JAVA FOUNDATIONS 1Z0-811

ORACLE ACADEMY
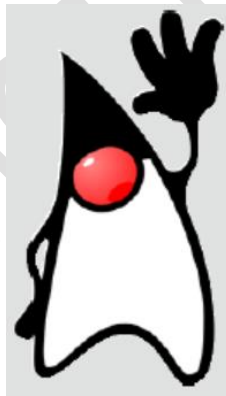
Contenido

→

1. Introduction

1.1. Technological Requirements:

Java JDK   https://www.oracle.com/java/technologies/downloads/   jdk-8u202-windows-x64.exe
VS Code   https://code.visualstudio.com/Download   VSCodeSetup-x64-1.103.2.exe
        Extensions: **Extension Pack for Java**

Integrated Development Environment (IDE)
Eclipse IDE: https://www.eclipse.org/downloads/packages/
NetBeans IDE   https://netbeans.apache.org/download/index.html

Variables de entorno
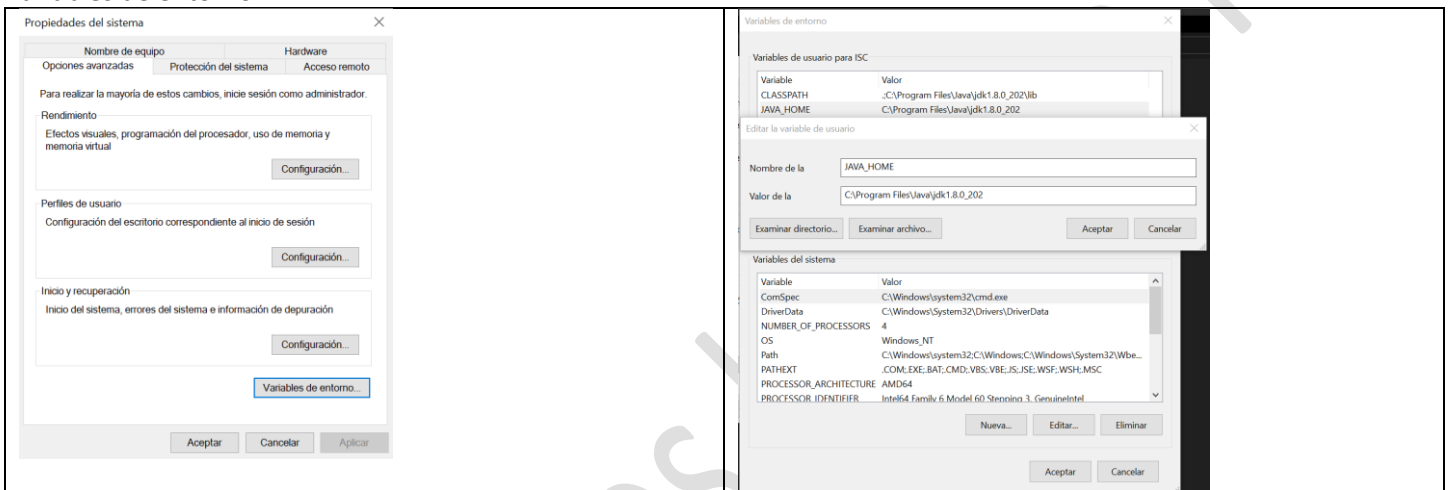
Panel de control -> Sistema -> Configuracion avanzada del sistema
Opciones avanzadas -> Variables de entorno -> Variables de Usuario

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<br>java version "1.8.0_202"<br><br>C:\dev>javac -version<br>javac 1.8.0_202<br><br>C:\dev\poo>javac Hola.java<br><br>C:\dev\poo>java Hola<br>Hello World! | ```java
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:

File Edit Selection View Go

EXPLORER ...

∨ NO FOLDER OPENED

opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

Clone Repository

To learn more about how to use Git and source control in VS Code read our docs.

You can also open a Java project folder, or create a new Java project by clicking the button below.

Create Java Project

> OUTLINE
> TIMELINE

Select the project type

No build tools
Work with source code directly without any build tools

Maven create from archetype
Provided by ⊞ Maven for Java

Gradle
Provided by ⊞ Gradle for Java

Spring Boot
Provided by ⊞ Spring Initializr Java Support

Quarkus
Provided by ⊞ Quarkus

MicroProfile
Provided by ⊞ MicroProfile Starter

Connect to...

Generate New Workspace...

