

Type of Expression	Example
Declare a class	<pre>class Foo {</pre> <pre>}</pre>
Declare the method named main	<pre>public static void main (String[] args) {</pre> <pre>}</pre>
PRINTING STRINGS	
Print a String to the console	<pre>System.out.println("I am a string");</pre>
DECLARING A SCANNER	
Declare a Scanner that reads from the keyboard	<pre>Scanner kb = new Scanner(System.in);</pre>
DECLARING VARIABLES	
Declare an variable for an integer and initialize	<pre>int i = 0;</pre>
Declare a variable for an character and initialize	<pre>char c = 'C';</pre>
Declare a variable for a decimal and initialize	<pre>double d = 0.0;</pre>
Declare a variable for a boolean value and initialize	<pre>boolean b = true;</pre>
Declare a variable for a String and initialize	<pre>String s = "I am a string";</pre>
PRINTING STRINGS AND VALUES	
Print the value of a variable	<pre>System.out.println(i);</pre>
Print a String followed by the value of a variable	<pre>System.out.println("The value of i is: " + i);</pre>
SETTING VARIABLES	
Set a variable equal to a literal value	<pre>i = 5;</pre>
Declare a variable and initialize it to the value held in another variable.	<pre>int j = i;</pre>
READING INTEGERS FROM KEYBOARD	
Set a variable equal to the value read from the keyboard	<pre>j = kb.nextInt();</pre>
DOING ARITHMETIC	

Set a variable equal to the sum of the values held in two variables	<code>int k = i + j;</code>
Set a variable equal to the product of the value in a variable and a literal value.	<code>k = i + 3;</code>
Set a variable equal to the difference of two integer values	<code>k = i - 2;</code>
Set a variable equal to the quotient of two integers values	<code>// Note: the result is an integer (rounded down)</code> <code>i = j / 5;</code>
READING DECIMAL VALUES FROM KEYBOARD	
Read a decimal number from the keyboard	<code>d = kb.nextDouble();</code>
FLOATING-POINT (DECIMAL) DIVISION	
Declare a variable that can hold a decimal value and initialize it to the quotient of an integer and a decimal number.	<code>// Note: the result is a decimal number</code> <code>double f = i / 2.0;</code>
READ STRING FROM KEYBOARD	
Read a String from the keyboard	<code>s = kb.next();</code>
CONDITIONALS WITH INT COMPARISONS	
Run a block of code if the value in a primitive variable is equal to a literal value	<code>if (i == 2) {</code> <code>// do something</code> <code>}</code>
Run a block of code if the value in a primitive variable is equal to the value in another primitive variable	<code>if (i == j) {</code> <code>// do something</code> <code>}</code>
Run a block of code if a primitive value is NOT equal to a literal value	<code>if (i != 2) {</code> <code>// do something</code> <code>}</code>
Run a block of code if a primitive value is less than a literal value	<code>if (i < 2) {</code> <code>// do something</code> <code>}</code>
Run a block of code if the value in a variable is less than or equal to a literal value; otherwise run a second block of code	<code>if (i <= 2) {</code> <code>// do something</code> <code>}</code> <code>else {</code> <code>// do something else</code> <code>}</code>
CONDITIONALS WITH CHAR COMPARISONS	

Run one block of code if a value in a variable is equal to one value; else run a second block of code if the value in the variable is a second value; otherwise run a third block of code	<pre> if (c == 'a') { // do something } else if (c == 'b') { // do something else } else { // do something possibly different } </pre>
CONDITIONALS WITH STRINGS	
Run a block of code if the value in a String is equal to a string literal.	<pre> if (s.equals("hello")) { // do something } </pre>
Run a block of code if the value in a String is equal to the value in another String	<pre> String t = "test"; ... if (s.equals(t)) { // do something } </pre>
Run a block of code if the value in a String is NOT equal to the value in another String	<pre> if (!s.equals(t)) { // do something } </pre>
GETTING A CHAR FROM KB	
Read a single character from the keyboard	<pre> c = kb.next().charAt(0); </pre>
MODULUS OPERATOR	
Store in z the remainder of x when divided by y	<pre> int z = x % y; </pre>
?: OPERATOR	
set b to true if c equals 'y' else set b to false using ?:	<pre> boolean b = (c == 'y') ? true : false; </pre>
METHODS	
define a static method named foo that has two integer parameters and returns a double	<pre> static double foo (int a, int b) { // do something with a and b // ... // return a decimal value } </pre>
call foo, passing in the values in x and y, and store the value returned by the method in d	<pre> d = foo(x,y); </pre>
MATH CLASS	
set f equal to the square root of d	<pre> double f = Math.sqrt(d); </pre>
ARRAYS	
declare an array that can hold 3 integers	<pre> int[] arr = new int[3]; </pre>

