

Language Fundamentals

Part 2:

* Identifier and keywords Reserved

* Data types

* Variables

* Literals

* main() method

* Methods

* Operators

* Control Statements

* Arrays

Identifiers and Reserved Keywords

Identifiers: Identifiers are the names that will be used to identify the programming elements like, Interface, classes, methods, variables... etc

Ex:

```
class Ati1
{
    public static void main (String[] args)
    {
        int x = 20;
        System.out.println ("Hello world");
        System.out.println (x);
    }
}
```

Identifiers (pointing to Ati1, main, args, x, String, args, System.out, println, x)

In the above program we have identifiers as,

classes \Rightarrow Ati1, System, String

variables \Rightarrow x, out, args

methods \Rightarrow main(), println()

Rules to follow when you define an Identifier

① The only allowed characters in Java identifiers are

a to z } small alphabets
A to Z } capital alphabets
0 to 9 } numbers from 0 to 9 or combination
_ } under score symbol
\$ } dollar symbol

Ex: a, b, c, A, Z, x, argt, -ati, \$ati,
name, STUDENT, last-name
last-name,

- ② First character of an identifier must be Alphabet (a-z) dollar (\$) or underscore (_)

Ex: amount ✓ amount25 ✓ we can use numbers after an alphabet.
Amount ✓ amount25\$ ✓
\$amount ✓ amount#25 X } in between also we can't use special char.
_amount ✓
#amount X } should not start with other symbol or number.
25amount X

- ③ Java Identifiers are case sensitive

Ex: Number } All are different.
number
NUMBER

- ④ There is no length limit for Java Identifiers, But it's not recommended to use more than 15 length (>15).

- ⑤ Reserved keywords can't be used as Identifiers

Ex: int, if, while, class, public, static, void, ...

↑ Invalid Identifiers.

we can't use reserved keywords.

[see the list of keywords on next page]

- ⑥ All Java predefined classes and Interface names can be used as Identifiers, But not recommended

Ex: int \$tring = 10; ✓ } not recommended
int Runnable = 20; ✓

- ⑦ Space is not allowed between words of Identifier

Ex: student_name ✓ lname ✓
student name X last name X

* which are valid/invalid Identifiers in Java?

Hello

Int

getClass Name

student Email

Total-FEE

1st class

true

Student number

-\$-

total #

Integer

all@hands

total-number-students

* Keywords *

- These are simple English words, which are having predefined meaning in Java programming lang.
- All the keywords defined in lower case.
- We can't use keywords as names of variables, methods, classes, Identifiers...
- true, false, null are not keywords but reserved as literals
- goto and const are ~~not~~ banned in Java ⇒ CTE

List of keywords

⇒ P.T.O

- Total 54 reserved keywords are in 'Java'
- 49 are used keywords
- 2 are reserved but not used keywords *
- 3 are reserved and are literals **

List of Keywords

Category	Keywords name
Data types (8)	byte, short, int, long, float, double, boolean, char
Class and object (10)	class, interface, extends, implements, enum, this, super, new, instanceof, new
Package (2)	Package, import
Access modifiers (3)	Private, Protected, Public
Modifiers (8)	final, native, abstract, synchronized, transient, volatile, static, strictfp
Control statements (11)	if, else, switch, case, default, do, while, break, for, continue, return
Exception Handling (6)	try, catch, finally, throw, throws, assert
Other data type (1)	void
Reserved (not used) (2)	const, goto
Keywords reserved (literal) (3)	true, false, null.

Perform the Assignment provided on Identifier & keywords

Data types

- Data types are used to define the type of the data to be stored.
- Specifies the Amount of data memory allocation required for your data.
- These are mainly two categories of data types are in java

① Primitive data types

② Non-Primitive data types (user defined data types)

* Primitive data types:

- Primitive data types are predefined data types
- There are 8 predefined data types such as boolean, char, int, short, byte, long, float and double

* Non-Primitive data types (user defined data types)

(discuss over the period of this course)

- User defined data types are the type of data types which are defined by user from an already present data type.