Seleccionar carpeta

Este equipo > Disco local (C:) > dev

Organizar ▾   Nueva carpeta

Escritorio   Nombre   Fecha de modificación   Tipo   Tamaño
Descargas
Documentos   Ningún elemento coincide con el criterio de búsqueda.
Imágenes
Google Drive
OneDrive
Este equipo
USB 512M (D:)
poo
Red

Carpeta: dev

Select the project location   Cancelar

---

poo

Input a Java project name (Press 'Enter' to confirm or 'Escape' to cancel)

File Edit Selection View Go

EXPLORER ...

∨ POO
  > .vscode
  > lib
  ∨ src
    J App.java
  ⓘ README.md

J App.java ✕

src > J App.java > ⌗ App > ⊘ main(String[])
```java
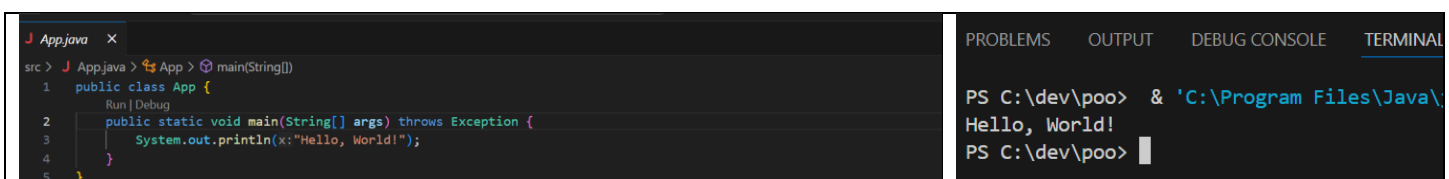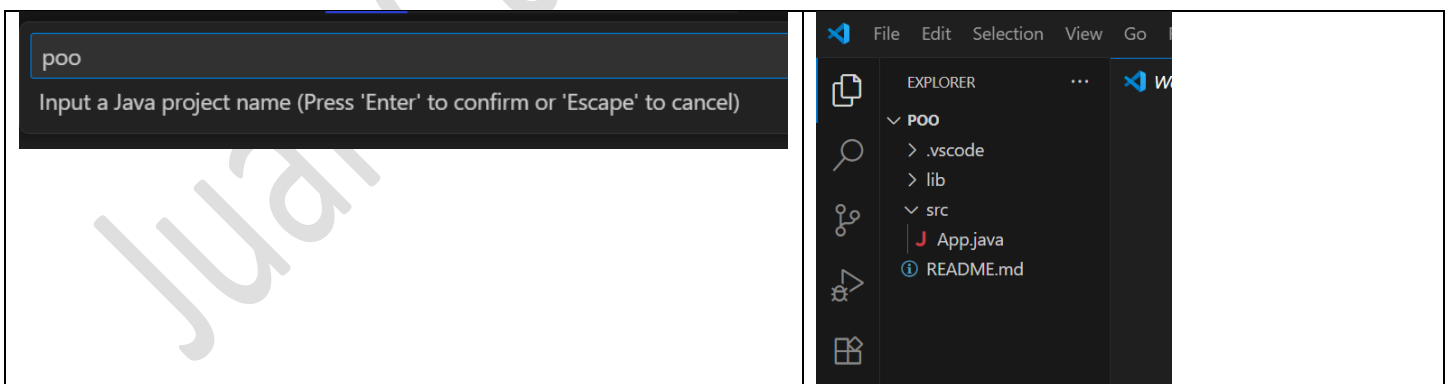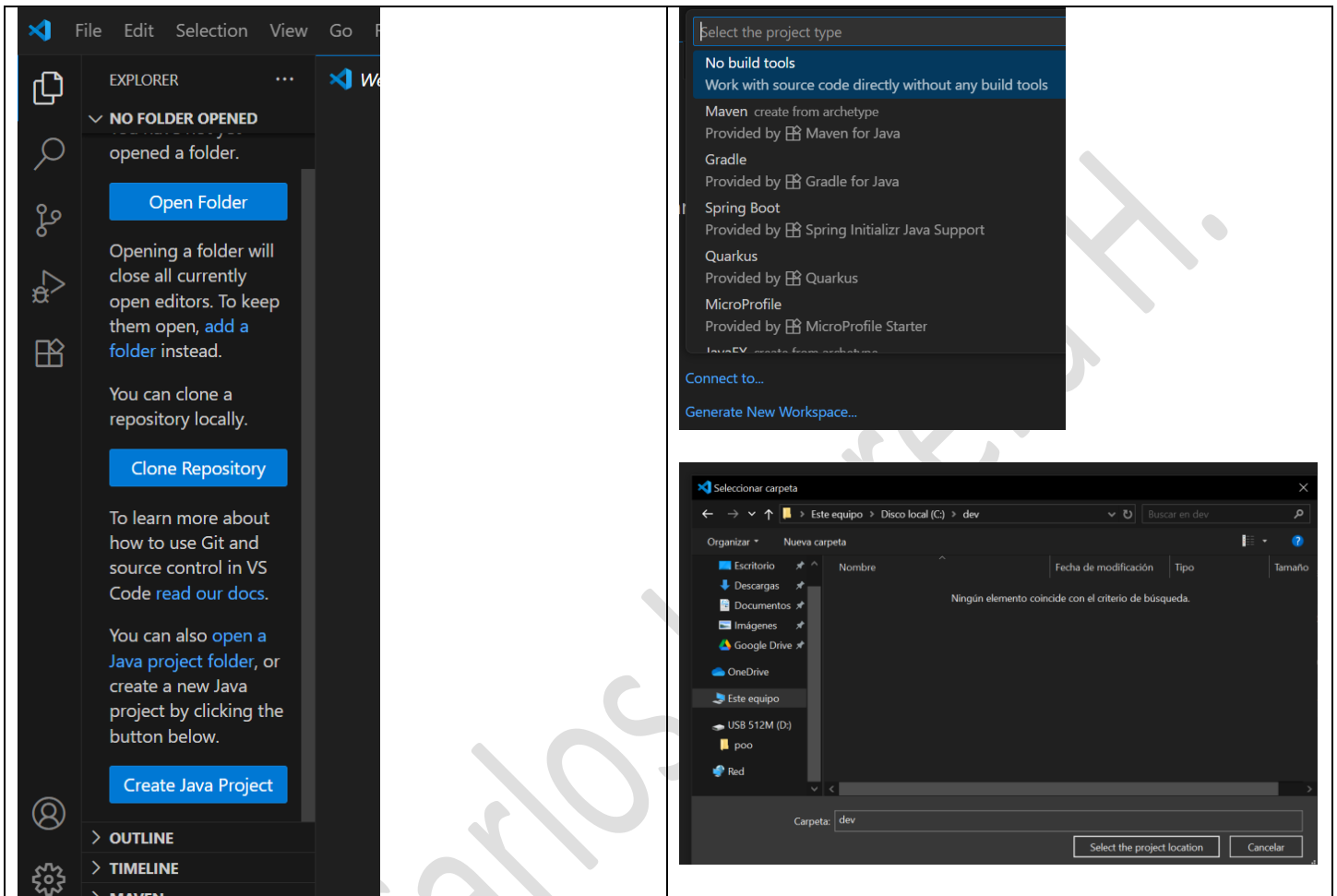1   public class App {
        Run | Debug
2       public static void main(String[] args) throws Exception {
3           System.out.println(x:"Hello, World!");
4       }
5   }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\dev\poo> & 'C:\Program Files\Java\
Hello, World!
PS C:\dev\poo>

→

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

Target Device Types

ORACLE Academy
JFo 1-2
A Brief History
Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Java is Platform-Independent	Java Programs Run in a JVM

Java Runtime Environment (JRE)	Java Development Kit (JDK)
Includes:	Includes:
• The Java Virtual Machine (JVM)	• JRE Java Compiler
• Java class libraries	• Additional tools
Purpose:	Purpose:
• Read bytecode (.class)	Compile bytecode (.java ⮕.class)
• Run the same bytecode anywhere with a JVM	

Compiling and Running a Java Program

You have an idea for a Java program

Your code is contained in a .java file

The result is a .class file (bytecode)

Your code works!

Write → .java → Compile → .class → Run

A Java IDE

JDK

JRE

ORACLE

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?

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

2.3. Introduction to Object-Oriented Programming Concepts

Procedural languages …	Object-oriented languages…
• Read one line at a time • The C language is procedural	• Read one line at a time • Model objects through code • Emphasize object interaction • Allow interaction without a prescribed order • Java and C++ are object-oriented languages

Object-Oriented Programming • Interaction of objects • No prescribed sequence	

Modeling Properties and Behaviors

Quiz: JFo - Section 2   Questions 15

→

3. Java Data Types

3.1. What is a Variable?

