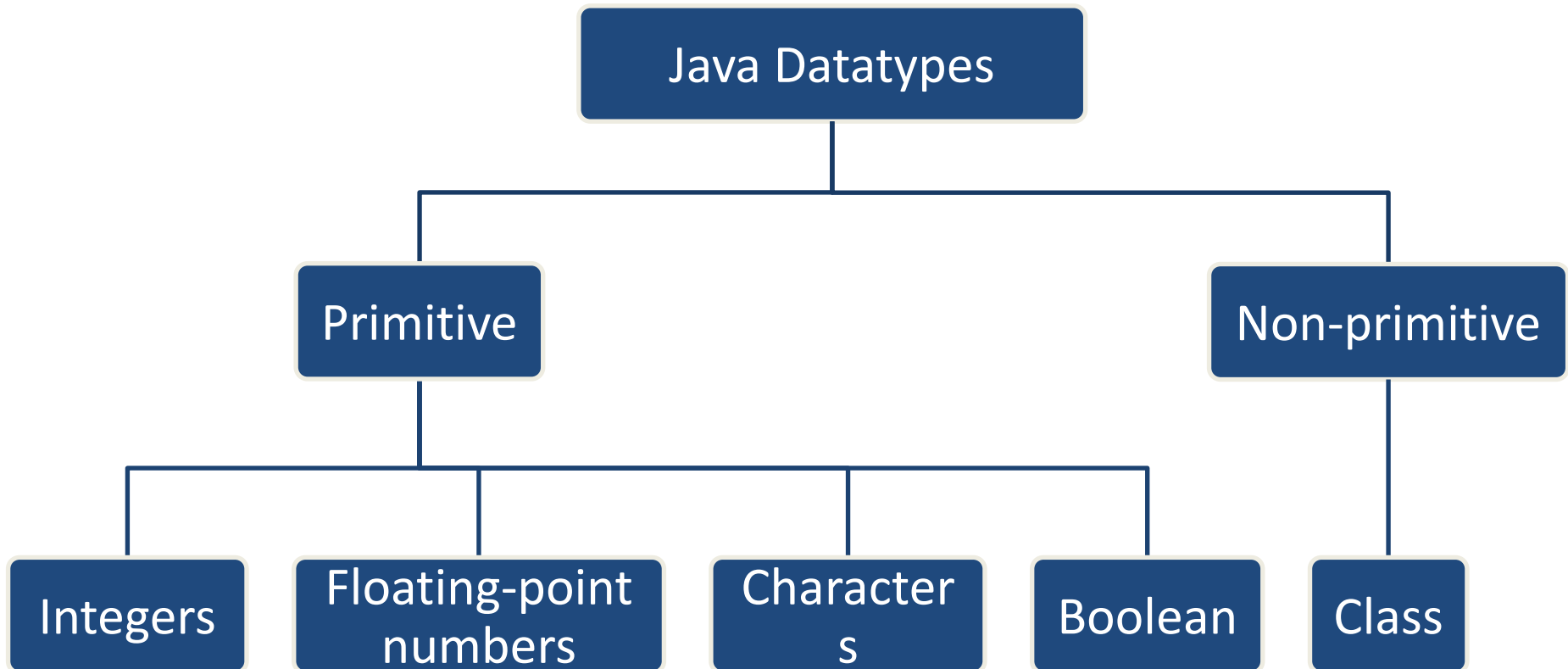


Data Types




Primitive Data Types


byte	1 Byte	-128 to 127	byte a = 10;
short	2 Bytes	-32,768 to 32,767	short a = 200;
int	4 Bytes	-2,147,483,648 to 2,147,483,647	int a = 50000;
long	8 Bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	long a = 20;
float	4 Bytes	1.4e-045 to 3.4e+038	float a = 10.2f;
double	8 Bytes	4.9e-324 to 1.8e+308	double a = 10.2;
char	2 Bytes	0 to 65536 (Stores ASCII of character)	char a = 'a';
boolean	Not defined	true or false	boolean a = true;

Type Casting

- Assigning a value of one type to a variable of another type is known as Type Casting.
- In Java, type casting is classified into two types,
 - Widening/Automatic Type Casting (Implicit)

byte → short → int → long → float → double

widening

- Narrowing Type Casting(Explicitly done)

double → float → long → int → short → byte

Narrowing

Operators

1. Arithmetic Operators
2. Relational Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Conditional / Ternary Operator
7. Instance of Operator

Relational Operators

Note : A = 10 & B = 20

==	Equals	(A == B) is not true.
!=	Not Equals	(A != B) is true.
>	Greater than	(A > B) is not true.
<	Less than	(A < B) is true.
>=	Greater than equals	(A >= B) is not true.
<=	Less than equals	(A <= B) is true.
==	Equals	(A == B) is not true.

Arithmetic Operator

Note : A = 10 & B = 20

+	Addition	$A + B = 30$
-	Subtraction	$A - B = -10$
*	Multiplication	$A * B = 200$
/	Division	$B / A = 2$
%	Modulus	$B \% A = 0$
++	Increment	$B++ = 21$
--	Decrement	$B-- = 19$

Bitwise Operators

Note : A = 60 & B = 13

&	Binary AND Operator	A & B = 12 which is 0000 1100
	Binary OR Operator	A B = 61 which is 0011 1101
^	Binary XOR Operator	A ^ B = 49 which is 0011 0001
~	Binary Ones Complement Operator	~A = -61 which is 1100 0011 in 2's complement form due to a signed binary number.
<<	Binary Left Shift Operator	A << 2 = 240 which is 1111 0000
>>	Binary Right Shift Operator.	A >> 2 = 15 which is 1111
>>>	Shift right zero fill operator.	A >>>2 = 15 which is 0000 1111

Logical Operators

Note : A = true & B = false

&&	Logical AND operator	(A && B) is false.
	Called Logical OR Operator	(A B) is true.
!	Called Logical NOT Operator	!(A && B) is true.

Assignment Operators

=	Simple assignment operator	$C = A + B$ will assign value of $A + B$ into C
+=	Add AND assignment operator	$C += A$ is equivalent to $C = C + A$
-=	Subtract AND assignment operator	$C -= A$ is equivalent to $C = C - A$
*=	Multiply AND assignment operator	$C *= A$ is equivalent to $C = C * A$
/=	Divide AND assignment operator	$C /= A$ is equivalent to $C = C / A$
%=	Modulus AND assignment operator	$C \% = A$ is equivalent to $C = C \% A$
<<=	Left shift AND assignment operator	$C <<= 2$ is same as $C = C << 2$
>>=	Right shift AND assignment operator	$C >>= 2$ is same as $C = C >> 2$
&=	Bitwise AND assignment operator	$C \&= 2$ is same as $C = C \& 2$
^=	bitwise exclusive OR and assignment operator	$C \wedge= 2$ is same as $C = C \wedge 2$
=	bitwise inclusive OR and assignment operator	$C = 2$ is same as $C = C 2$

Conditional Operator (Ternary)

- Conditional Operator (? :)

- Syntax:

variable x = (expression) ? value if true : value if false

- Example:

```
b = (a == 1) ? 20 : 30;
```

instanceof Operator

- instanceof Operator

- Syntax:

(Object reference variable) instanceof (class/interface type)

- Example:

boolean result = name instanceof String;

Precedence of Java Operators

Postfix	() [] . (dot operator)	Left to right
Unary	++ -- ! ~	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	>> >>> <<	Left to right
Relational	> >= < <=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR		Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %= >>= <<= &= ^= =	Right to left
Comma	,	Left to right