Introduction to JAVA

CREATED BY :- ANKIT M PATEL.

The EasyLearn Academy

9662512857

Introduction to java

- Java is general purpose, **object oriented programming language**
- It was developed by sun Microsystems of USA in 1991.
- Initially it was called "oak" by its one of the creator **James gosling.**
- It was designed for the software used in electronic home applicences like TV,VCR, toaster and other electronic devices.
- JAVA language is simple , portable, and highly reliable and highly powerful.
- Java is just name and not any acronym. There is no fully form of JAVA.
- Java programmer discovered that there are some limitation of C programming language. and C++
- These limitation is about reliability and portability.
- Java remove above limitations.

Features or advantages of java

- Compiled and interpreted
- Platform independent and portable
- Object Oriented
- Robust and secure
- Distributed
- Familiar, simple and small
- Multithreaded and interactive
- High Performance
- Dynamic and Extensible

Requirements for java

- To develop java program one can use any simple text editor like notepad, notepad++, **visual studio code**.
- One can also use some modern IDE like **Jcreator, Eclipse, netbeans** etc for same.
- But to compile and run java program one must have **JDK**.
- Full form of JDK is **java development kit**.
- It is used to compile, debug and run java program.
- Let us see JDK in detail

JDK and its components

- JDK stands for java development kit.
- It is the collection of tools that is used to develop and run the java application
- JDK tools are following
  - **Applet viewer** :- it used to view applet created by a programmer. We can use this tool as replacement of java compatible web browser.
  - **Javac** :- it is java compiler. It convert source code into byte code. This is necessary to run any program. It produce class file
  - **Java** :- it is used to execute java program. It execute class file of java. Java file must be compiled using javac before it run.
  - **Javap** :- this is used to decompile java byte code into java code(program code !!!!). It is disassembler.
  - **Javah** :- this tool is used to create file which can be used later in program
  - **Javadoc** :- it is used to create HTML format documentation from java source code file.
  - **Jdb** :- this tool is known as java debugger. It is used to find error in java program.

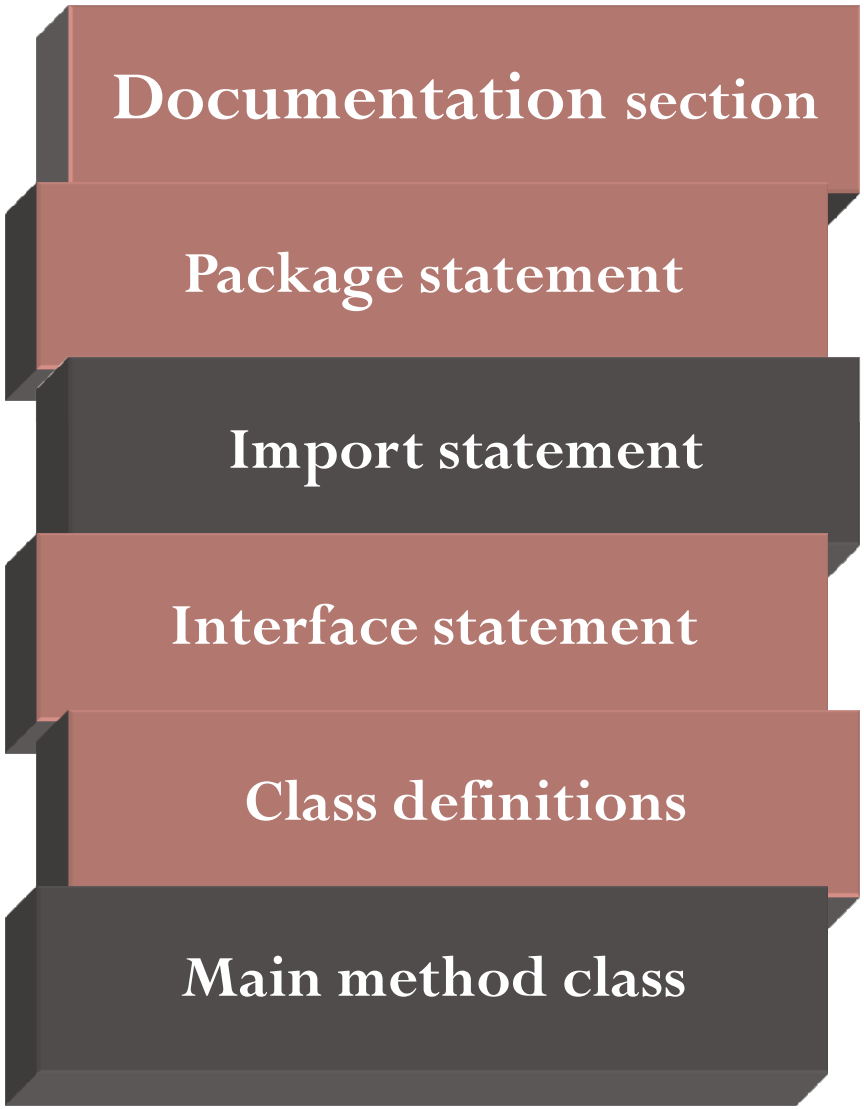
Simple java program

```
public class lesson1{
    public static void main(String arguments[]){
        System.out.println("Hello world, we are learning java");
    }
}
```

