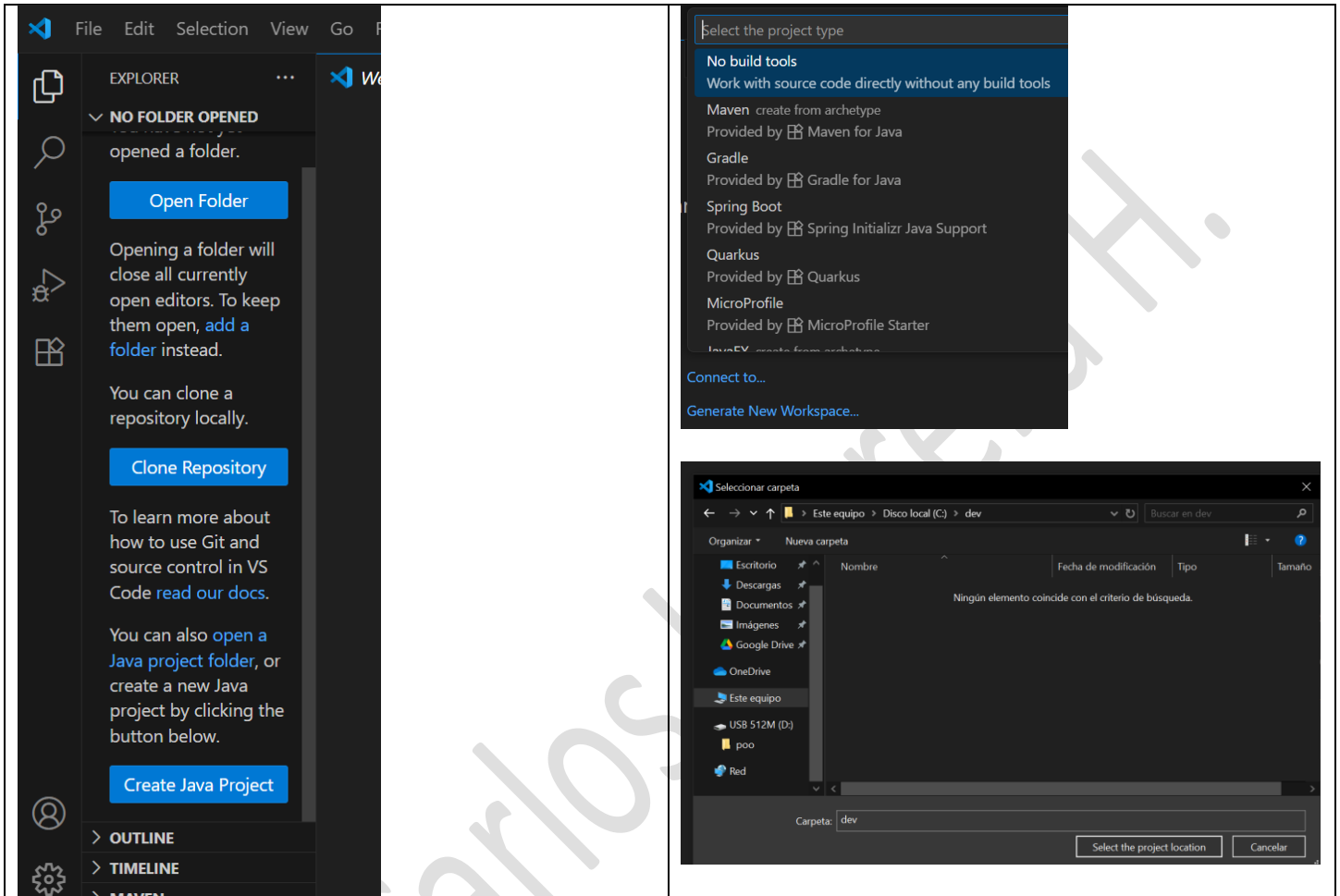
→

Crear un Directorio de trabajo

C:\> mkdir dev

C:\dev>

## 1.2. Create Java Project:



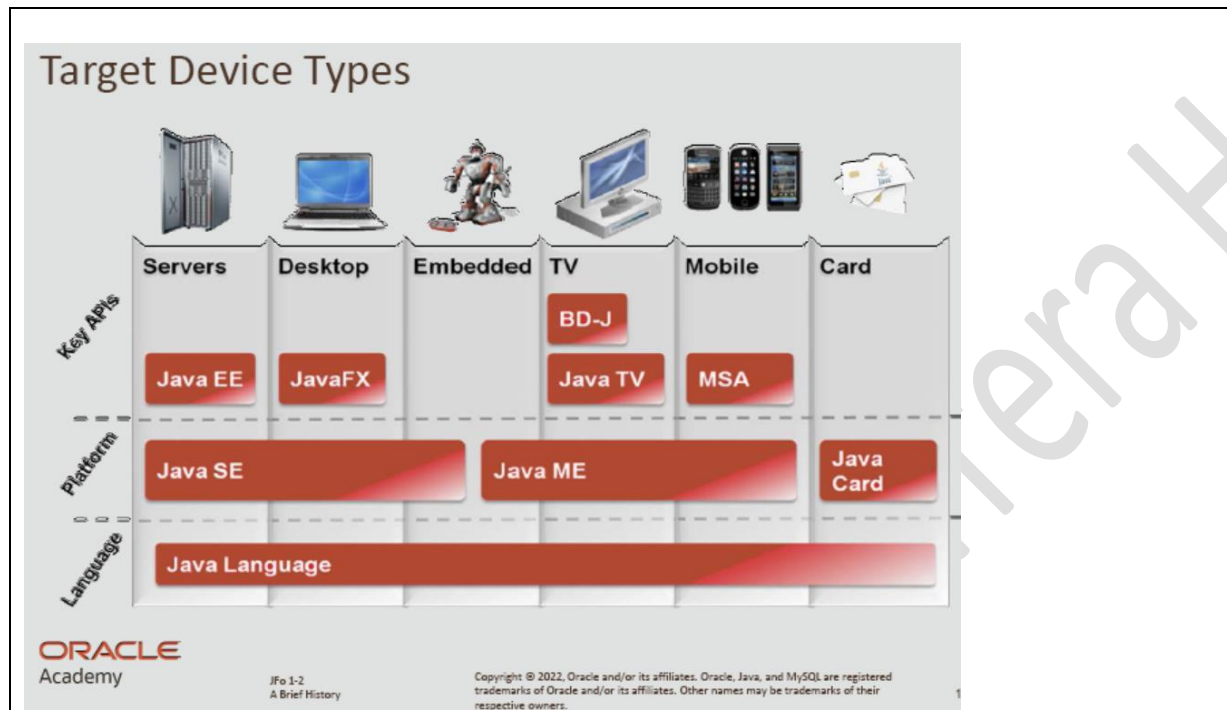
→

### 1.3. Setting Up Java

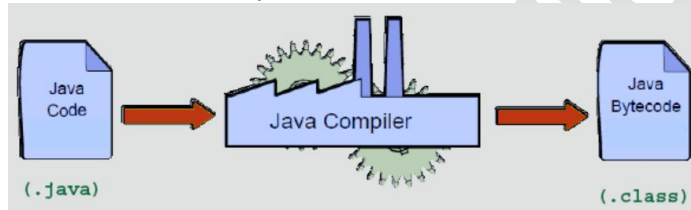
James Gosling is considered the “Father of Java”. Duke, the Java Mascot.

Oracle acquired Sun Microsystems in 2010, and released JDK 7 in 2011, and JDK 8 in 2014.