```
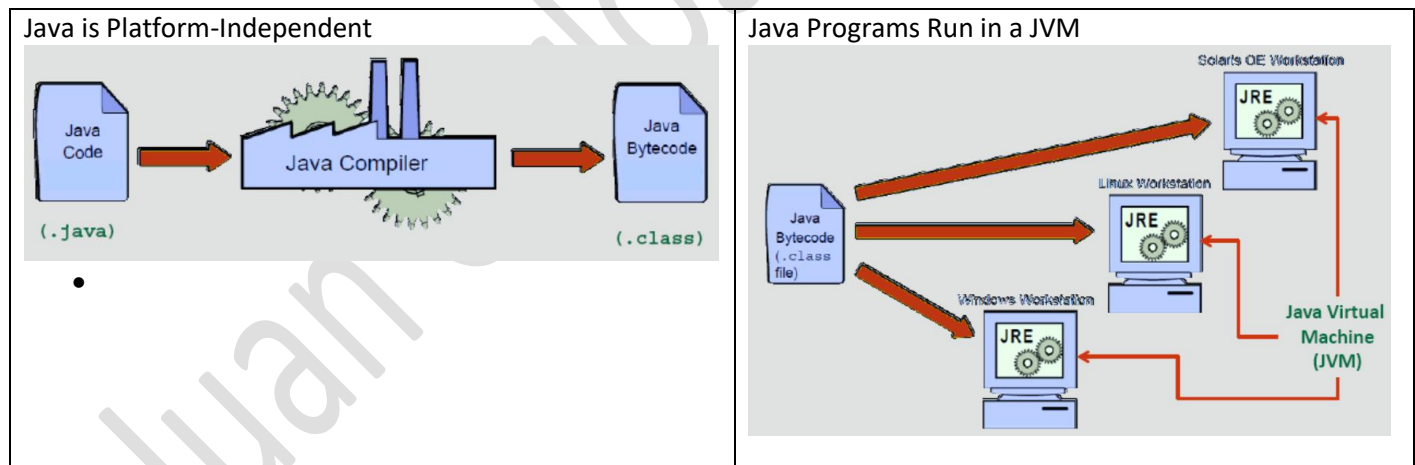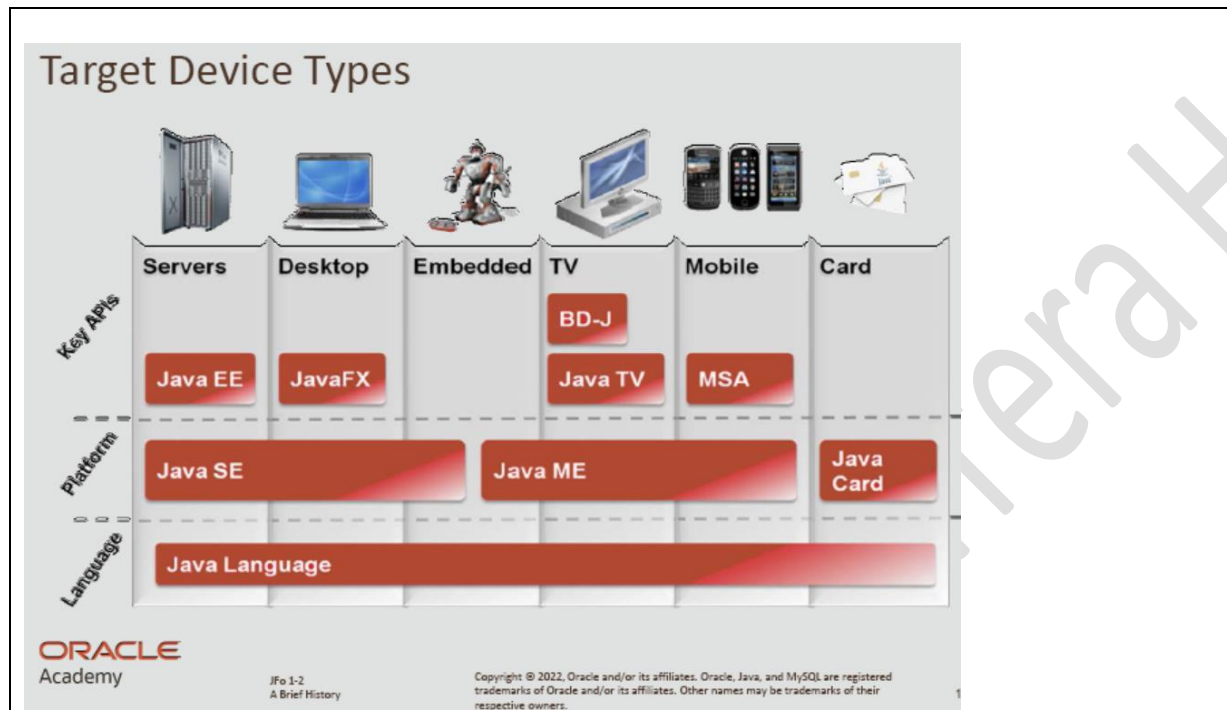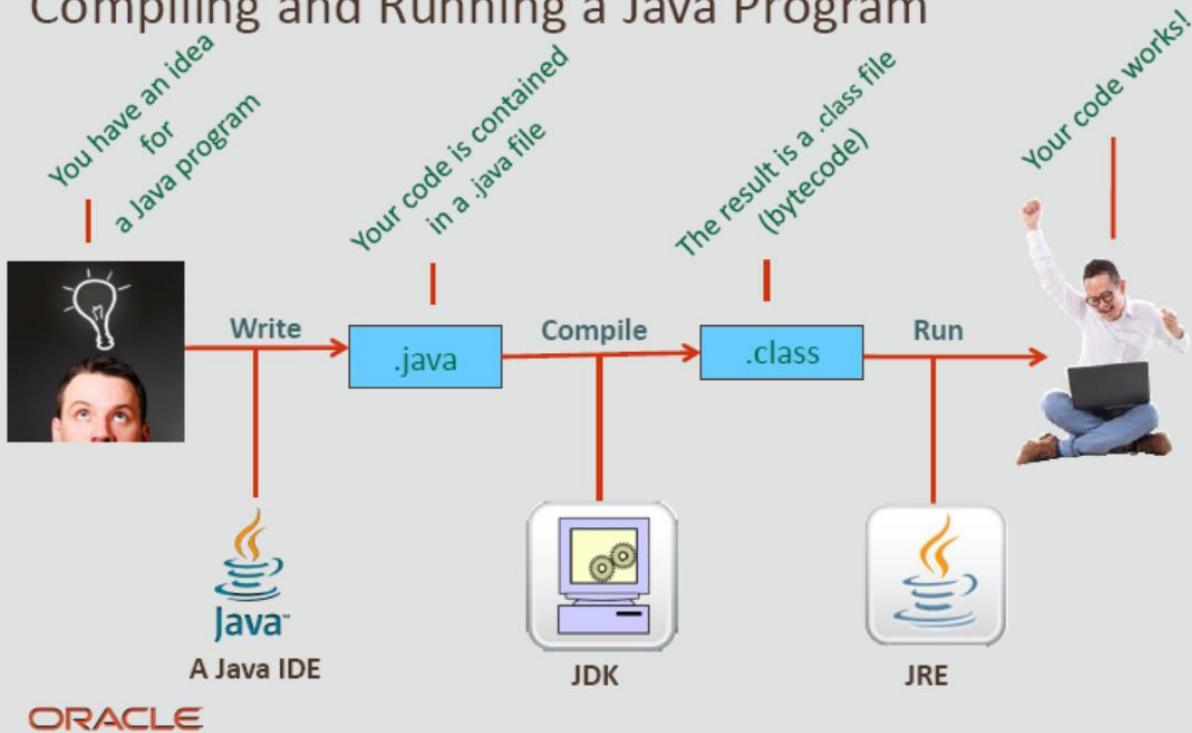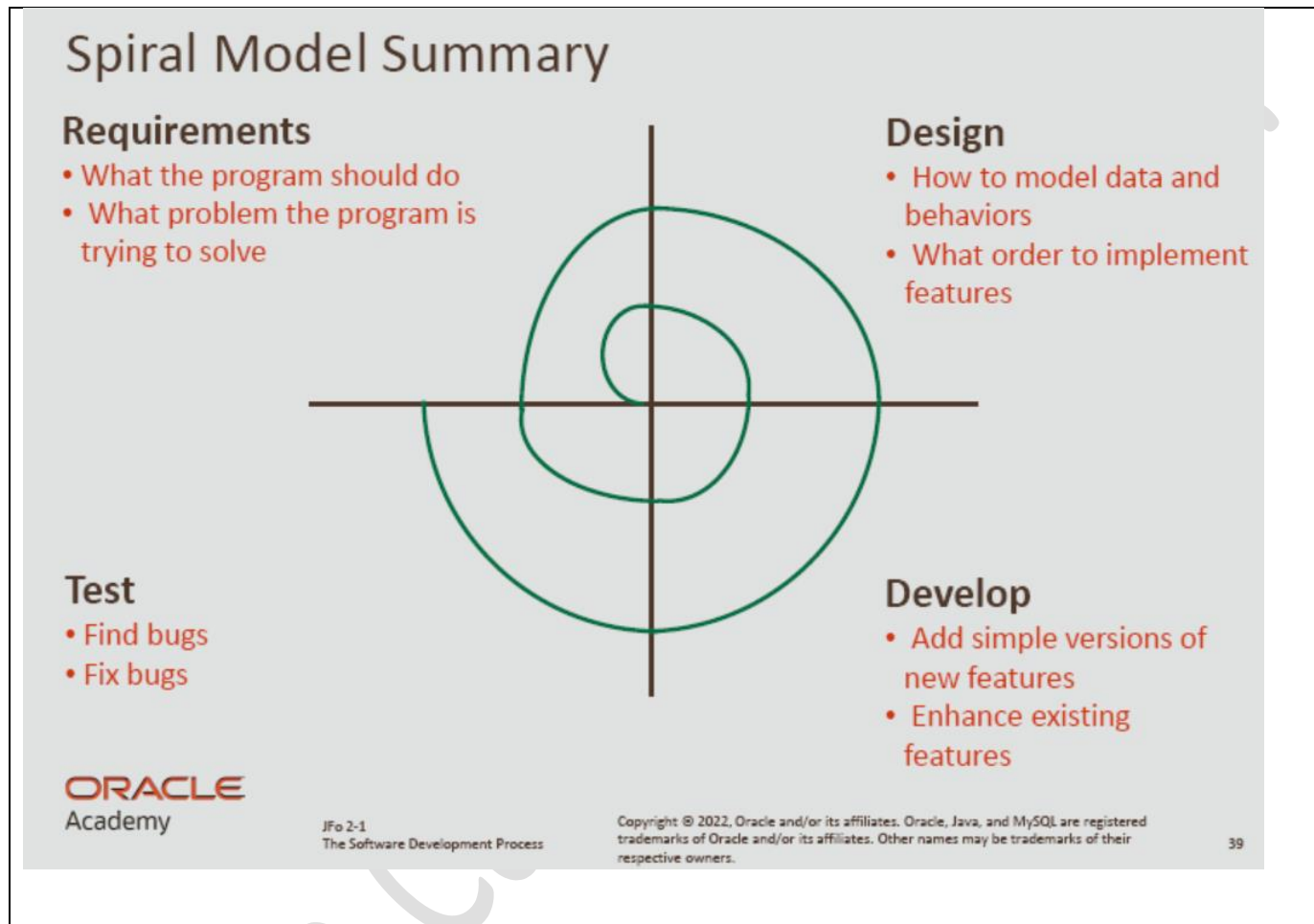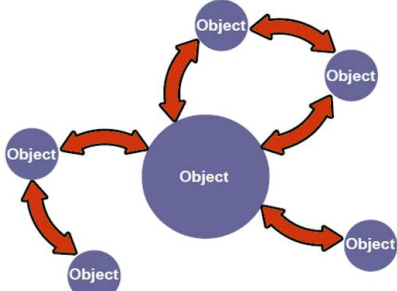String x ="Sam";
System.out.println("My name is " + x);                    Variables03.java (There are 6 mistakes)
```

Type	Keyword	Example Values
Boolean	boolean	true, false
Integer	int	1, -10, 20000, 123_456_789
Double	double	1.0, -10.0005, 3.141
String	String	"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
Byte	8 bits	2^8, or… 256	-2^7, or… -128	2^7-1, or… 127
short	16 bits	2^{16}, or… 65,535	-2^{15}, or… $-32,768$	$2^{15}-1$, or… 32,767
int	32 bits	2^{32}, or… 4,294,967,296	-2^{31}, or… $-2,147,483,648$	$2^{31}-1$, or… 2,147,483,647
long	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 )
```

```
// pre y post incremento y decremento

    int players = 0;
    System.out.println("players online: " + players++);
    System.out.println("The value of players is " + players);
    System.out.println("The value of players is now " + ++players);
    System.out.println("The value of players is " + players);
```

