

C++ and Java Side-by-Side

C++ Data Types		
Category	Type	Range
Integer	short	-32,768 to 32,767
Integer	int	-2,147,483,648 to 2,147,483,647
Real number	float	-3.4^{38} to -1.2^{-38} , 0, 1.2^{-38} to 3.4^{38}
Real number	double	-1.8^{308} to -2.2^{-308} , 0, 2.2^{-308} to 1.8^{308}
Boolean	bool	true or false
Character	char	A single character, enclosed by single quotes
String	string	A sequence of characters, enclosed by double quotes
Java Data Types		
Category	Type	Range
Integer	byte	-128 to 127
Integer	short	-32,768 to 32,767
Integer	int	-2,147,483,648 to 2,147,483,647
Integer	long	-2^{63} to $2^{63}-1$
Real number	float	-3.4^{38} to -1.4^{-45} , 0, 1.4^{-45} to 3.4^{38}
Real number	double	-1.8^{308} to -4.9^{-324} , 0, 4.9^{-324} to 1.8^{308}
Boolean	bool	true or false
Character	char	A single character, enclosed by single quotes
String	String	A sequence of characters, enclosed by double quotes

C++ Declarations	
Statement	Purpose
int i;	Declare integer i
int x = 55;	Declare integer x with value 55
const int LAST = -99;	Declare integer constant LAST with value -99
Java Declarations	
Statement	Purpose
int i;	Declare integer i
int x = 55;	Declare integer x with value 55
final int LAST = -99;	Declare integer constant LAST with value -99

C++ and Java Side-by-Side

C++ Assignments, Inputs, Outputs	
Statement	Purpose
<code>i++;</code>	Add one to i
<code>i--;</code>	Subtract one from i
<code>x = y + z;</code>	Add y and z and place result in x
<code>letter = 'A';</code>	Place character 'A' in letter
<code>greeting = "Hello";</code>	Place string "Hello" in greeting
<code>cin >> x;</code>	Read value into x from keyboard
<code>cout << "Hello" << "World";</code>	Write strings "Hello" and "World" to screen
<code>cout << "Hello" << endl;</code>	Write string "Hello" to screen and move to next line
Java Assignments, Inputs, Outputs	
Statement	Purpose
<code>i++;</code>	Add one to i
<code>i--;</code>	Subtract one from i
<code>x = y + z;</code>	Add y and z and place result in x
<code>letter = 'A';</code>	Place character 'A' in letter
<code>greeting = "Hello";</code>	Place string "Hello" in greeting
<code>x = keyboard.nextInt();</code>	Read value into x from keyboard
<code>System.out.print ("Hello" + "World");</code>	Write strings "Hello" and "World" to screen
<code>System.out.println ("Hello");</code>	Write string "Hello" to screen and move to next line

C++ and Java Side-by-Side

C++ Arithmetic Operators	
Operator	Purpose
+	Addition
−	Subtraction
*	Multiplication
/	Division
%	Modulo (integer remainder)
Java Arithmetic Operators	
Operator	Purpose
+	Addition (and concatenation)
−	Subtraction
*	Multiplication
/	Division
%	Modulo (integer remainder)

C++ Relational Operators	
Operator	Purpose
==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
Java Relational Operators	
Operator	Purpose
==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

C++ and Java Side-by-Side

C++ Logical Operators	
Operator	Purpose
!	Not
&&	And
	Or
Java Logical Operators	
Operator	Purpose
!	Not
&&	And
	Or

C++ Escape Sequences	
Sequence	Purpose
\n	Start on a new line.
\t	Tab on the current line.
\'	Output a single quote.
\"	Output a double quote.
\\	Output a backslash.
\a	Beep user.
Java Escape Sequences	
Sequence	Purpose
\n	Start on a new line.
\t	Tab on the current line.
\'	Output a single quote.
\"	Output a double quote.
\\	Output a backslash.

C++ and Java Side-by-Side

C++ Statements	
(<block> is one or more statements; if <block> is more than one statement, it must be delimited with curly braces {})	
Statement	Purpose
if	<pre> if (<condition>) <block> else if (<condition>) <block> else <block> </pre>
while	<pre> while (<condition>) <block> </pre>
do while	<pre> do <block> while (<condition>) </pre>
for	<pre> for (<initialization>; <condition>; <update>) <block> </pre>
switch	<pre> switch (<expression>) { case <constant-1>: <block>; break; case <constant-2>: <block>; break; ... case <constant-n>: <block>; break; default: <block>; } </pre>

C++ and Java Side-by-Side

Java Statements	
(<block> is one or more statements; if <block> is more than one statement, it must be delimited with curly braces {})	
Statement	Purpose
if	<pre> if (<condition>) <block> else if (<condition>) <block> else <block> </pre>
while	<pre> while (<condition>) <block> </pre>
do while	<pre> do <block> while (<condition>) </pre>
for	<pre> for (<initialization>; <condition>; <update>) <block> </pre>
switch	<pre> switch (<expression>) { case <constant-1>: <block>; break; case <constant-2>: <block>; break; ... case <constant-n>: <block>; break; default: <block>; } </pre>