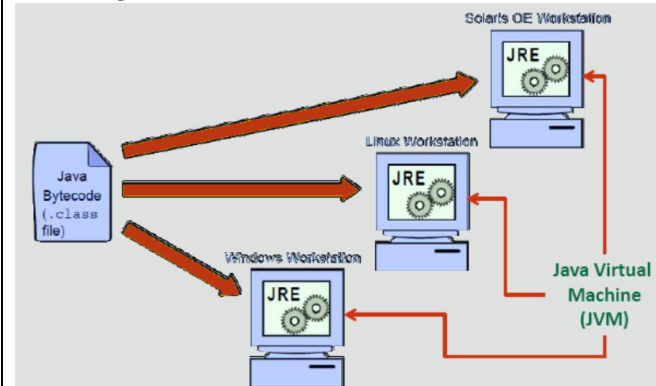
Jakarta EE Is used to create large enterprise, server-side, and client-side distributed applications



#### Java is Platform-Independent



#### Java Programs Run in a JVM



#### Java Runtime Environment (JRE)

Includes:

- The Java Virtual Machine (JVM)
- Java class libraries

Purpose:

- Read bytecode (.class)
- Run the same bytecode anywhere with a JVM

#### Java Development Kit (JDK)

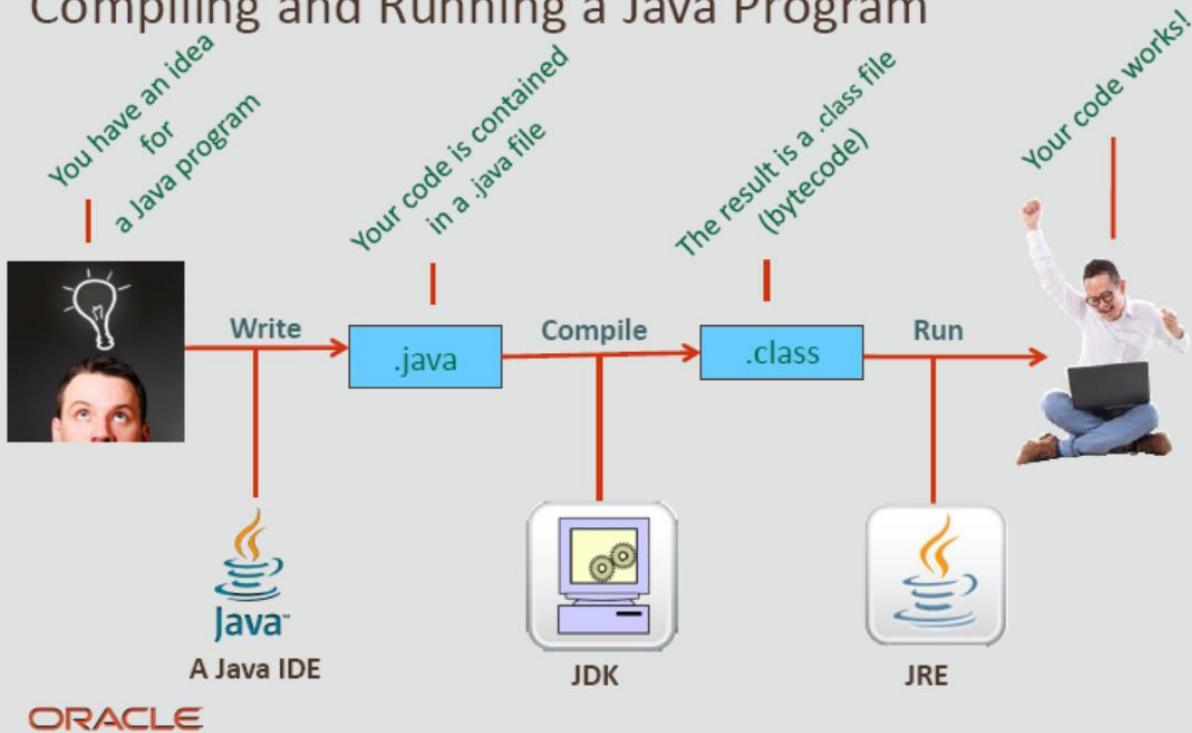
Includes:

- JRE Java Compiler
- Additional tools

Purpose:

Compile bytecode (.java → .class)

# Compiling and Running a Java Program



A Java IDE is used to **write** source code ( `.java` )



The JDK **compiles** bytecode ( `.java` → `.class` )



