

Programming I
Java Language Examples

declare class with main method	<pre>class MyClass { public static void main(String[] args) { _____ } }</pre>
variable declaration and initialization	<pre>int i = 10; char c = 'Z'; double d = 10.4; boolean b = false;</pre>
Print value in variable	<pre>int i = ... System.out.println(i);</pre>
arithmetic operators	<pre>int i = ... int j = ... int k = i + j; k = i * j; k = i / j; k = i % j;</pre>
integer division	<pre>double d = ... double e = d / i;</pre>
floating-point division	<pre>double d = ... double e = d / (double) i;</pre>
override operator precedence with parenthesis	<pre>int x = (1 + 2) * 3;</pre>
boolean expressions: comparing primitive types	<pre>int i = ...; char c = ...; double d = ...; boolean b = ... (i == 10) (c == 'f') (d == 12.3) (b == true) (i != 10)</pre>

boolean expressions: relational operators	<pre>int i = ... (i < 10) (i <= 10) (i > 10) (i >= 10)</pre>
boolean expressions: logical operators	<pre>int i = ... (i > 0 && i < 10) (i == 1 i == 2) !(i == 10)</pre>
conditional block	<pre>int i = ... if (i > 10) { // i is greater than 10 _____ } else if (i < 0) { // i is NOT greater than 10, but i is less than 0 _____ } else { // i is not greater than 10 and i is not less than 0 _____ }</pre>
?: operator	<pre>int i = ... int x = (i < 10) ? 7 : 11;</pre>
while-loop	<pre>int i = ... while (i < 10) { // i is less than 10 _____ i++; }</pre>
for-loop	<pre>for(int i = 0; i < 10; i++) { // i is less than 10 } }</pre>
do-while loop	<pre>do { // loop is executing for the first time OR i is less than 10 _____ i++; } while(i < 10);</pre>

switch statement	<pre> int i = ... switch(i) { case 1: _____ break; case 1: _____ break; default: _____ } </pre>
array declaration	<pre> int[] array = new int[10]; </pre>
get length of array	<pre> int[] array = ... int len = array.length; </pre>
set value (at index i) in array (to val)	<pre> int i = ... int[] array = ... int val = ... array[i] = val; </pre>
get value (at index i) in array	<pre> int i = ... int[] array = ... int z = array[i]; </pre>
get each element in array	<pre> int[] array = ... for(int i = 0; i < array.length; i++) { int elm = array[i]; _____ } </pre>
get each element in array	<pre> int[] array = ... for(int elm : array) { // elm holds different element during each iteration of loop _____ } </pre>
set each element in array (to val)	<pre> int[] array = ... int val = ... for(int i = 0; i < array.length; i++) { array[i] = val; _____ } </pre>

declare 2D array	<pre>int numRows = ... int numCols = ... int[][] array = new int[numRows][numCols];</pre>
get each element in 2D array	<pre>int[][] array = ... for(int i = 0; i < array.length; i++) { for(int j = 0; j < array[i].length; j++) { int elm = array[i][j]; } }</pre>
get each element in 2D array (with for-each loop)	<pre>int[][] array = ... for(int[] row : array) { for(int elm : row) { // elm holds a different element during each iteration of loop } }</pre>
set each element in 2D array (to val)	<pre>int[][] array = ... int val = ... for(int i = 0; i < array.length; i++) { for(int j = 0; j < array[i].length; j++) { array[i][j] = val; } }</pre>
String declaration and initialization	<pre>String s = "Joe";</pre>
length of string	<pre>int len = s.length();</pre>
determine if strings are equal	<pre>String s = ... String t = ... if (s.equals(t)) { // characters in s are the same as characters in t _____ }</pre>
determine if strings are NOT equal	<pre>String s = ... String t = ... if (!s.equals(t)) { // characters in s are NOT the same as characters in t _____ }</pre>

get character (at index i) in String	String s = ... int i = ... char c = s.charAt(i);
Scanner declaration and initialization for reading keyboard	Scanner kb = new Scanner(System.in)
changing Scanner delimiters (to commas and newline characters)	Scanner sc = ... sc.useDelimiter(", \\n");
Scanner declaration and initialization for reading file	String filename = ... Scanner input = null; try { input = new Scanner(new File(filename)); } catch(FileNotFoundException e) { // no file exists with name in filename }
PrintWriter declaration and initialization for writing to file.	String filename = ... PrintWriter output = null; try { output = new PrintWriter(new File(filename)); } catch(FileNotFoundException e) { // error opening file }
method declaration (with single parameter and return value)	boolean foo(int x) { if (x == 10) { return true; } return false; }
method invocation (aka method call)	int k = ... boolean result = foo(k);

entity (model) class definition	<pre>class Entity { private int x = 0; public Entity(int y) { this.x = y; } public int getX() { return x; } public void setX(int y) { this.x = y; } }</pre>
Override toString	<pre>@Override public String toString() { String str = ""; _____ return str; }</pre>