- In java first we create class and then we create method in it.
- Method is program in program you can call it small program used to do specific task.
- Each public class has public static void main method. It is the entry point of the program. Means from here program start executing it self.
- We use System.out.println to print message on screen. Also called output.
- Each line in program ends with semicolon

Java program structure

- A simple java program may contains many classes of which only one class defines main methods.
- Classes contain data members and methods that operate on data member of the class.
- Methods may contains data type declarations and executable statements.



Java program structure

- **Documentation section** : include comments lines used to specify name of program, author and other details, which may be useful to programmer. It is optional part.
- **Package statement** :- the first statement allowed in java file is a package statement.
- Package is collection of logically related classes.
- Programmers can define their own packages to bundle group of classes/interfaces,
- This statement declares a package name and informs the compiler that the classes defined here belongs to package.
- **Import statement** :- this statement is used to load given class from specific package.
- This is similar to #include statement in c.

Java program structure

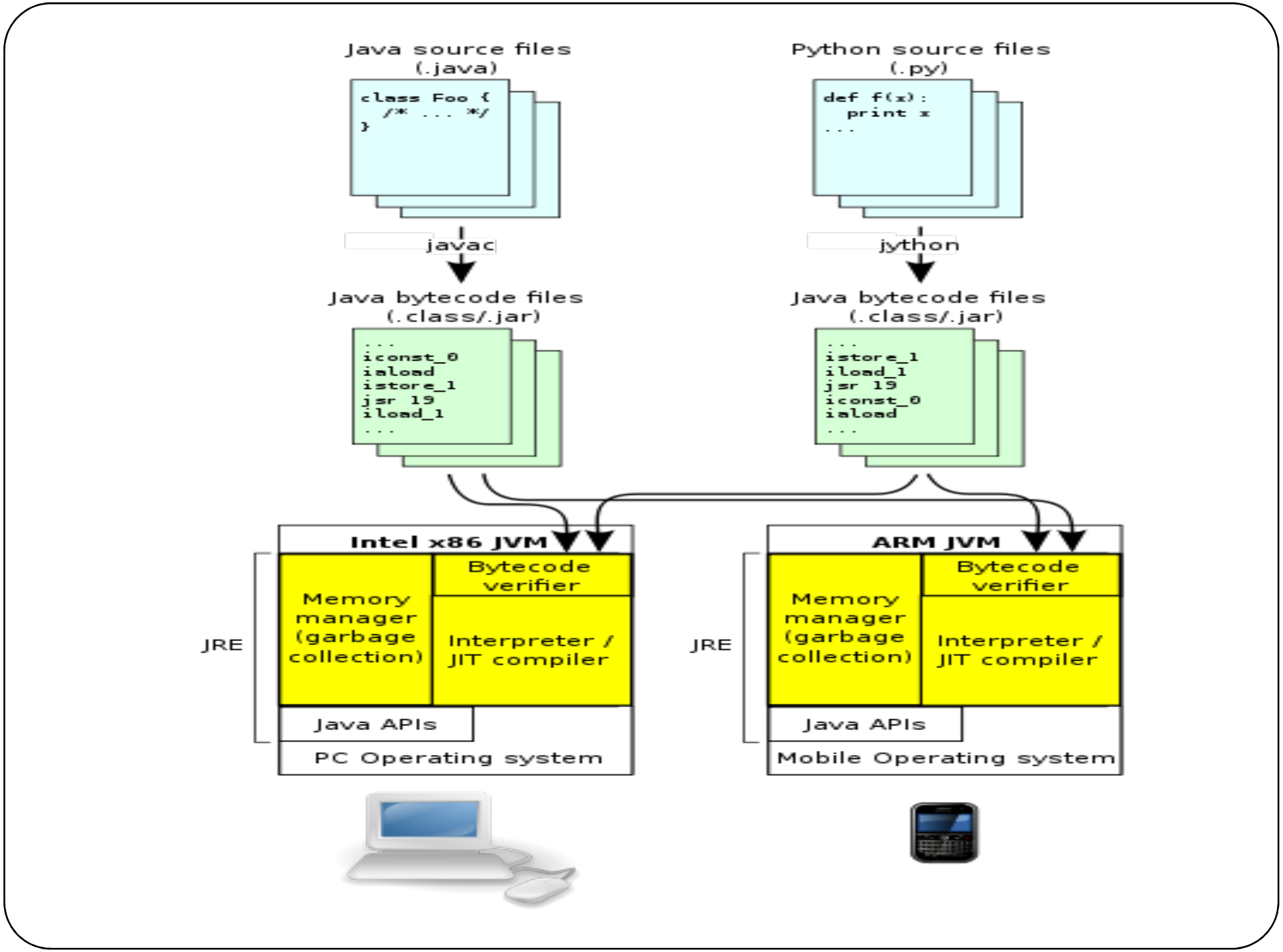
- **Interface statement** :- an interface is like a class but includes a group of methods declarations.
- This is optional section.
- It is used only when we wish to implement multiple inheritance feature in the program.
- **Class definition** :- a java program may contain multiple class definitions.
- Classes are the primary and essential elements of java program.
- One can add any number of class used in a program depends upon the complexity of the program.
- **Main method class** :- main method class contains public static void main method.
- Java program start from this methods
- Name of java program must be same as name of main method class.
- A simple java program may contains only this class.

