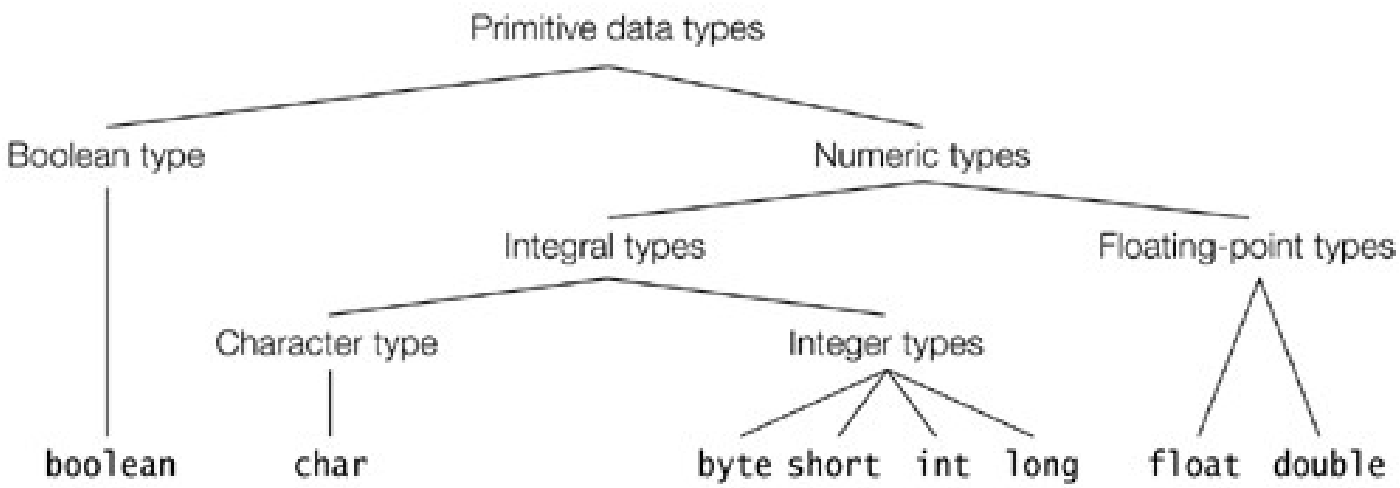
Java Expressions

Hani Safadi

Primitive Data Types



Primitive Data Types

Data Type	Width (bits)	Minimum Value, Maximum Value
`boolean`	not applicable	`true`, `false` (no ordering implied)
`byte`	8	-2^7, 2^7-1
`short`	16	-2^{15}, $2^{15}-1$
`char`	16	`0x0`, `0xffff`
`int`	32	-2^{31}, $2^{31}-1$
`long`	64	-2^{63}, $2^{63}-1$
`float`	32	±`1.40129846432481707e-45f`, ±`3.40282347663852886 0e+38f`
`double`	64	±`4.94065645841246544e-324`, ± `1.79769313486231570e+308`

Variable Declaration

int a;
int b;
int c;
 	
a=1;
b=2;
c=3;
	int a, b, c;
	
	a=1;
	b=2;
	c=3;
		int a=1, b=2, c=3;

Operators

Unary	+	Addition	–	Subtraction		
Binary	*	Multiplication	/	Division	%	Remainder
	+	Addition	–	Subtraction		

Q: Operators

```
int i = 7;
int j = 5;
System.out.println(i / j); //integer division
System.out.println(i % j);
double x = 7;
double y = 5;
System.out.println(x / y);
System.out.println(x % y); //works on the integer part
```

Assignment with Operators

```
int i = 5;
i += (i = 3);
System.out.println(i);

int j = 5;
j = j + (j = 3);
System.out.println(j);
```

Increment and Decrement Operators ++, --

```
int i = 1;
i = i+1;
i +=1;
i++;
++i;
System.out.println(i);
System.out.println(++i +1);
System.out.println(i++ + 1);
```

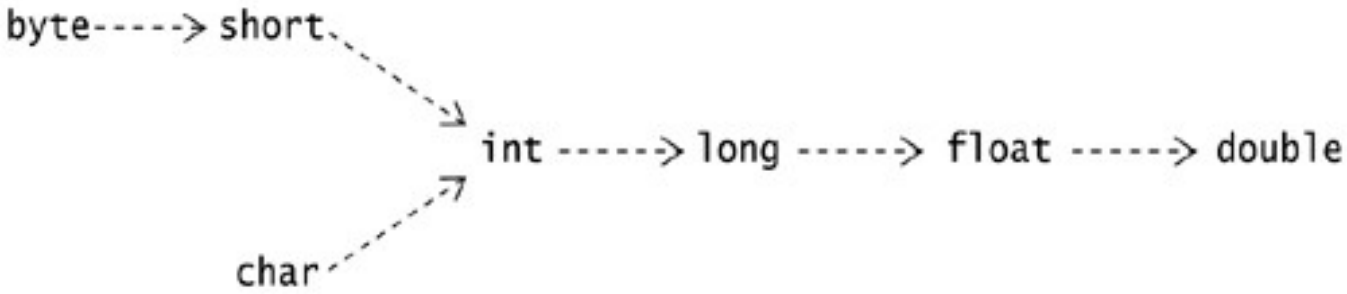
Precedence

Postfix operators	`[] . (parameters) expression++ expression--`
Unary prefix operators	`++expression --expression +expression -expression !`
Unary prefix creation and cast	`new (type)`
Multiplicative	`* / %`
Additive	`+ -`
Shift	`<< >> >>>`
Relational	`< <= > >=`
Equality	`== !=`
Assignment	`= += -= *= /= %=`

Q: Precedence

- System.out.println(2 + 3 * 4);
- System.out.println(1 + 2 – 3);
- System.out.println(- - 4);
- int b = 10;
  System.out.println((b=3) + b);
- int i = 5;
  i = i + 1;
  i = 20 - i * 2;
  System.out.println(i);

Conversion



Q: Conversion

```
int i = 3;
long l = i; //OK

byte b = i; //NOT OK
byte b = (byte) i; //OK, but not correct if i>256
```

Strings

- String s = “Test”;
- String s = new String(“Test”);
- System.out.println(s.length());
- System.out.println(s.equals(“Test”));
- String s2 = “this is a” + s;
- System.out.println(s2);

Q: Strings

```
String s1 = “7UP”;
String s2 = new String(“7UP”);
System.out.println(s1.length());
System.out.println(s1 = = s2);
System.out.println(s1.equals(s2));
```

References

- Java Software Solutions
- Java for Engineers and Scientists
- A Programmer's Guide to Java Certification