Floating Point Primitive Types

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

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

final double PI = 3.141592;

Final variable naming conventions:

- Capitalize every letter
- Separate words with an underscore
        MINIMUM_AGE

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!"; // Hard-coding

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

Where are Strings?

String is capitalized
- Strings are an object, not a primitive
- Object types are capitalized by convention

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	Description
\t	Insert a new tab
\b	Insert a backspace
\n	Insert a new line
\r	Insert a carriage return
\f	Insert a formfeed
\'	Insert a single quote character
\"	Insert a double quote character
\\	Insert a backslash character

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

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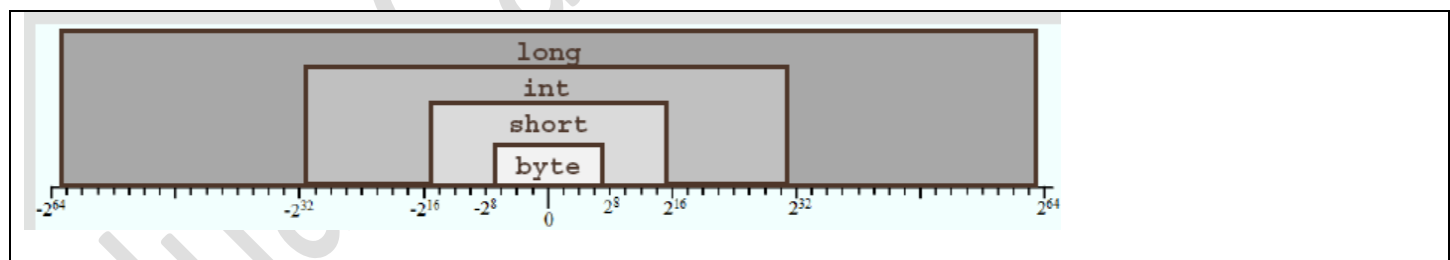System.out.println("This is the first line."
            + "This is NOT the second line.");
```
sout tab   "Metodo abreviado"

3.4. Converting Between Data Types

```
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
```

```
int    num1 = 7;
double num2 = 2;
double num3;
num3 = num1 / num2; // num3 is 3.5
```

- **Automatic promotions:**
  - If you assign a smaller type to a larger type:

byte	→	short	→	int	→	long

  - If you assign an integral value to a floating-point type:

4	→	4,0

- **Examples of automatic promotions:**
  - `long intToLong = 6;`
  - `double intToDouble = 4;`

- **When to cast:**
  - If you assign a larger type to a smaller type:

byte	←	short	←	int	←	long

  - If you assign a floating point type to an integral type:

3	←	3,0

- **Examples of casting:**
  - `int longToInt = (int)20L;`
  - `short doubleToShort = (short)3.0;`

```
double pi = 3.1416
int entero = (int) pi
```

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

- **Solution using larger data type:**

```java
int num1 = 53;
int num2 = 47;
int num3;          ── Changed from byte to int
num3 = (num1 + num2);
```

- **Solution using casting:**

```java
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
```

Automatic Promotion

- **Example of a potential problem:**

```java
short a, b, c;
a = 1 ;
b = 2 ;        a and b are automatically promoted to integers
c = a + b ; //compiler error
```

- **Example of potential solutions:**
  - Declare c as an `int` type in the original declaration:
    - `int c;`
  - Type cast the (a+b) result in the assignment line:
    - `c = (short)(a+b);`

```
int x = 123_456_789;
int x = 123456789;

intintVar1 = Integer.parseInt("100");
doubledoubleVar2 = Double.parseDouble("2.72");
```

→