• ~~Class~~ Class type, Interface type, Enum, Annotation type are the four types of user defined data types

Ex: String, Array.... etc

Primitive data types

① byte : default value: 0

• Size: 8-bits (1 Byte)

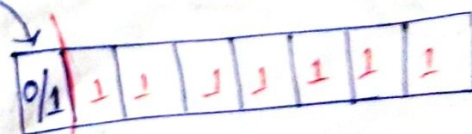
• Max-Value : 127

• Min-Value : -128

• Range : -128 to 127

Ex: byte b=100;
byte b=127;

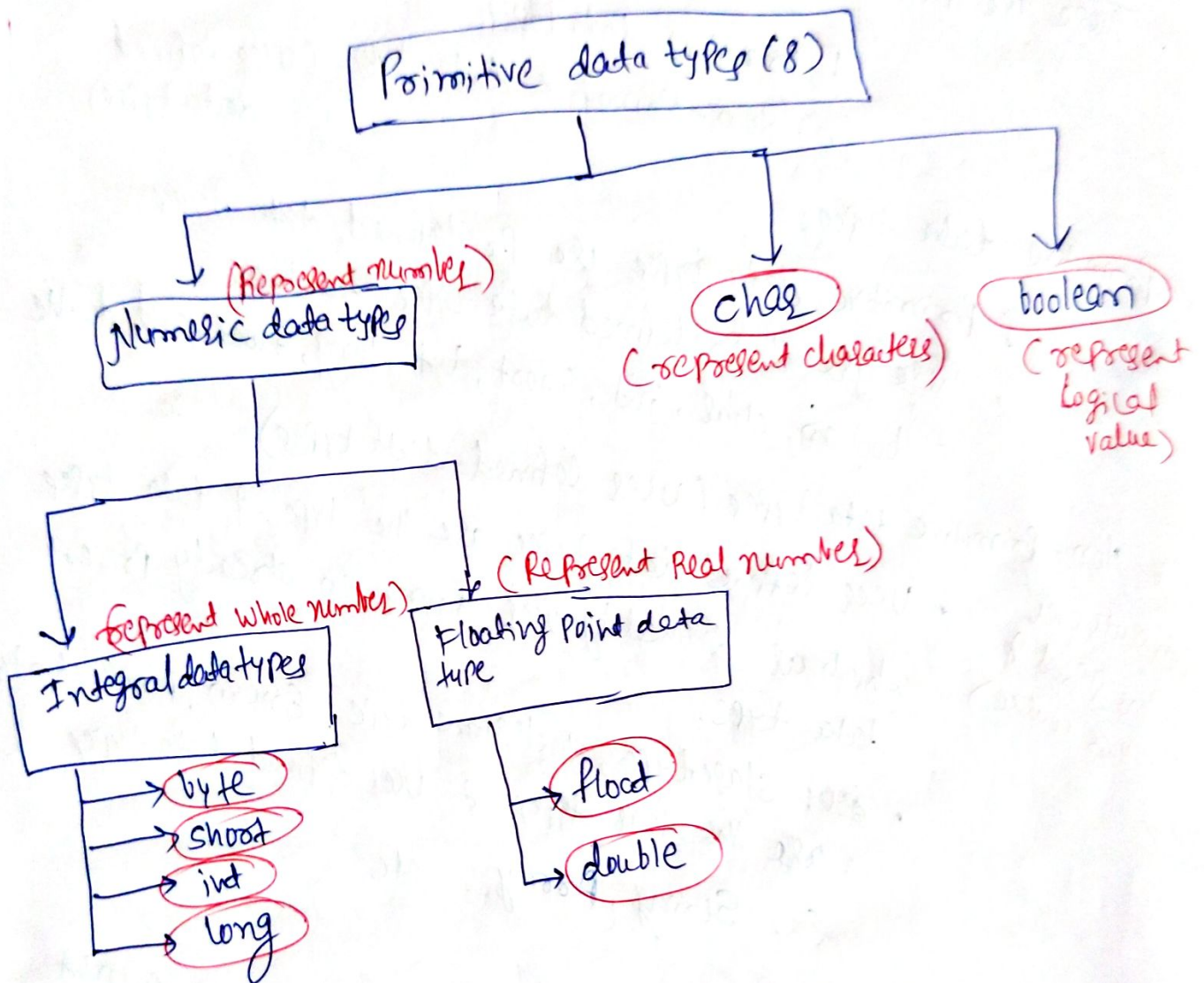
MSB (most significant bit) → sign bit



0 ⇒ 127 (+ve)
1 ⇒ -128 (-ve)

byte b = 130 : ✓
 byte b = 123.46 : ✗
 byte b = true : ✗
 byte b = "pure" : ✗

Note: byte data type is used for handling data in terms of streams from file or over network.



