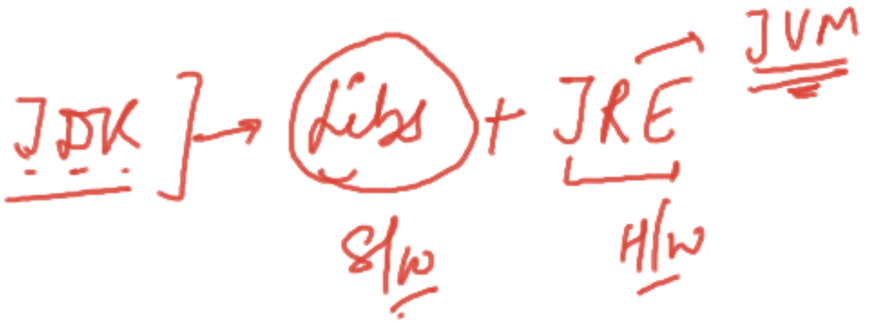
There are many java versions that has been released. Current stable release of Java is Java SE 8.

JDK Alpha and Beta (1995)
JDK 1.0 (23rd Jan, 1996)
JDK 1.1 (19th Feb, 1997)
J2SE 1.2 (8th Dec, 1998)
J2SE 1.3 (8th May, 2000)
J2SE 1.4 (6th Feb, 2002)
J2SE 5.0 (30th Sep, 2004)
Java SE 6 (11th Dec, 2006)
Java SE 7 (28th July, 2011)
Java SE 8 (18th March, 2014)



Java technology is both a programming language and a platform.

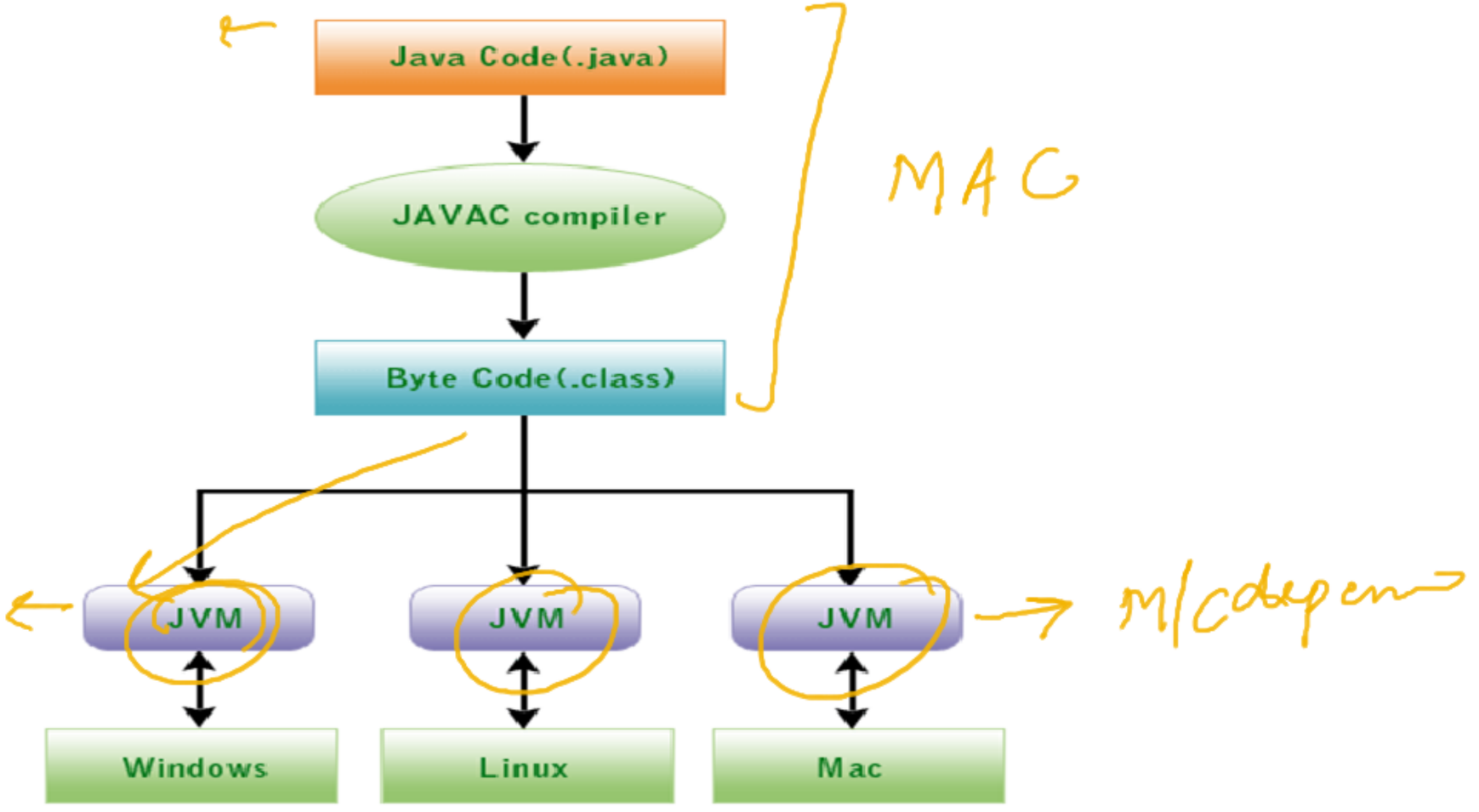
The Java programming language is a high-level language that can be characterized by following buzzwords,

