Julián Gonzalez Endzweig c

**Trabajo práctico nro. 3**

|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Laboratorio I** | |
|  | |
| **Cursado:**Primer Trimestre | **Horas** **semanales**: |
|  | **Horas semestrales:**  *Cantidad estimada de horas semestrales/anuales.* |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

**Integrantes de la Cátedra:**

* **DOCENTES:**

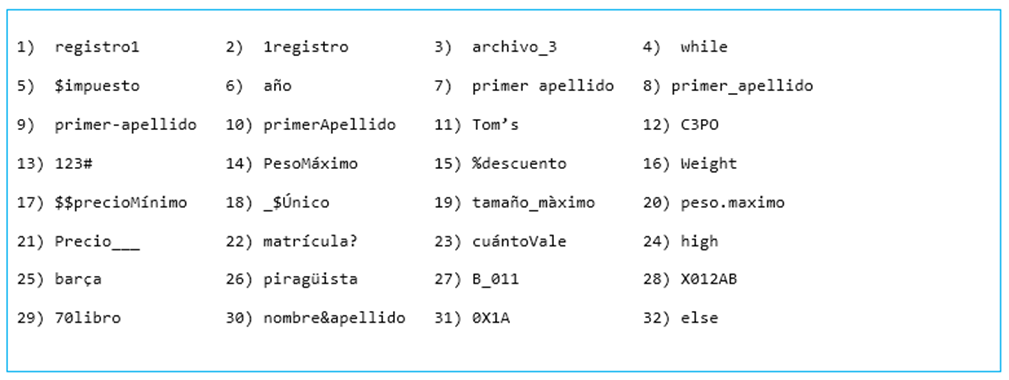
|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
|  |  | 6 horas |

1. **Tema:**

**Ejercicios Básicos iniciales para principiantes**

1. **Enunciados:**
2. **Ejercicio sobre identificadores: indica qué identificadores Java son válidos.**

Indica cuáles de los siguientes identificadores son válidos en Java. Si el identificador no es válido explica porqué no lo es.

1.1)Válido.

1.2)Identificador no válido porque comienza por un dígito. No válido porque contiene el carácter especial #.

1.3)Válido.

1.4)No válido porque es una palabra reservada del lenguaje.

1.5)Válido.

1.6)inVálido no se puede usar por la letra ñ

1.7)No válido, no puede tener espacios el identificador.

1.8)Válido.

1.9)No válido porque contiene el carácter -.

1.10)Válido.

1.11)No válido porque contiene el carácter '.

1.12)Válido.

1.13)Identificador no válido porque comienza por un dígito.

1.14)invalido(no se puede por la tilde

1.15)No válido, (%) es un carácter especial y no se puede usar en identificadores.

1.16)Válido.

1.17)invalida(tiene tilde)

1.18)invalida(tiene tilde)

1.19)invalida(tiene tilde)

1.20)No válido, no se aceptan puntos en los identificadores.

1.21)Válido.

1.22)No es válido, (?) no está permitido en los identificadores.

1.23)Válido

1.24)Válido.

25)invalida (no se puede usar ç)

26) invalida ( ü no se puede usar)

1.27)Válido.

1.28)Válido.

1.29)Identificador no válido porque comienza por un dígito.

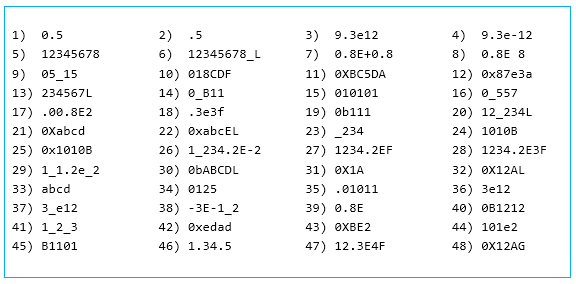
1.30)No valido, (&) es un carácter especial, los identificadores no pueden tener caracteres especiales.

1.31)Identificador no válido porque comienza por un dígito.

1.32)No válido porque es una palabra reservada del lenguaje.

1. **Ejercicio sobre literales: indica qué literales Java son válidos**.

Indica cuáles de los siguientes literales son válidos en Java. Si el literal es válido indica además de qué tipo es (int, double, long, etc) y el sistema de numeración en el que está escrito (decimal, binario, octal, hexadecimal). Si el literal no es válido explica por qué no lo es.



2.1)double/decimal.

2.2)double/decimal.

2.3)double/decimal.

2.4)double/decimal.

2.5)int/decimal.

2.6)No válido, (\_)mal posicionado.

2.7)No válido para el exponente en un literal Java tampoco puede contener decimales.

2.8)No es válido, hay un espacio entre E y 8.

2.9)int/octal.