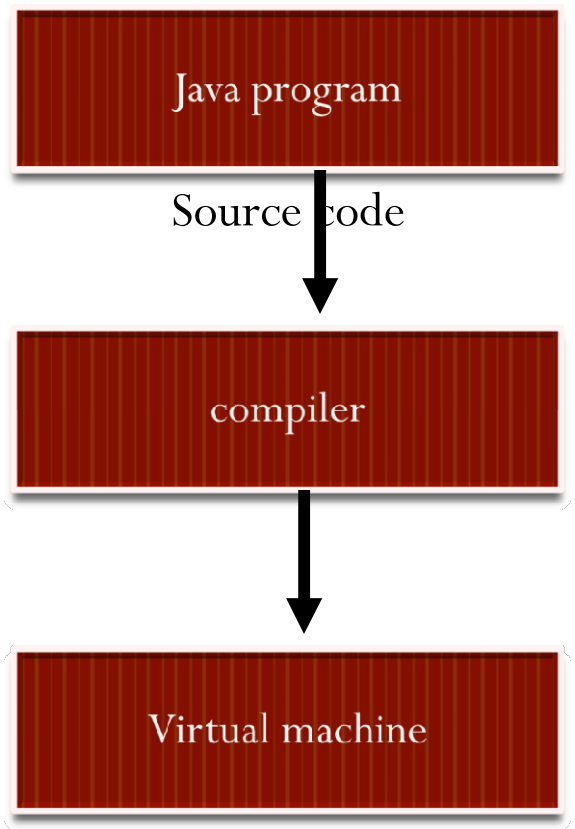
What is byte code? And What is JVM?

- Byte code is specific set of instruction produced by the java compiler from java source code.
- Java compiler compiles the java program and converts the source code into byte code.
- Byte codes are machine independent and are executed on Virtual machine called Java Virtual Machine.
- Java program is highly secure and portable because of the byte-code.
- So now learn what is JVM?

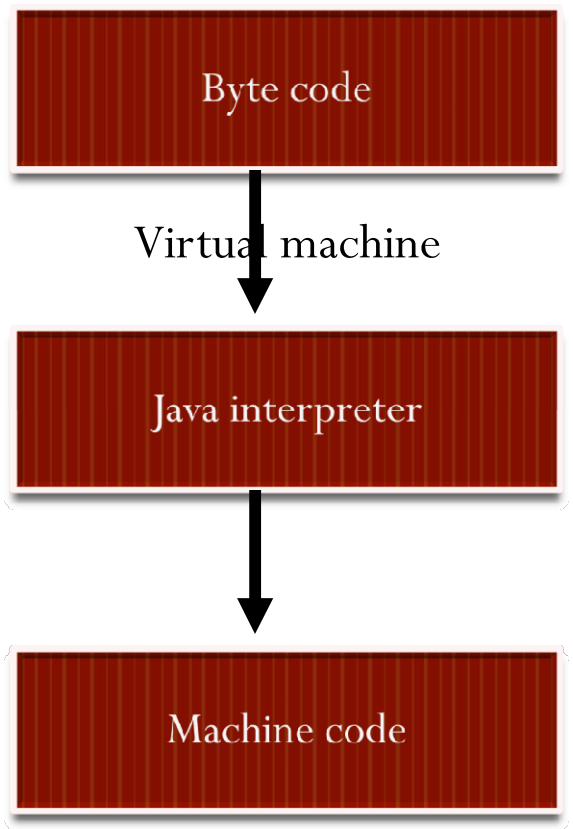
JVM (java virtual machine)

- A Java virtual machine (JVM) interprets compiled Java binary code (called bytecode) for a computer's processor (or "hardware platform") so that it can perform a Java program's instructions.
- James Gosling designed Java to allow programmers to write code that could run on any platform without the need for rewriting or recompilation for each separate platform.
- A Java virtual machine makes this possible because it knows specific instruction lengths and other particularities of the platform.