- ② short :
- least used data type
 - Save memory in large arrays.
 - outdated data type.
 - Size: 2 bytes (16 bits)
 - Range: -2^{15} to $2^{15}-1$
 -32768 to 32767
 - default value: 0
- ~~min~~ value max value

Ex: short s = 20; ✓
 short s = -32768; ✓
 short s = 32768; ✗
 short s = 123.768; ✗
 short s = true; ✗

(3)

int : →

• Most commonly used data type

• Size : 4 bytes (32 bits)

• Range : -2^{31} to $2^{31}-1$

$[-2147483648 \text{ to } 2147483647]$
~~min~~ value max value

• default value : 0

• Ex: int i = 20; ✓

int i = 2147483647; ✓

int i = 2147483649; ?

int i = -2147483648; ✓

int i = -2147483657; ?

int i = 5673214.78;

int i = true; ✗

(4)

long : →

• For storing large number

• Size : 8 bytes (64 bits)

• Range : -2^{63} to $2^{63}-1$

$[-9223372036854775808 \text{ to } 9223372036854775807]$

• default value : 0L

Ex: long l = 123000 × 60 × 60 × 24 × 1000;

* amount of distance travelled by light in 1000 days?

* To count number of char present in Big file?

⑤ float:

- size : 4 bytes (32 bits)
- Range : -3.4×10^{38} to 3.4×10^{38}
- If we want 5 to 7 decimal places of accuracy go for float
- default value : 0.0f

Ex: float f1 = 45.6 ; ✓
float f2 = 763458492.345 ; ✓
float f3 = -76345 ;
float f4 = true ; ✗
float f5 = 'a' ; ✗

⑥ double:

- size : 8 bytes (64 bits)
- Range : -1.7×10^{308} to 1.7×10^{308}
- default value : 0.0d
- If we want upto 15 decimal place of accuracy go for double.

Ex: double d1 = 345658 ;
double d2 = 6781435624.5124532 ;
double d3 = -34156348 ;
double d4 = true ; ✗
double d5 = 'a' ; ✗

⑦ char :

- C/C++ size of char is 1 byte as it follows ASCII characters.
- Java size of char is 2 byte as it follows Unicode characters (which covers wide all language alphabets sets)

Size : 2 bytes

Range : 0 to 65535

- No -ve values (not numeric value)
- default value : min : 'u0000' (0) 0
max : 'uFFFF' (65535) 65535

Ex:
char c1 = 'a'; char c6 = 'OK'; ✗
char c2 = '2';
char c3 = '\$';
char c4 = '-';
char c5 = true; ✗
char c7 = '65535';

⑧ boolean (1 bit or 1 byte)

* Size →

* Range → "Not applicable"
values allowed only → true, false

* Ex:
boolean b1 = 0; ✗
boolean b2 = true; ✓
boolean b3 = false; ✓
boolean b4 = True; ✗
boolean b5 = 'T'; ✗
boolean b6 = "true"; ✗

Data Types Summary Table (Primitive)

Type	Size		default value	Range	
	Byte	Bits		Min value	Max value
byte	1	8	0	-128	127
short	2	16	0	-2^{15} (-32768)	$2^{15}-1$ (32767)
int	4	32	0	-2^{31}	$+2^{31}-1$
long	8	64	0L	-2147483648 -2^{63}	2147483647 $2^{63}-1$
float	4	32	0.0f	-3.4e38 -9223372036854775808	3.4e38 9223372036854775807
double	8	64	0.0d	-1.7e308	1.7e308
char	2	16	'\u0000' @ 0	'\u0000' @ 0	65535
boolean	1 bit or 1 byte	1 bit or 1 byte	false	NA true/false	NA true/false