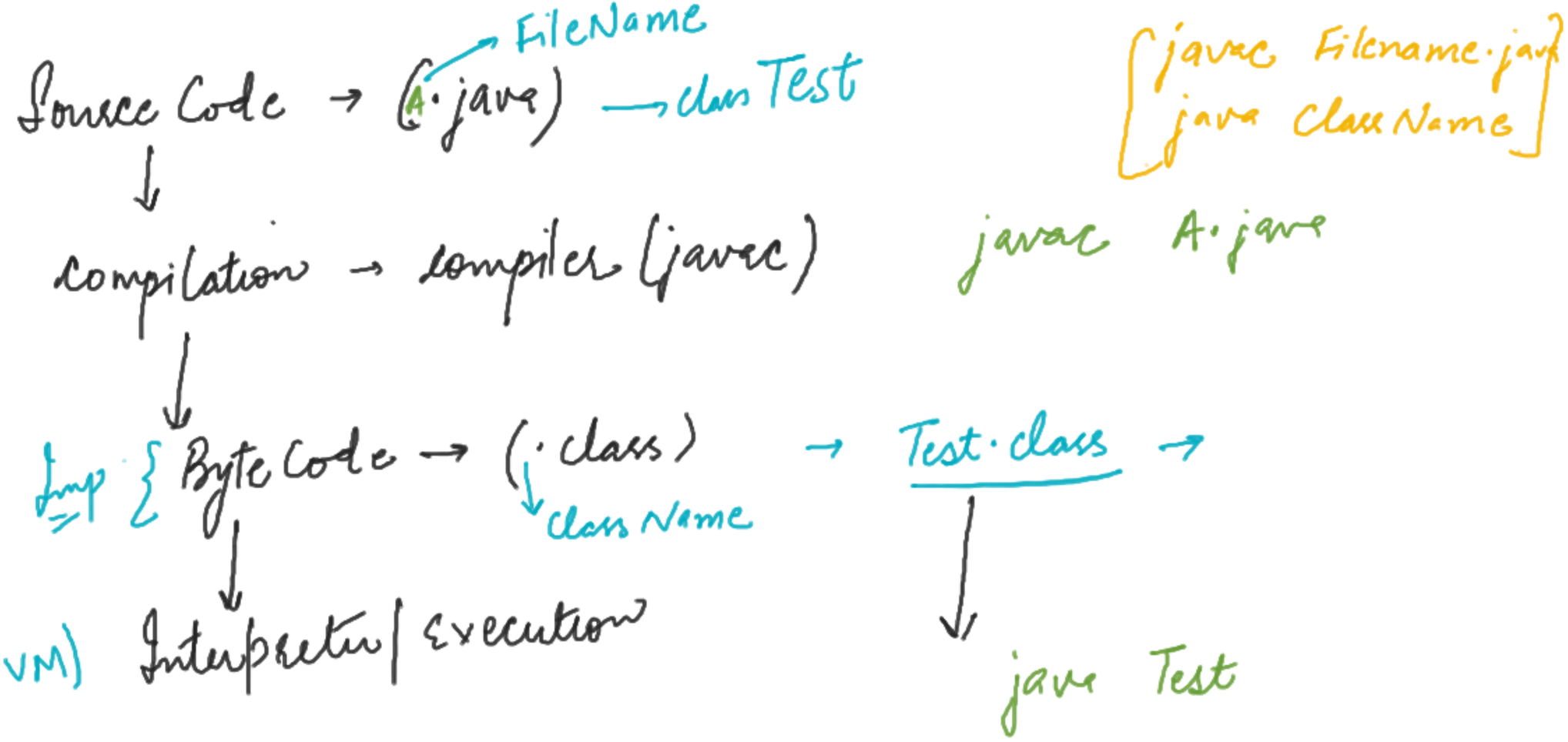
- Simple
- Object-Oriented
- Platform independent
- Portable → No pointer | oops | Byte Code | Exception Handling | Typecasting | Arrays Out of Bound
- High Performance
- Secured
- Robust
- Multithreaded
- Architecture Neutral