2.10)No válido, no puede contener letras si se quiere poner del tipo octal.

2.11)int/hexadecimal.

2.12)int/hexadecimal.

2.13)long/decimal.

2.14)No válido, no puede estar el (\_) entre 0 y B.

2.15)int/octal.

2.16)No válido, no puede estar el (\_) al lado de un número.

2.17)No valido, solo puede tener un (.) que separe la parte decimal de la entera.

2.18)float/decimal.

2.19)int/binario.

2.20)long/decimal.

2.21)int/hexadecimal.

2.22)long/hexadecimal.

2.23)No válido, no puede estar el (\_) al inicio y tampoco al lado de un número.

2.24)No válido, debe empezar por 0b.

2.25)int/hexadecimal.

2.26)double/decimal.

2.27)No válido, falta el exponente entre E y F.

2.28)float/decimal.

2.29)No válido, no puede estar el(\_) entre e y 2.

2.30)No válido, debería ser seguido el 0b por 1 y 0.

2.31)int/hexadecimal.

2.32)long/hexadecimal.

2.33)No es válido, no puede contener caracteres especiales como a,b,c,d.

2.34)int/octal.

2.35)double/decimal.

2.36)double/decimal.

2.37)No válido, no puede estar el(\_) al lado de e.

2.38)double/decimal.

2.39)No válido, falta exponente de e

2.40)No válido, es binario sólo puede tener 0 y 1.

2.41)int/decimal.

2.42)int/hexadecimal.

2.43)int/hexadecimal.

2.44)double/decimal.

2.45)No válido, debería empezar por 0b para ser binario.

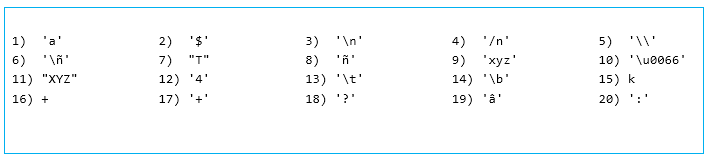
2.46)No válido, solo puede contener un (.).

2.47)float/decimal.

2.48)No es válido, porque g no es un carácter válido para hexadecimal.

1. **Ejercicio sobre literales de tipo char.**

Indica cuáles de los siguientes literales de tipo char son válidos en Java. Si el literal no es válido explica el motivo.



3.1)Válido.

3.2)Válido.

3.3)Válido.

3.4)No válido solo puede estar formado por carácter, excepto cuando es secuencia de escape.

3.5)Válido.

3.6)No válido, no está reconocido y no tiene un valor asignado.

3.7)No válido, por la doble comilla.

3.8)Válido.

3.9)Válido.

3.10)Válido.

3.11)No válido, por la doble comilla.

3.12)Válido.

3.13)Válido.

3.14)Válido.

3.15)No válido, debe ir entre comillas’.

3.16)No válido, debe ir entre comillas ‘.

3.17)Válido.

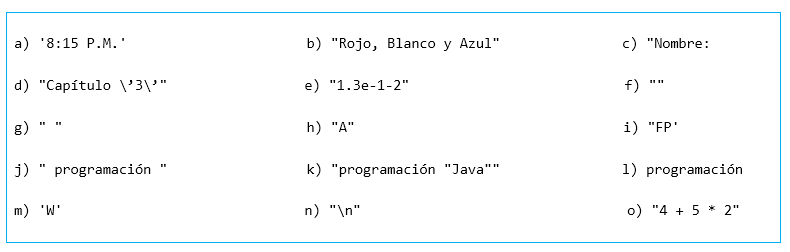
3.18)Válido.

3.19)Válido.

3.20)Válido.

1. **Ejercicio sobre literales de tipo String.**

Indica cuáles de los siguientes literales de tipo String son válidos en Java. Si el literal no es válido explica el motivo.



4.a)No válido, debe tener comillas dobles para ser string.

4.b)Válido.

4.c)No válido, le falta las comillas del final.

4.d)Válido

4.e)Válido.

4.f)Valido.

4.g)Válido.

4.h)Válido.

4.i)No valido, la comilla del final debe ser doble.

4.j)Válido.

4.k)No válido, las comillas interiores deberían ser tipo escape.

4.l)No válido, no tiene ninguna comilla.

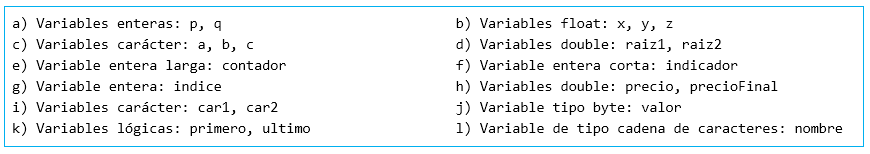
4.m)No válido, es un carácter.

4.n)Válido.

