INTERPRETED vs. COMPILED LANGUAGES

Interpreter translates one statement at a time into machine code.
Compiler scans the entire program & translates whole of it into machine code.

Interpreter:

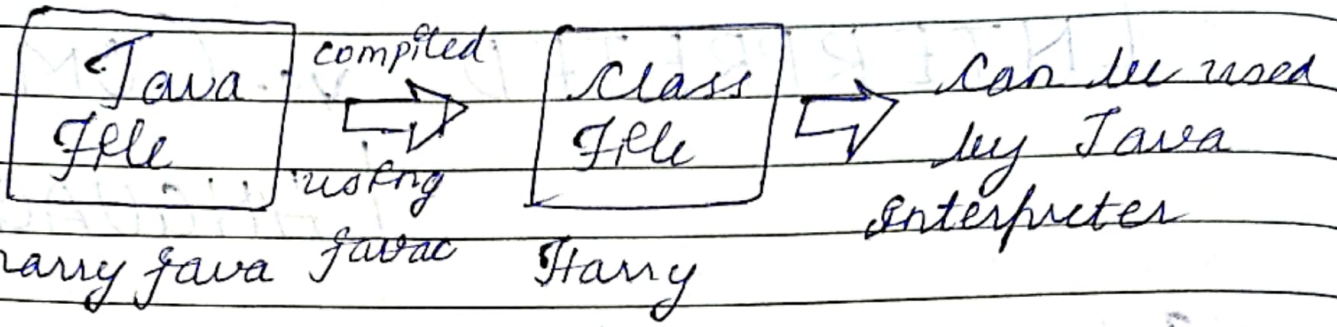
→ One statement at a time
→ An interpreter is needed every time
→ Partial execution if an error occurs in the programs.
→ Easy for programmers

Compiler:

→ Entire program at a time.
→ Once compiled, it's not needed
→ No execution if an error occurs
→ Usually not as easy as interpreted ones

Is Java interpreted or compiled?

Java is a hybrid language both compiled & interpreted



Harry.java → Harry.class

→ A JVM can be used to interpret the byte code.
→ The byte code can be taken to any platform (Win / Mac / Linux) for execution.
→ Hence Java is platform independent. (Write once run everywhere)

Executing a Java program

javac HelloWorld.java
java HelloWorld

PACKAGES IN JAVA

→ A package is used to group related classes.
→ Packages help in avoiding name conflicts.

There are 2 types of packages

→ Built in packages - Java API
→ User defined packages - Custom packages

songs, photos, videos → Org. as folder → song, photos videos.

Java classes → Org. as packages → classes

Using Java packages

import java.lang.*; → import everything from java.lang

import java.lang.string; → import string from java.lang

s = new java.lang.string("Harry"); → Use without importing

How to create packages in Java?

~~javac -d~~

javac Harry.java → creates a Harry.class

~~javac -d Harry.java~~ → creates a package

javac -d . Harry.java folder

Blank spaces

We can keep adding package files like this

We can also create inner packages by adding "package.inner" as package name.

↓ folder ↳ subfolder

These packages once created can be used by other classes.

Access modifiers in Java

Access modifiers determine whether other classes can use a particular field or invoke a particular method. can be public, private, protected or default (no modifier.

Modifier	Class	Package	Subclass	World
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
Default (no)	Y	Y	N	N
Private	Y	N	N	N

Static Keyword

→ Static keyword is applicable for variables as well as methods.
→ It is used with an instance variable of a class or with a method of a class.
→ The main method is always declared as static.
→ The static can be:
1) variable 2) method
3) block 4) nested class

→ Static variables are available for all the objects of the class
→ These variables are generally maintained to store values common to the class.
E.g. Each student has distinct Id but common college name.

How are they different from other variables?
→ There is only one copy of data which is shared by all the objects of the class.
→ Static data members are visible within the class but it remains in the memory.
→ It saves memory.

Static Methods

→ If you apply static keyword with any method, it is known as static method
- A static method belongs to the class rather than object of the class.
- A static method can be invoked without the need for creating an instance of the class
- Static method can access static data members & can change value of it.

Restrictions for static method

→ Static method cannot use non-static data members or call non-static method directly.
→ this & super cannot be used in static context

for e.g.

```
class A{
    int a=40;

p s v m (    ){
    sout(a);
    }
}
```

o/p: Compile time error

Java static block

→ It is used to initialize the static data member
→ It is executed before the main method at the time of class loading.

```
class A{
    static{sout("static block invoked");}

    p s v m (    ){
        sout("Hello");
    }
}
```

o/p: static block is invoked
Hello

Strings

Java.lang.String class is used to create string object.

→ String objects are stored in a special memory area known as string const. pool.
→ Java uses this to make it more memory efficient (because no new objects are created if it exists already in string const. pool)