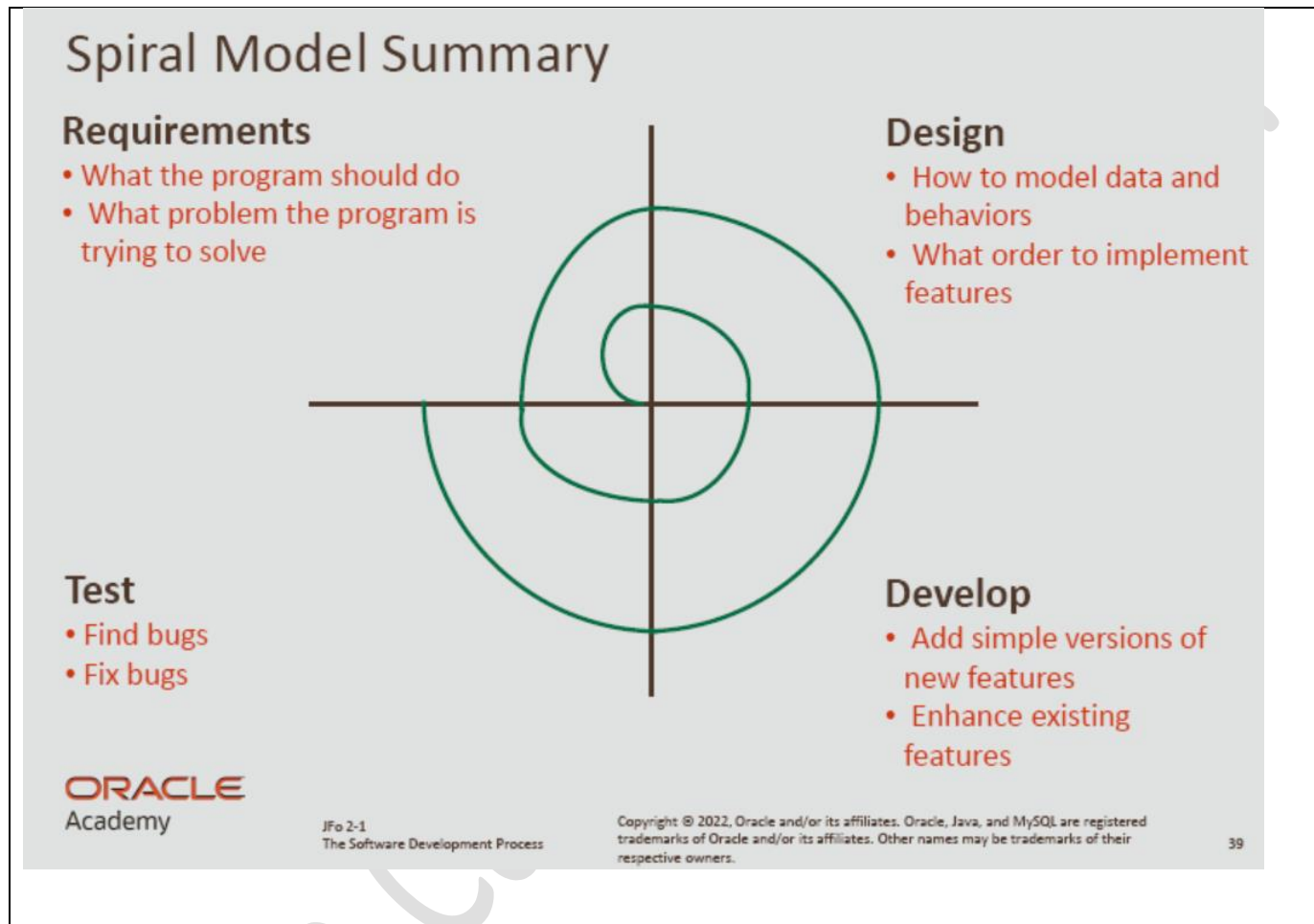
Bytecode **runs** in a JVM, which is part of the JRE

