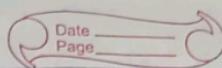


float & double
vary in precision
after point.



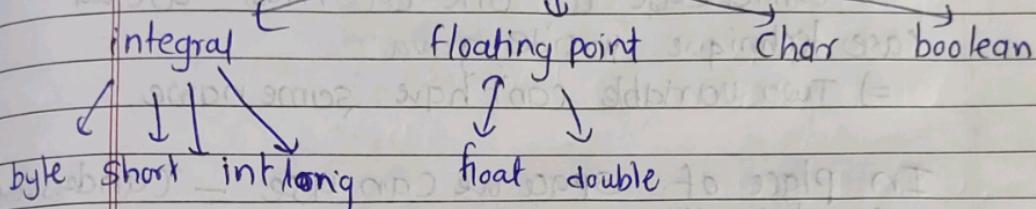
Read unicode.org

Data type & variable inside Java

In program we have to store data so for that we will use variables. for store data.

Each variable has own data type

Primitive datatype In Java



Each datatype has different size Range & default value

⇒ In Java if you initialise a variable you will not get a garbage value

- ⇒ there are also class is available for each primitive datatype

G
Ext

Int a;

also Integer a = new Integer();
↑

this has particular attribute like

MIN_VALUE ⇒ minimum possible int var

MAX_VALUE ⇒ maximum — 1 —

BYTES ⇒ size of object

Boolean Don't have size & Range

You should remember range such that you not
tackle error in next steps.

Variable in Java

- ⇒ start with alphabet, - or \$
- ⇒ can't start with number
- ⇒ Variable may contain numbers in them
- ⇒ Camel Case is one of most useful case technique inside Java
- ⇒ Two variable can't have same name

In place of space we can use - (underscore)
or \$ (Dollar symbol)

⇒ You can't use default java keywords inside Java,

literals in java

the constant value inside java code
are called a literals

$$Z = 10 * Y + 15 * X$$

① 10 & 15 are both the literals

to be more specific 10 & 15 are integer
literals

Int value = 25; → Int literal

float XXX = 26.789; ⇒ float literal

Double literal

any value with decimal point is
 double literal (all float variable)
 values are also double
 (literal)

char c='A'; string z="Hello world";
 ↑ character literal ↑ z ⇒ object
 string literal

literal distribution
 (continued)

Datatype	literal name
Byte	Int 10
short	Int 12
Int	Int 136789
long	Long 05L
float	Float 5f
double	Double 5d
char	'a'
boolean	true/false

2.5 ⇒ double literal

2.5f ⇒ float literal.

Int number system

You are able to write the numbers in
 3 or many number system

Decimal ⇒ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Binary ⇒ 0, 1

Octal ⇒ 0, 1, 2, 3, 4, 5, 6, 7

Hexadecimal ⇒ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

You can give input as any one of

use the datatype wisely it
will enable to make a optimised code.

above to the variable.

⇒ multiple integer types are there due to the technology at which java, CPP is created can't support 64 bit architecture (older machine)

⇒ so As evolution came the technology persists & we get more & more optimised data type in accordance to architecture of system

Say

You have architecture of 32 bit
i.e 4 byte

Scenario 1 ⇒ You're using a short datatype

Here you can pass 4 byte data but you're sending only 2 so this is called under processing

Scenario 2 ⇒ You're using a long datatype

Here you can pass 4 byte so as you're passing 8 byte so CPU have to fetch it in 2 cycle

↓

When Input size / architecture

Cycles will come to play

All integral datatypes in Java are signed
i.e. can have both positive or negative
signs

→ 2's Complement

S

find -5

$$00000101 \Rightarrow +5 \quad \text{as } S \text{ is 2's}$$

~~10000~~

compln(5)