int → 4 Bytes
long → 8 Bytes

JVM - M/C dependent

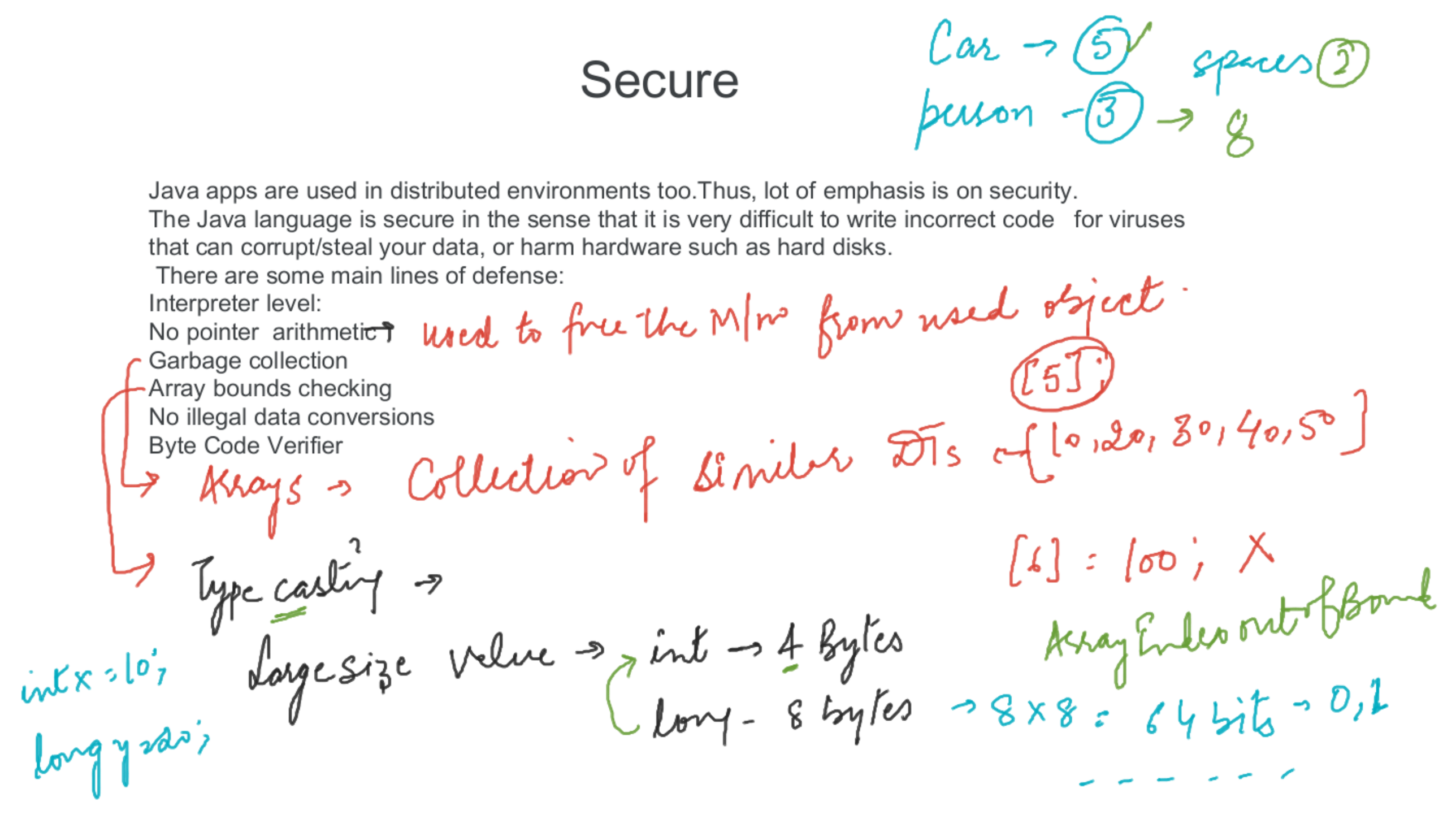
JAVA - M/C Independent




FileName
Source Code → (A.java) → class Test
[javac Filename.java
java ClassName]
compilation → compiler (javac)
javac A.java
Jvm { Byte Code → (.class) → Test.class →
className
(JVM) Interpreter | execution
java Test

Secure

Java apps are used in distributed environments too.Thus, lot of emphasis is on security.
The Java language is secure in the sense that it is very difficult to write incorrect code for viruses that can corrupt/steal your data, or harm hardware such as hard disks.
There are some main lines of defense:
Interpreter level:
No pointer arithmetic
Garbage collection
Array bounds checking
No illegal data conversions
Byte Code Verifier