set the element at index 0 in arr to the value 7	<code>arr[0] = 7;</code>
get the element at index 1 in arr and store in variable named x	<code>x = arr[1];</code>
get the length of arr and store in variable named len	<code>int len = arr.length;</code>
INCREMENT OPERATOR	
increment the value of a variable named x	<code>x++;</code> // equivalent to <code>x = x + 1</code>
WHILE-LOOPS	
print the elements of an array	<code>int index = 0;</code> <code>while(index < arr.length) {</code> <code>SOP(arr[index])</code> <code>index++;</code> <code>}</code>
set all of the elements in an array to 21.	<code>int index = 0;</code> <code>while(index < arr.length) {</code> <code>arr[index] = 21;</code> <code>index++;</code> <code>}</code>
counting the number of positive elements in an array	<code>int index = 0;</code> <code>int count = 0;</code> <code>while(index < arr.length) {</code> <code>if (arr[index] > 0) {</code> <code>count++;</code> <code>}</code> <code>index++;</code> <code>}</code> <code>SOP(count);</code>
FOR-LOOPS	
print the elements of an array	<code>while(int i = 0; i < arr.length; i++) {</code> <code>SOP(arr[i])</code> <code>}</code>
set all of the elements in an array to 21.	<code>while(int i = 0; i < arr.length; i++) {</code> <code>arr[i] = 21;</code> <code>}</code>
counting the number of positive elements in an array	<code>int count = 0;</code> <code>while(int i = 0; i < arr.length; i++) {</code> <code>if (arr[i] > 0) {</code> <code>count++;</code> <code>}</code> <code>}</code> <code>SOP(count);</code>
FOR-EACH LOOPS	
Print elements in array of integers	<code>for (int elm : arr) {</code> <code>SOP(elm);</code> <code>}</code>
2D ARRAYS	

Declare 3x4 2D array of integers	<pre>int[][] m = new int[4][5];</pre>
Print elements in 2D array of integers using for-each loop	<pre>for(int[] row : m) { for(int elm : row) { SOP(elm); } }</pre>
Print elements in 2D array of integers using indices	<pre>for(int i = 0; i < m.length; i++) { for(int j = 0; j < m[i].length; j++) { SOP(m[i][j] + " "); } SOP("\n") }</pre>
Set element at row 3, column 2 to 7	<pre>m[3][2] = 7;</pre>
Get the value of the element at row 3, column 2.	<pre>int value = m[3][2];</pre>
DO-WHILE LOOPS	
Repeatedly ask user to enter letter until the enter 'e'.	<pre>char choice = '\0'; do { SOP('enter letter'); choice = kb.next().charAt(0); } while (choice != 'e');</pre>
SWITCH STATEMENTS	
Print command based on letter entered by user.	<pre>SOP('enter letter'); char choice = kb.next().charAt(0); switch(choice) { case 'r': SOP("run"); break; case 's': SOP("stop") break; default: SOP("not run or stop"); }</pre>
BREAK	
Loop from 0 to 9. Break out of loop when i is equal to input.	<pre>int input = kb.nextInt() int i = 0; while (i < 10) { if (i == input) { break; } SOP(i); i++; }</pre>
CONTINUE	

Loop from 0 to 9. Skip SOP (and continue at top of loop) when i is equal to input.	<pre> int input = kb.nextInt() int i = 0; while (i < 10) { if (i == input) { continue; } SOP(i); i++; } </pre>