→

## 2. Java Basics

### 2.1. The Software Development Process

#### Spiral Model of Development



<https://objectstorage.uk-london-1.oraclecloud.com/n/lrvrlgaqj8dd/b/Games/o/JavaPuzzleBall/index.html>

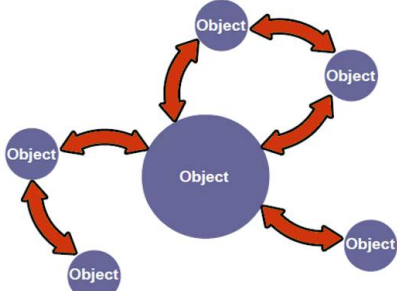
→

## 2.2. What is my Program Doing?

<p>Code within curly braces is called a block of code</p> <p>Indentation before a line of code (4 spaces)</p> <p>Whitespace</p> <p>End statements with semicolons (;)</p> <p>// Single-line comments</p> <p>Multi-line comments</p> <pre>/* Bienvenidos    a poo */</pre>	<pre>public static void NombreMetodo() { . . } NombreMetodo(); // llamar al método</pre> <p>Debug</p> <p>To set a breakpoint</p> <p>Press Step Over</p>
---	---

## 2.3. Introduction to Object-Oriented Programming Concepts

<p>Procedural languages ...</p> <ul style="list-style-type: none"> <li>• Read one line at a time</li> <li>• The C language is procedural</li> </ul>	<p>Object-oriented languages...</p> <ul style="list-style-type: none"> <li>• Read one line at a time</li> <li>• Model objects through code</li> <li>• Emphasize object interaction</li> <li>• Allow interaction without a prescribed order</li> <li>• Java and C++ are object-oriented languages</li> </ul>
---	---

<p>Object-Oriented Programming</p> <ul style="list-style-type: none"> <li>• Interaction of objects</li> <li>• No prescribed sequence</li> </ul>	
---	--

### Modeling Properties and Behaviors

<p><b>Customer class</b></p> <pre>name address billing info age customer number order number  requestDiscount() setAddress() shop() displayCustomer()</pre> <p><b>Class name</b></p> <p><b>Fields</b></p> <p><b>Methods</b></p>	<p><b>Class declaration</b></p> <pre>1 public class Customer { 2     public String name = "Junior Duke"; 3     public int custID = 1205; 4     public String address; 5     public int orderNum; 6     public int age; 7 8     public void displayCustomer() { 9         System.out.println("Customer: "+name); 10    } //end method displayCustomer 11 } //end class Customer</pre> <p><b>Fields (Properties) (Attributes)</b></p> <p><b>Methods (Behaviors)</b></p>
---	---





### 3. Java Data Types

#### 3.1. What is a Variable?

```
String x = "Sam";  
System.out.println("My name is " + x);
```

Variables03.java (There are 6 mistakes)

Type	Keyword	Example Values
Boolean	<code>boolean</code>	<code>true</code> , <code>false</code>
Integer	<code>int</code>	1, -10, 20000, 123_456_789
Double	<code>double</code>	1.0, -10.0005, 3.141
String	<code>String</code>	"Alex", "I ate too much dinner."

#### Variable Naming Conventions

- Begin each variable with a lowercase letter
- Subsequent words should be capitalized: `myVariable`
- Choose names that are mnemonic and that indicate the intent of the variable to the casual observer
- Remember that ...
- Names are case-sensitive
- Names can't include white space

```
Int studentAge = 20;  
String myCatchPhrase = "Enjoy Alex Appreciation Day!";
```

#### 3.2. Numeric Data

##### Integral Primitive Types

Type	Length	Number of Possible Values	Minimum Value	Maximum Value
<code>Byte</code>	8 bits	$2^8$ , or... 256	$-2^7$ , or... -128	$2^7-1$ , or... 127
<code>short</code>	16 bits	$2^{16}$ , or... 65,535	$-2^{15}$ , or... -32,768	$2^{15}-1$ , or... 32,767
<code>int</code>	32 bits	$2^{32}$ , or... 4,294,967,296	$-2^{31}$ , or... -2,147,483,648	$2^{31}-1$ , or... 2,147,483,647
<code>long</code>	64 bits	$2^{64}$ , or... 18,446,744,073,709,551 ,616	$-2^{63}$ , or... -9,223,372,036, 854,775,808L	$2^{63}-1$ , or... 9,223,372,036, 854,775,807L

