OBJECT ORIENTED PROGRAMMING
UNIT1: COMPONENTS FROM A COMPUTER PROGRAM

JESUITAS
Sagrado Corazón
Jesuitas · Logroño

Index

Telling the computer what to do

- A computer program (or *software*), is a way to tell the computer what to do.
- Everything you can see on a computer is a program: Windows 8; the warning you see when you get an email; the email itself; etc.

Telling the computer what to do

- A program is equivalent to giving a computer a to-do list (*code*).
- Each of the elements of the list will be the *statements* you wrote in order the computer to do what you want.

```
1 PRINT "Shall we play a game?"
2 INPUT A$
```

Dear personal computer,

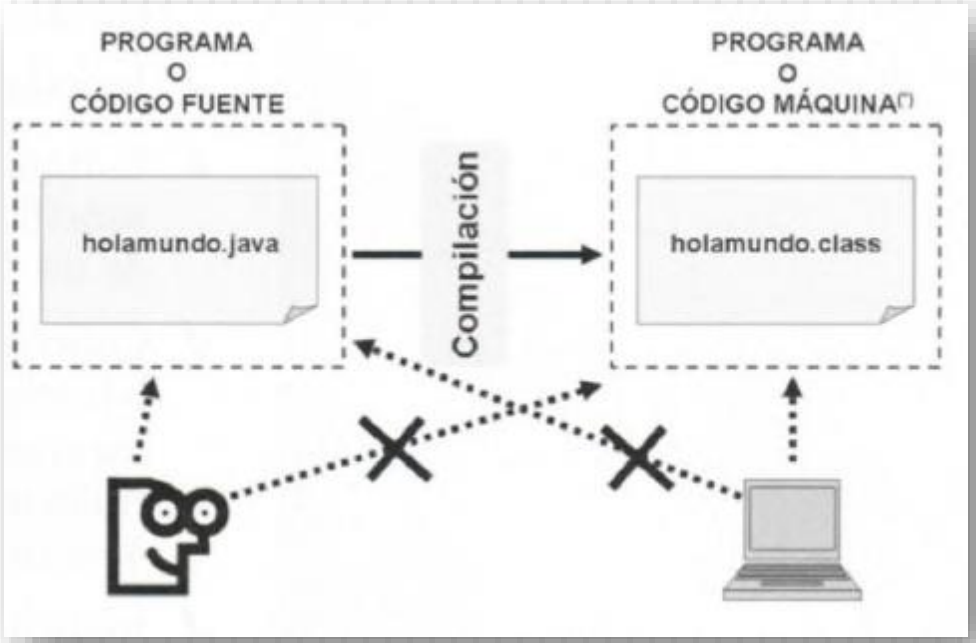
Item 1: Display the question, "Shall we play a game?"

Item 2: Give the user a chance to answer the question.

Love,

Telling the computer what to do

- Apart from that, when we write a program in any editor, it is readable for us but not for the machine.
- Therefore, a translation is required between the code you write, and the language that understands the computer. That work is made by the *compiler*.



Telling the computer what to do

- When we write any code, we can have different kind of errors:
  - *Syntax errors*: These are detected by the computer.
  - *Logic errors*: These, instead, will be detected by the programmer while *testing* the program.
- The usual names for programming errors are *bugs*.
- The process of fixing errors is called *debugging*.

Telling the computer what to do

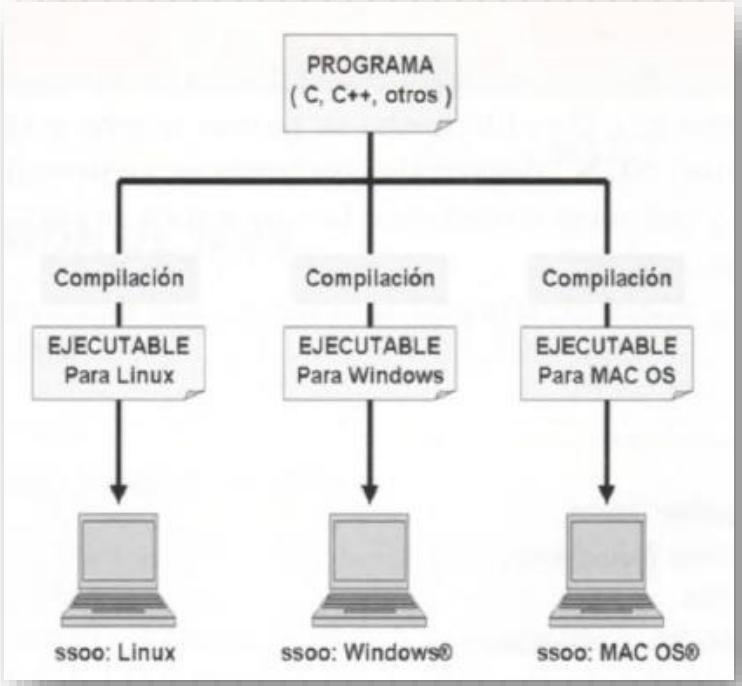
- The usual names for programming errors are *bugs*.
- The process of fixing errors is called *debugging*.