Java source files
(.java)
class Foo {
/* ... */
}
javac
Java bytecode files
(.class/.jar)
...
iconst_0
iaload
istore_1
jsr 19
iload_1
...
Python source files
(.py)
def f(x):
print x
...
jython
Java bytecode files
(.class/.jar)
...
istore_1
iload_1
jsr 19
iconst_0
iaload
...
Intel x86 JVM
JRE
Memory manager (garbage collection)
Bytecode verifier
Interpreter / JIT compiler
Java APIs
PC Operating system
ARM JVM
JRE
Memory manager (garbage collection)
Bytecode verifier
Interpreter / JIT compiler
Java APIs
Mobile Operating system

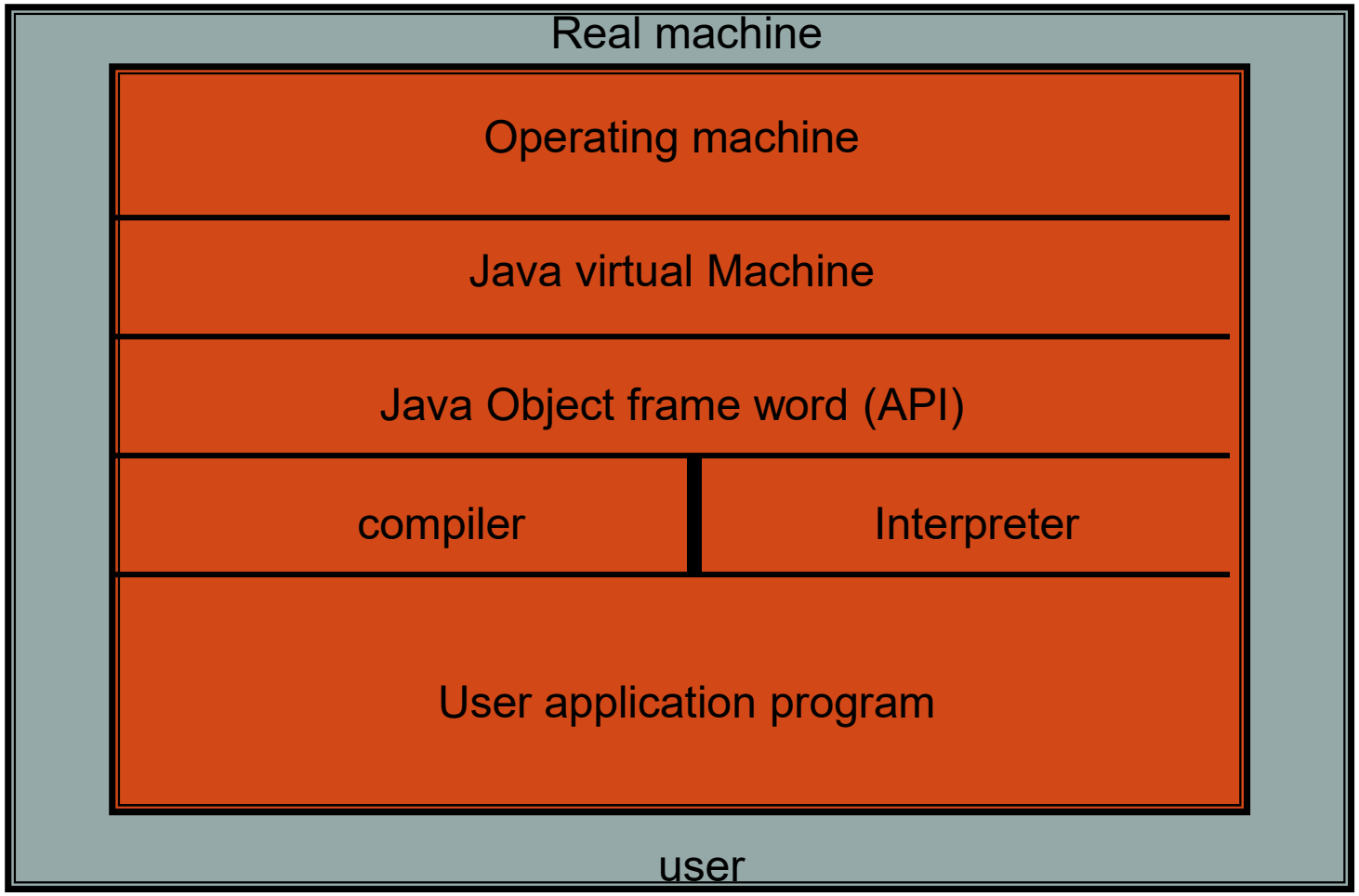


Byte code



Real machine

Layers of interaction of java programs



Now let use see another java program which accept input from user using Library class