`++`   `--`   `*=`   `/=`   `%=`   `++`   `--`   Pre/Post   `a+=b`   `a = a + ( b )`

## Floating Point Primitive Types

Type	Float Length	When will I use this?
<code>float</code>	32 bits	Never
<code>double</code>	64 bits	Often

`double x = 9/2;`      `double x = 9/2.0;`

`final double PI = 3.141592;`

### Rules of Precedence

- Operators within a pair of parentheses
- Increment and decrement operators (++or --)
- Multiplication and division operators, evaluated from left to right
- Addition and subtraction operators, evaluated from left to right
- If operators of the same precedence appear successively, the operators are evaluated from left to right

`int x = (((25 - 5) * 4) / (2 - 10)) + 4;`

`int y = 25 - 5 * 4 / 2 - 10 + 4;`

→

### 3.3. Textual Data

Use the char data type

Use Strings

Concatenate Strings

Understand escape sequences

Understand print statements better

Char is used for a single character (16 bits) char shirtSize= 'M';	A String can handle multiple characters String greeting = "Hello World!";
---	--

#### Primitives

Type	Length	Data
boolean	1 bit	true / false
byte	8 bits	Integers
short	16 bits	Integers
int	32 bits	Integers
long	64 bits	Integers
float	32 bits	Floating point numbers
double	64 bits	Floating point numbers
char	16 bits	Single characters
<b>Where are Strings?</b>		
String is capitalized		
<ul style="list-style-type: none"><li>• Strings are an object, not a primitive</li><li>• Object types are capitalized by convention</li></ul>		

Combining multiple Strings is called concatenation

```
String totalPrice = "Total: $" +3 +2 +1;
```

```
String totalPrice = 3 +2 + 1 + "Total: $";
```

```
String totalPrice = "Total: $" +(3 +2 +1);
```

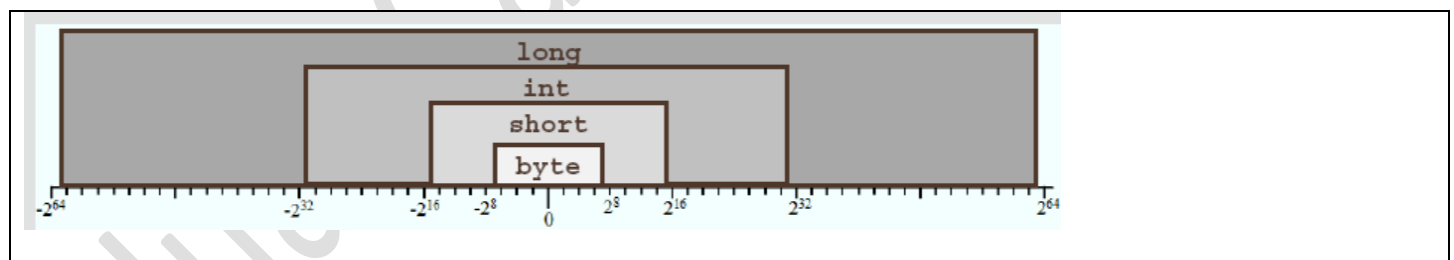
## Escape Sequence

Escape Sequence	Description	<pre>System.out.println("The cat said \"Meow!\" to me."); println() vs. print()  System.out.println("1\t2\t3\t\"Hola\" mundo"); 1    2    3    "Hola" mundo  System.out.println("Hola\nAdios"); Hola Adios</pre>
<code>\t</code>	Insert a new tab	
<code>\b</code>	Insert a backspace	
<code>\n</code>	Insert a new line	
<code>\r</code>	Insert a carriage return	
<code>\f</code>	Insert a formfeed	
<code>\'</code>	Insert a single quote character	
<code>\"</code>	Insert a double quote character	
<code>\\</code>	Insert a backslash character	

```
System.out.println("This is the first line."
    + "This is NOT the second line.");
sout tab "Metodo abreviado"
```