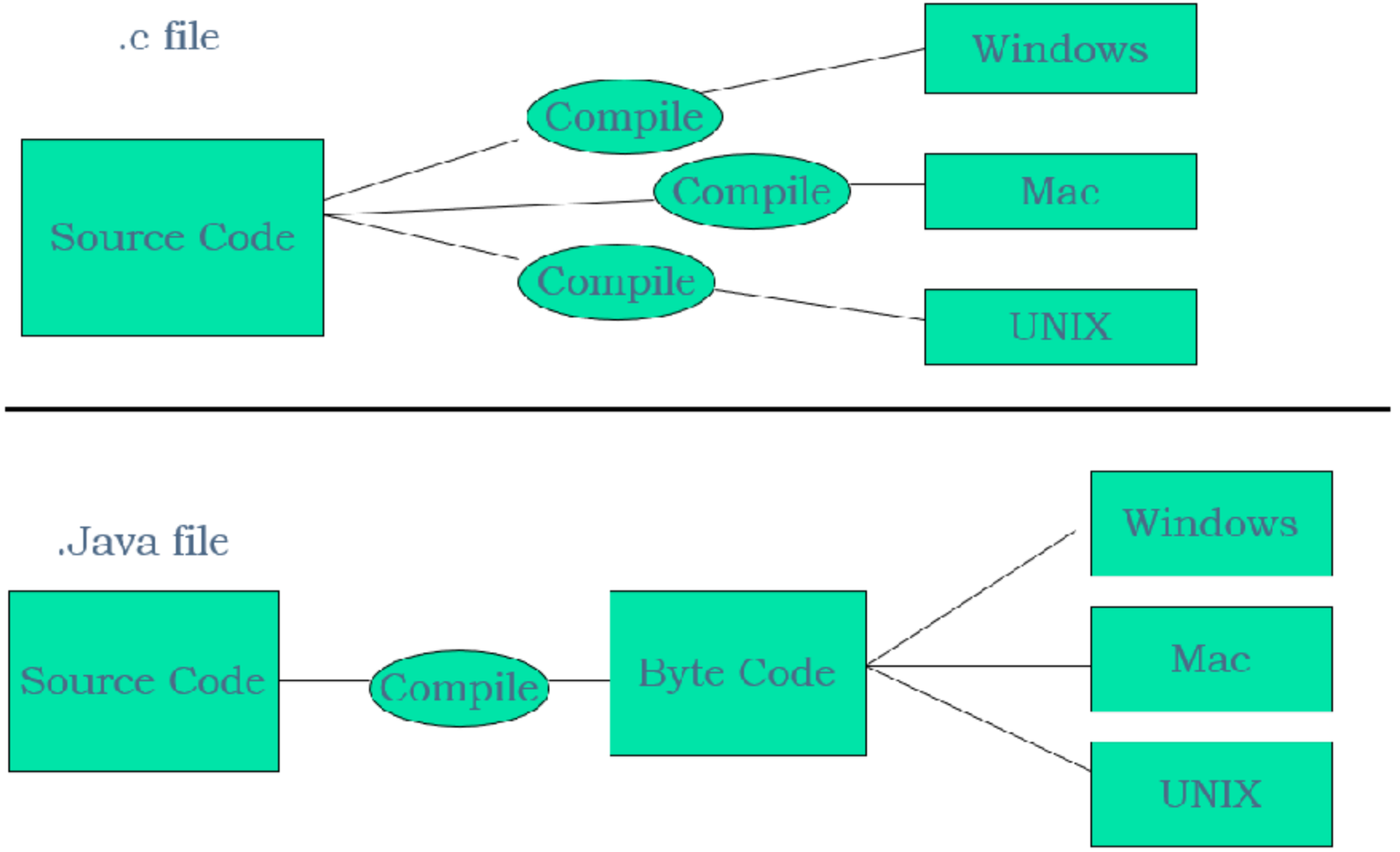


Robust

Reliable
Early checking for potential problems.
Dynamic checking to eliminate error-prone situations.
Developer doesn't have to worry about
Bad pointers
Memory allocation errors

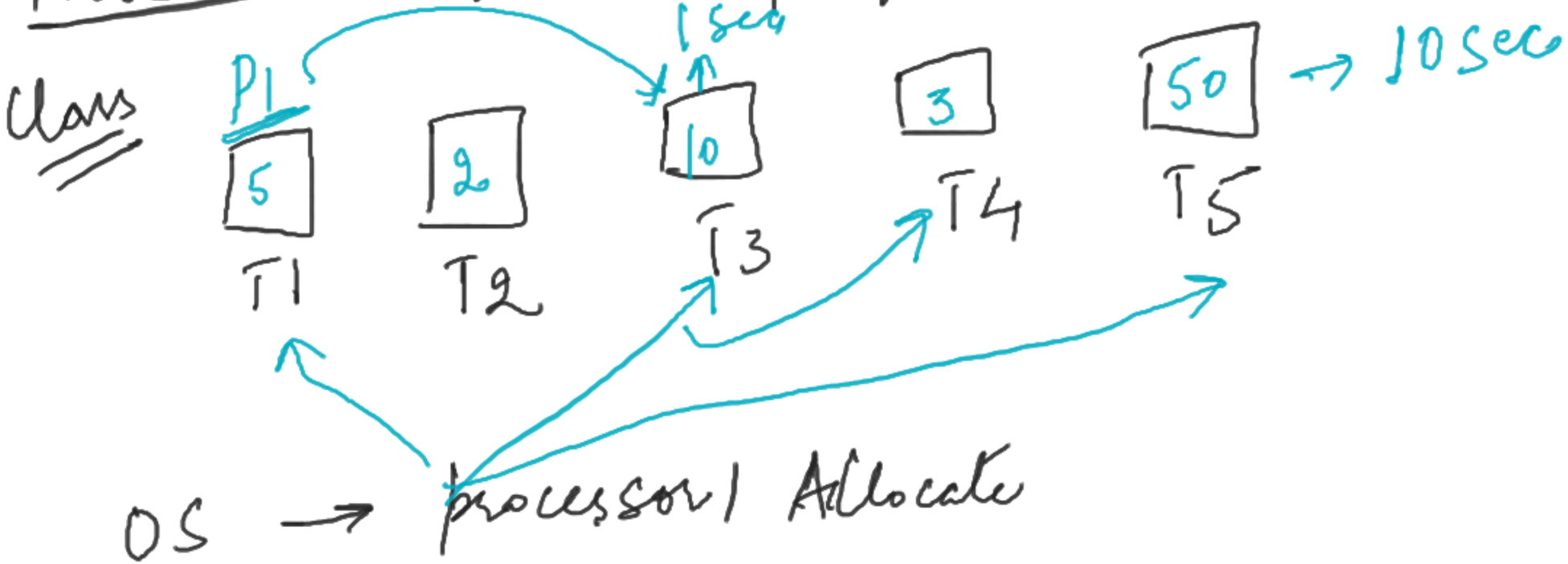
→ compile time
→ Run time / Interpreter level