+1

$$-5 \Rightarrow 11111011 \Rightarrow 2\text{'s complement}$$

166.67

16667 $\times 10^{-2}$ \Rightarrow Exponent

mantissa

178.62575

So IEEE format provides 17862575×10^{-4}

$17862575 E -4$

$$166.67 = \underline{\underline{16667 E -2}}$$

~~166~~ 78.67

$$6.23 \times 10^{24} \Rightarrow \underline{\underline{623 \times 10^{22}}}$$

$623 E 22$

No leading zeros after decimal point in IEEE

characters

A B C D Z

a b c d ... z

Java not support character so each character is binded with some ascii value & through which we can fetch the character.

a	b	c	z
65	66	97	98	99

A	B	C	z
65	66	67	97	98

Java take 2 byte to fetch character

\Rightarrow Boolean

Size \Rightarrow ~~length~~ (not defined)

True or false

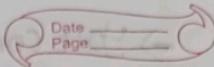
Interpreter VS Compiler

Interpreter run line by line check for error, warning & syntax, indentation & when it found error it will break current process & printout problem / error.

① Compiler

- It will run a whole code at once & check for error warning syntactical problem & all & at last it will print out all errors & warnings at same time.

Interpreter is slow than Compiler



If you resolve all error from compiler message you will not get errors further but in interpreter you may get it
→ |

⇒ Interpreter prints one error at a time.

⇒ Compiler return object file so if you want to reexecute program just use that object file

⇒ Interpreter needs to run on file as it not return any object file on demand fetching a file

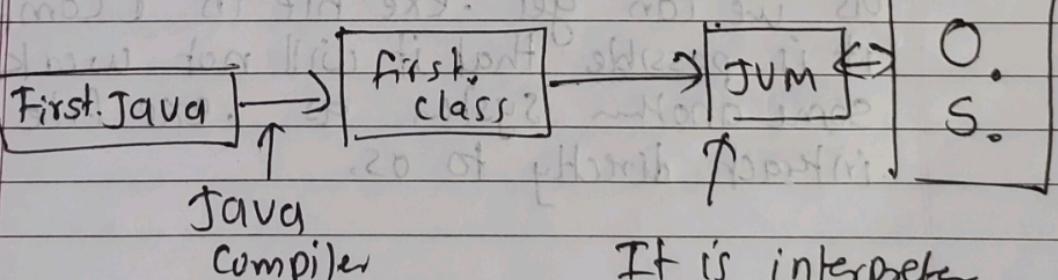
⇒ Python, JS ⇒ Interpreter language

⇒ C, CPP ⇒ Compiler language

But Java ⇒ It is a hybrid language

⇒ Interpreter is slow but while learning language the interpreter found more useful for removing bugs from programme

Working of Java code



JVM =

JIT

It is interpreter to run Java program.

- you compile .java file
- you interpret .class file
- JVM → Java virtual machine
- OS interacts with the hardware & JVM in Java file running execution
- Program interacts with hardware with the help of system calls.

In Java the executable Java file not directly interact with OS it uses JVM & JVM talk with OS & run our code

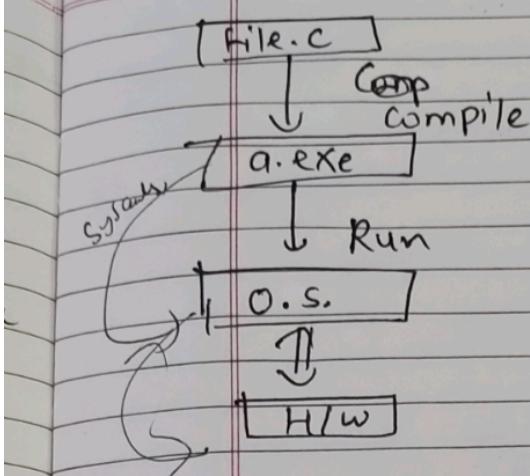
⇒ so same class file (executable) can run on JVM of Linux, Mac or any system. Hence Java is platform independent

⇒ JVM talk with OS

Every OS has its own supported JVMs

As we can get .exe file in C compiler it is possible that it will not work on some another system as .exe file interact directly to OS.

C workflow



Java Workflow