## 3.4. Converting Between Data Types

<pre>double x = 9 / 2; // Should be 4.5 System.out.println(x); // prints 4.0  double y = 4; System.out.println(y); //prints 4.0</pre>	<pre>int    num1 = 7; double num2 = 2; double num3; num3 = num1 / num2; // num3 is 3.5</pre>
---	--



<ul style="list-style-type: none"> <li>• Automatic promotions: <ul style="list-style-type: none"> <li>- If you assign a smaller type to a larger type: <pre> byte → short → int → long </pre> </li> <li>- If you assign an integral value to a floating-point type: <pre> 4 → 4.0 </pre> </li> </ul> </li> <li>• Examples of automatic promotions: <ul style="list-style-type: none"> <li>- <code>long</code> <code>intToLong</code> = 6;</li> <li>- <code>double</code> <code>intToDouble</code> = 4;</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• When to cast: <ul style="list-style-type: none"> <li>- If you assign a larger type to a smaller type: <pre> byte ← short ← int ← long </pre> </li> <li>- If you assign a floating point type to an integral type: <pre> 3 ← 3.0 </pre> </li> </ul> </li> <li>• Examples of casting: <ul style="list-style-type: none"> <li>- <code>int</code> <code>longToInt</code> = (<code>int</code>) 20L;</li> <li>- <code>short</code> <code>doubleToShort</code> = (<code>short</code>) 3.0;</li> </ul> </li> </ul> <pre> double pi = 3.1416 int entero = (int) pi </pre>
---	---

127 in binary is 01111111; 128 in binary is 10000000.

Java uses the first bit in a number to indicates sign (+/-)

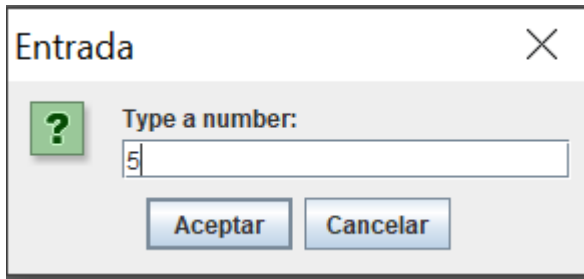
byte, short, and char values are automatically promoted to int prior to an operation

<ul style="list-style-type: none"> <li>• Solution using larger data type: <pre> int num1 = 53; int num2 = 47; int num3; num3 = (num1 + num2); </pre> <p>Changed from byte to int</p> </li> <li>• Solution using casting: <pre> int num1 = 53; // 32 bits of memory to hold the value int num2 = 47; // 32 bits of memory to hold the value byte num3; // 8 bits of memory reserved num3 = (byte)(num1 + num2); // no data loss </pre> </li> </ul>	<h3>Automatic Promotion</h3> <ul style="list-style-type: none"> <li>• Example of a potential problem: <pre> short a, b, c; a = 1; b = 2; c = a + b; //compiler error </pre> <p>a and b are automatically promoted to integers</p> </li> <li>• Example of potential solutions: <ul style="list-style-type: none"> <li>- Declare c as an <code>int</code> type in the original declaration: <pre> int c; </pre> </li> <li>- Type cast the (a+b) result in the assignment line: <pre> c = (short) (a+b); </pre> </li> </ul> </li> </ul> <pre> int x = 123_456_789; int x = 123456789;  intintVar1 = Integer.parseInt("100"); doubledoubleVar2 = Double.parseDouble("2.72"); </pre>
---	---

→

### 3.5. Keyboard Input \$\$

```
String input = JOptionPane.showInputDialog("Type a number:");  
int number = Integer.parseInt(input);  
number *= 2;  
System.out.println("El doble es: " + number);
```



El doble es: 10

## 4. Java Methods and Library Classes

## 5. Decision Statements

## 6. Loop Constructs

## 7. Creating Classes

## 8. Arrays and Exceptions

## 9. JavaFX