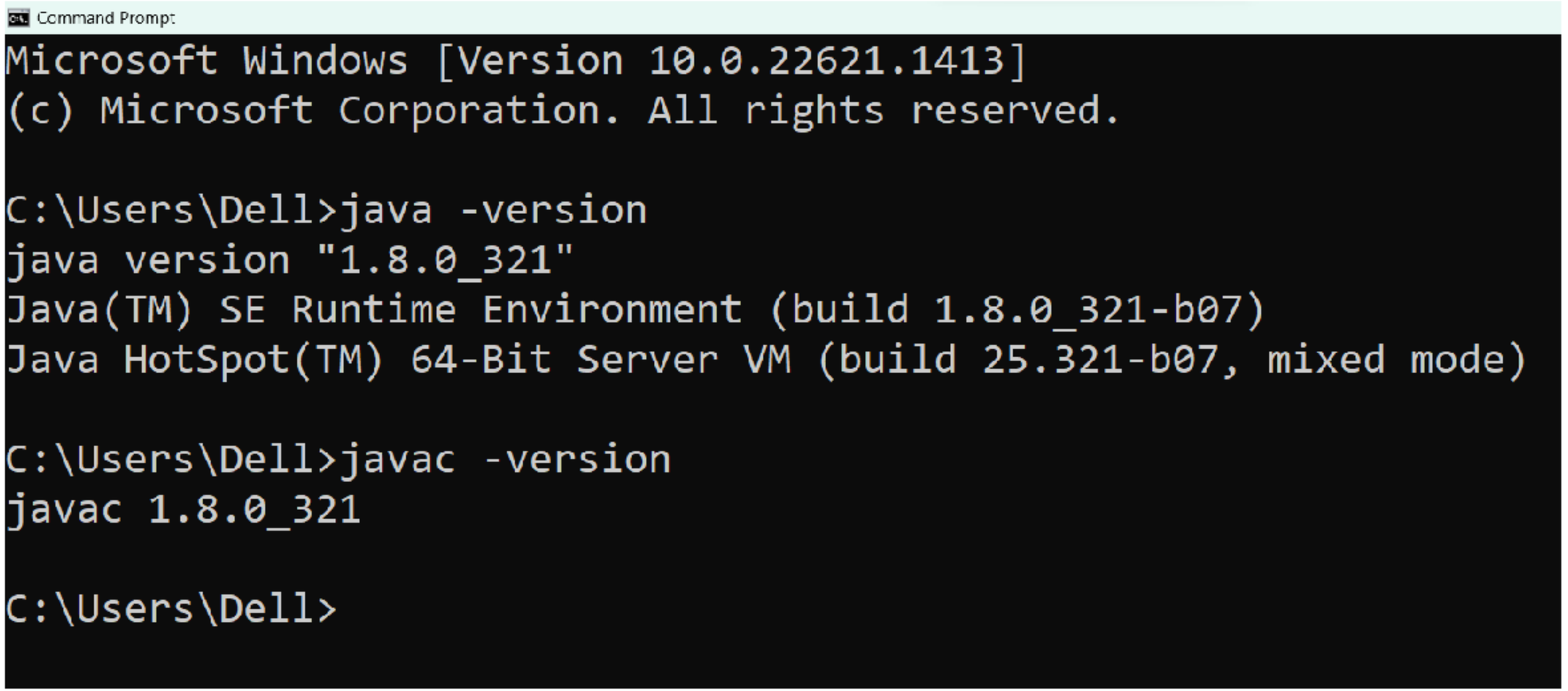
Java Feature:Architecture Neutral

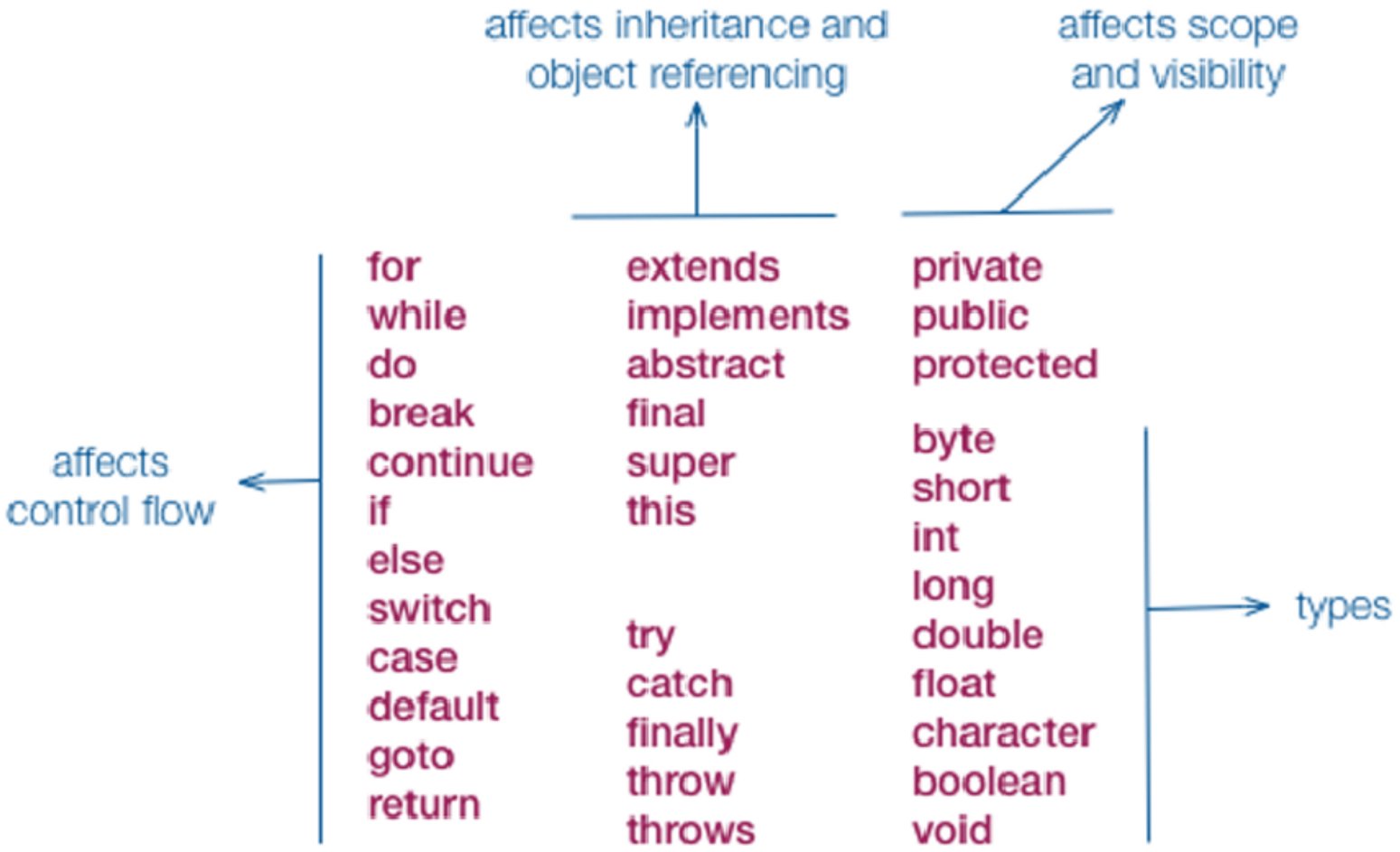


Multithreading → OS

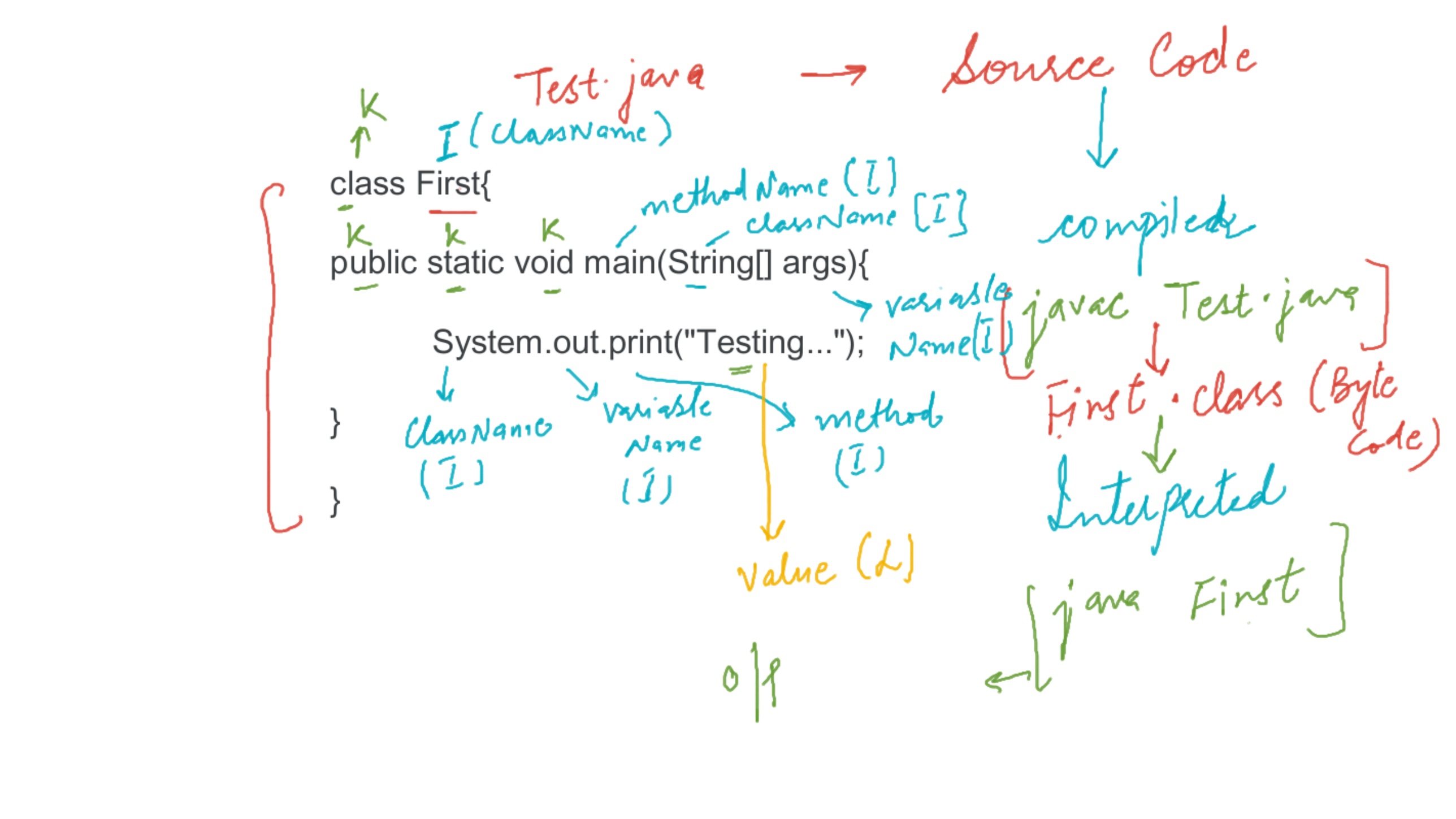
Thread → small program / process




Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Dell>java -version
java version "1.8.0_321"
Java(TM) SE Runtime Environment (build 1.8.0_321-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.321-b07, mixed mode)
C:\Users\Dell>javac -version
javac 1.8.0_321
C:\Users\Dell>



static package assert volatile synchronized
strictfp import enum native transient


Test.java → Source Code
K
I (className)
