Java and C similarities, by example

(not a complete list)

Function	Java	C
comment	/* comment */ // another kind of comment	/* comment */
assignment	i= i+j;	i= i+j;

block	{
	statement 1;
	statement 2;
	}
		{
		statement 1;
		statement 2;
		}

conditional statement	**if** (*expression*) *statement*
	else *statement*
		if (*expression*) *statement*
		else *statement*

for loop	**for** (**int** i=1; i<=10; i++) *statement*	**for** (**int** i=1; i<=10; i++) *statement*
for loop	**for** (**int** i=10; i>0; i--) *statement*	**for** (**int** i=10; i>0; i--) *statement*
while loop	**while** (i < 10) *statement*	**while** (i < 10) *statement*
return statement	**return** ; (in a procedure)	**return** ;

	return x; (in a function)	**return** x;
terminate a loop	break;	break;
terminate a loop body	continue;	continue;
function call	m(y,z)	m(y,z)
procedure call	m(y,z);	m(y,z);
equality and inequality	== and !=	== and !=
logical operators	&& (logical-and) || (logical-or) ! (logical-complement)	&& || !
arithmetic operators	unary -, +, -, *, /, %	unary -, +, -, *, /, %
string catenation	+	/* no C equivalent */
integral types	**byte** (8 bits), **short** (16 bits) **int** (32 bits) , **long** (64 bits)	**short** , **int** , **long**
floating point types	**float** (32 bit), **double** (64 bit)	**float** , **double**
character type	**char**	**char**
boolean type	**boolean**	**int** (C has no type boolean)
declarations of integer variables	**int** i,j,k;	**int** i,j,k;
declaration of a constant	**final int** MAX= 100;	#define MAX = 100;
declaration and creation of an array	**int**[] A= **new int**[10];	**int** A[10];

declaration and creation of two-dimensional array	**float**[][] B= **new float**[10][100];	**float** B[10][100];
declaration of a string variable	String s;	**char** *s; s = malloc(10);
declaration of a C "struct"	**class** r { **char** a; **int** b: }	**struct** r { **char** a; **int** b; }
declaration of a pointer variables	/* no Java equivalent. */	**int** *b;

declaration of a function	**char** a (**int** b) {
	...
	return 'X';
	}
		char a (**int** b) {
		...
		return 'X';
		}