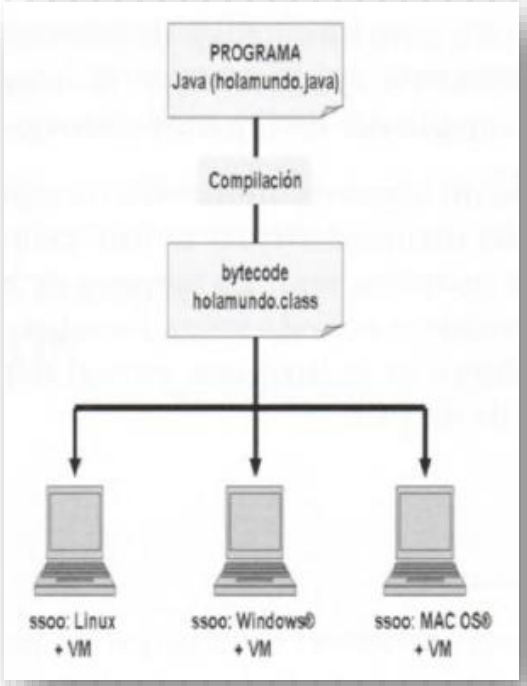
Choosing Java as a language

- We could use any other language in order to learn programming, but *Java* is chosen because:
  - It's used for the *World Wide Web* (WWW). We can implement, for instance, Applets, Servlets or JSP.
  - It's *multi-platform*. This means that it can be executed in different operating systems without the need of recompiling the code, because of the use of a *virtual machine* (VM) in the target system.
  - General purpose language

Choosing Java as a language

- In Java, the result of the process of compiling is called *bytecode*. And once it is understandable by the machine, it also needs and *interpreter* in order to execute that bytecode.
- In the case of Java, VM will make the role of the interpreter.





Before starting with Java

- Before you can start writing Java programs, you need to acquire and set up some kind of Java programming software.
- The Java Development Kit (also referred to as the *JDK*) is (now) in its version 1.8. It contains:
  - Console applications, compilation tools, debugging tools and documentation.
  - It also contains the JRE (Java Runtime Environment) which contains the common libraries and the virtual machine (Download Link, 1.8 documentation).

Before starting with Java

A1.1: Follow next steps in order to install the version 1.8 of JDK:

1. Download JDK 1.8.
2. Install JDK 1.8.
3. Go and inspect the path: C:\Program Files\Java

4. After that you have to change next two environment variables:

PATH: Add the path where the JDK /bin directory is located.

CLASSPATH: Add the path where the JDK classes (/lib) are located.

6. In order to know that it was well done, check the next line in a command shell:

Java -version

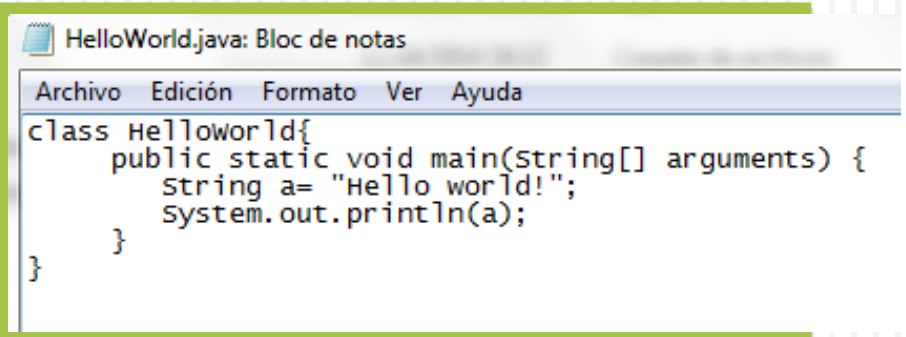
What You Need to Write Programs

- With most programming languages, computer programs are written by entering text into a *word processor* (*text editor*).
- Some programming languages, such as Visual C++ from Microsoft, come with their own "word processor".
- Moreover, we can be find different *integrated development tools* (IDE) that we could use (NetBeans, JBuilder, Geany, etc.).
- However, we will use Eclipse because it is *open source* and widely used.
- Ocassionally, we will also use Bluej environment, in order to understand what means Object Oriented Programming.
- But…if the code is written in plain text (ASCII Text), there is no need to use an IDE.

Writing your first program

A1.2: Follow these steps:

1. Open a new “Notepad”.
2. Copy this statements:




```
class HelloWorld{
    public static void main(String[] arguments) {
        String a= "Hello world!";
        System.out.println(a);
    }
}
```

3. Open a command shell and go to the directory where you put the HelloWorld.java
4. Execute this instruction in order to compile the program. It generates a *.class:

   javac HelloWorld.java
5. Execute the program using: java –cp . HelloWorld

Writing your first program


```
class HelloWorld{
    public static void main(String[] arguments) {
        String a= "Hello wold!";
            System.out.println(a);
    }
}
```

Name of the class=Name of the program=Name of the file *.java

These are called functions or methods.

Means that the program must start from here, from the main.

Writing your first program

A1.3: Now we are going to do the same through Eclipse:

1. Search the eclipse zip file.
2. Unzip and copy the eclipse folder into "C:\Program Files"
3. Go to : "New→Java Project". Project name: LearningJava
4. Right click into "src" folder and "New→class".
5. Write the code.
6. Compile+Execute it.