class First{
K k K
methodName (I)
className (I)
public static void main(String[] args){
variable Name (I)
System.out.print("Testing...");
className (I)
variable Name (I)
method (I)
value (L)
}
}
compiled
javac Test.java
First.class (Byte Code)
Interpreted
java First
0 1

(1) Keyword → Reserved words

(2) Identifier → which identifies something → It could be className, method name, etc variable name

(3) Literal → value

↓

[true, false, null] → predefined

Data Types (i) Primitive DT → Store single value. a = 10;

Type Name	Kind of Value	Memory Used	Range of Values
byte	Integer	1 byte	–128 to 127
short	Integer	2 bytes	–32,768 to 32,767
int	Integer	4 bytes	–2,147,483,648 to 2,147,483,647
long	Integer	8 bytes	–9,223,372,036,8547,75,808 to 9,223,372,036,854,775,807
float	Floating-point	4 bytes	$\pm 3.40282347 \times 10^{+38}$ to $\pm 1.40239846 \times 10^{-45}$
double	Floating-point	8 bytes	$\pm 1.7976931348623157 \times 10^{+308}$ to $\pm 4.9406564584124654 \times 10^{-324}$
char	Single character (Unicode)	2 bytes	All Unicode values from 0 to 65,535
boolean		1 bit	True or false

(ii) Non-Primitive Data Type → Multiple values Store
String / Array / Classes

int [ ] arr = { 10, 20, 30, 40 }

String x = "Test"

↓

" "