```
System.out.println("\033[H\033[2J"); // limpiar pantalla

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

Entrada ✕

? Type a number:
5

Aceptar   Cancelar

Mensaje ✕

ⓘ El doble es: 10

Aceptar
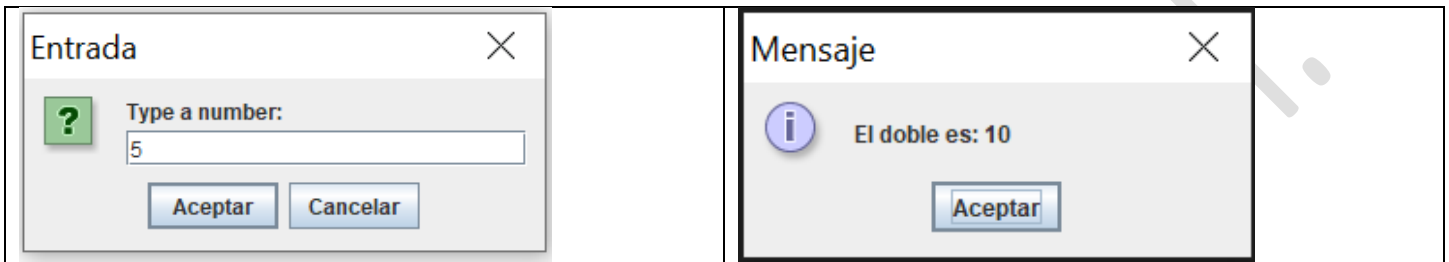
The Scanner searches for tokens

A few useful Scanner methods …
- nextInt() reads the next token as an int
- nextDouble() reads the next token as a double
- next() reads the next token as a String

```
Scanner sc = new Scanner(System.in);
```
The Scanner class considers space as the default delimiter while reading the input

Reading from a File
- nextLine() advances this Scanner past the current line and returns the input that was skipped
- findInLine("StringToFind") Attempts to find the next occurrence of a pattern constructed from the specified String, ignoring delimiters

```
Scanner sc = new Scanner(MyClase.class.getResourceAsStream("texto.txt"));
```

```
Scanner sc = new Scanner(System.in);
int    x = sc.nextInt();
double y = sc.nextDouble();
String z = sc.next();
String linea = sc.nextLine();
int numero = Integer.parseInt(z);
sc.close();
```

Quiz 1: JFo - Section 3 - L1-L2
Quiz 2: JFo - Section 3 - L3-L5
→

4. Java Methods and Library Classes

4.1. hola

Instantiate an object

These classes outline objects' ...
Properties(fields)
Behaviors(methods)
	Method name
	Parameters
	Method return type
	
	``java
	public double calculate(int x, double y){
	    double quotient = x/y;
	    return quotient;
	}//end method calculate
	``
	Implementation

```java
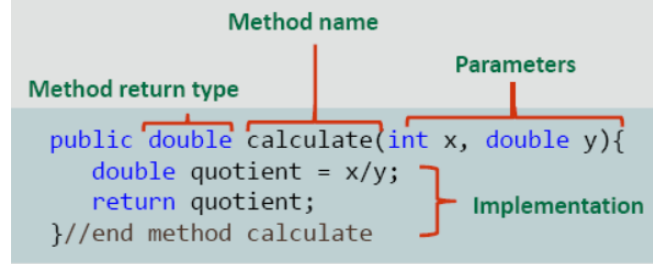double tax = 0.05;
double tip = 0.15;

double person1 = 10;
double total1  = person1*(1 +tax +tip);
System.out.println(total1);

double person2 = 12;
double total2  = person2*(1 +tax +tip);
System.out.println(total2);


 public void findTotal(double price, String name){
    double total = price * (1 + tax + tip);
    System.out.println(name + ": $ " + total);
 } //end method findTotal
```

Method Arguments and Parameters

- An argument is a value that's passed during a method call:

```java
Calculator calc = new Calculator();
calc.calculate(3, 2.0);      //should print 1.5
```
Arguments

- A parameter is a variable that's defined in the method declaration:

```java
                         3        2.0
public void calculate(int x, double y){
    System.out.println(x/y);
}//end method calculate
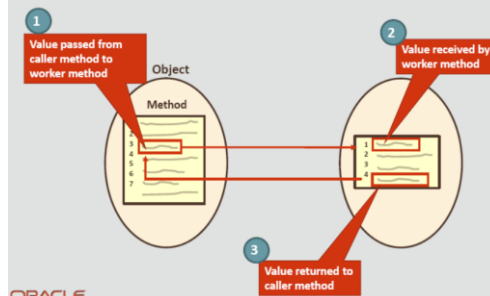```
**Parameters**

## Method Return Types

- Variables can have values of many different types:

  int double long char float byte
  short String boolean Calculator

- Method calls also return values of many different types:

  int double long char float byte
  short String boolean Calculator

## Passing Arguments and Returning Values

① Value passed from caller method to worker method
Object
Method
② Value received by worker method
③ Value returned to caller method
ORACLE

4_2

| |
|---|
| Quiz 1: JFo - Section 4 - L1-L2 |
| Quiz 2: JFo - Section 4 - L3-L5 |

5. Decision Statements

6. Loop Constructs

7. Creating Classes

8. Arrays and Exceptions

9. JavaFX