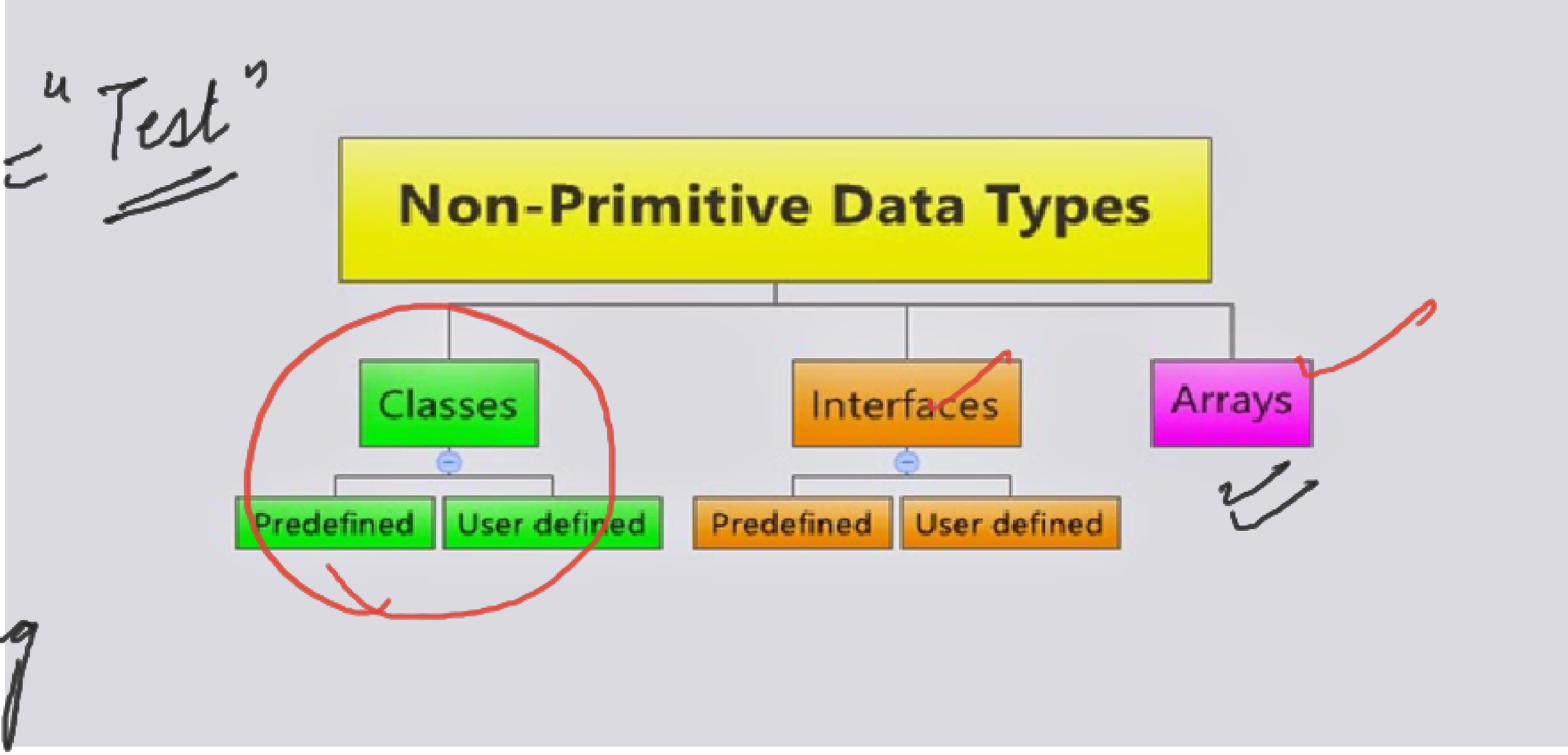
↓

string

' '

↓

character



Variable :: Storage / Container to store something

its value can be vary.

Syntax ::

declaration

```
int x;  or  int x, y, z;
String y;
```

declaration + Assignment

```
int b = 100;
```

Value Assignment

```
x = 100;
```

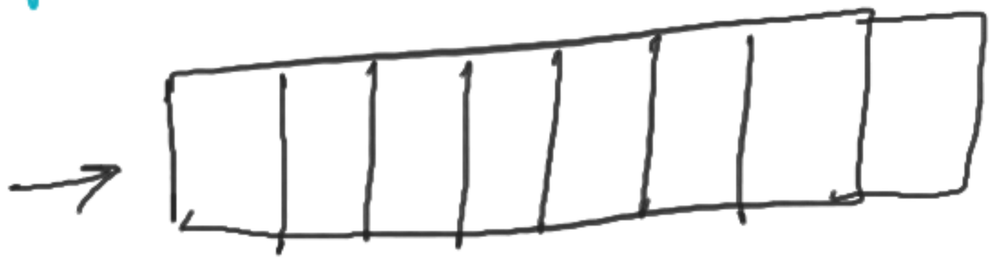
```
int a = 100; int b = 200;
int Sum;

Sum = a + b;
```

```
console [int a, b, Sum;]

Please enter Num1 →
        ↓
a = 10;

b = 20;

Sum = a + b;
```

int x = 10 ; → 4 Bytes

byte → 1 bitte → 8 bits < 0 / 1

0001



-128 to 127 64 32 16 8 4 2 1

128 0 0 0 0 0 0 0

1

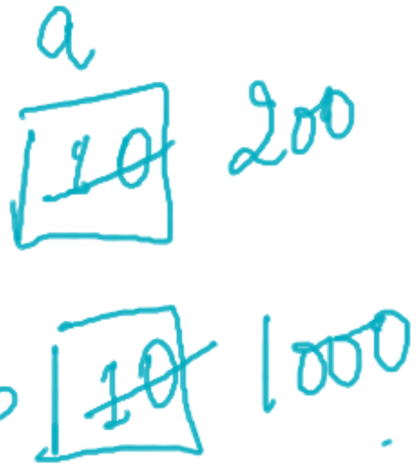
```
class FirstScript{
  public static void main(String[] args){
      int x=10;
      x=100;
      int a=1000;
      System.out.print("x");
      System.out.print(x);
      System.out.print("a");
      System.out.print(a);
  }
}
```

Syso(x) → 10

Syso(x) → 100

x → variable → 100

a → 1000



int a = 10;

int b = a;

b = 1000;

a = 200;

Syso(b); → 1000

Syso(a); → 200

```
class DataType{
  public static void main(String[] args){
      byte x=124;
      float flt = 12.34f;
      char ch = 'A';
      boolean bool = true;

      System.out.println(x);
      System.out.println(flt);
      System.out.println(ch);
       System.out.println(bool);

  }
}
```

Assignment :-

(i) Swapping of two Numbers with using third variable

(2) " " " without " " "