4.o)Validó.

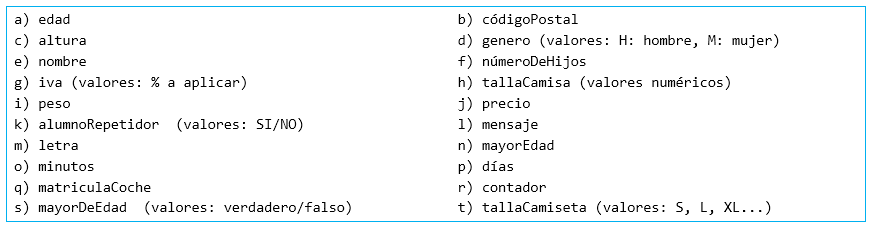
1. Ejercicio sobre declaraciones de variables

**Ejercicio 1**: Escribe la declaración para cada una de las variables y asignarles un valor inicial en la propia declaración de variable.





**Ejercicio 2**: Escribe la declaración más apropiada para cada una de las siguientes variables. El nombre de cada una indica el tipo de dato que contendrá y servirá para determinar el tipo de dato más adecuado en la declaración de la variable.

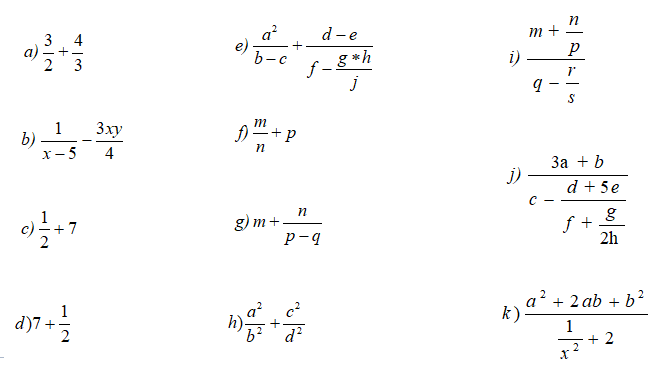


Texto

Descripción generada automáticamente;

6. Convertir expresiones algebráicas en expresiones algorítmicas

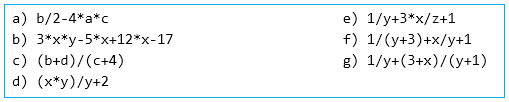
**Ejercicio:** Convierte en expresiones algorítmicas las siguientes expresiones algebráicas. Coloca paréntesis solamente donde sean necesarios.



1. 3/2 + 4/3
2. 1/ (x-5) - 3\*x\*y / 4
3. ½ + 7
4. 7 + ½
5. a\*a / (b-c) + (d-e) / (f -g\*h / j )
6. m/n + p
7. m + n / (p-q)
8. a\*a / (b\*b) + c\*c / (d\*d)
9. (m+n/p) / (q-r/s)
10. (3\*a+b) / (c-(d + 5 \* e) / (f+ g / (2\*h ))
11. (a\*a + 2\*a\*b + b\*b ) / (1 /( x\*x )) + 2

7. Convertir expresiones algorítmicas en expresiones algebráicas

**Ejercicio:** Convierte en expresiones algebráicas las siguientes expresiones algorítmicas.



1. b/2 -4ac
2. 3xy - 5x + 12x - 17
3. b+d / c+4
4. xy / y + 2
5. 1/y + 3x / z+1
6. ( 1 / (y + 3) + x / y ) + 1
7. 1 / y + ( 3 + x) / (y + 1)

**Ejercicio 2:**Si x, y, z son variables de tipo double con valores x= 88, y = 3.5, z = -5.2, determina el valor de las siguientes expresiones aritméticas. Obtén el resultado de cada expresión con un máximo de cuatro decimales.

|  |
| --- |
| a) x + y + z b) 2 \* y + 3 \* (x – z)  c) x / y d) x % y  e) x / (y + z) f) (x / y) + z  g) 2 \* x / 3 \* y h) 2 \* x / (3 \* y)  i) x \* y % z j) x \* (y % z)  k) 3 \* x – z – 2 \* x l) 2 \* x / 5 % y  m) x - 100 % y % z n) x - y - z \* 2 |

1. 86.3
2. 286.6
3. 25.1428
4. 0.5
5. 51.76
6. 19.9428
7. 205.3333
8. 16.7619
9. -59.2307
10. .
11. 93.2
12. 0.2
13. 88.182
14. 94.9

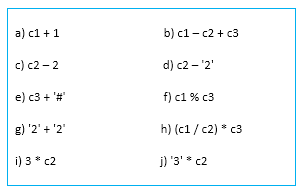
**Ejercicio 3.**Si c1, c2 y c3 son variables de tipo char con valores c1=’E’, c2=’5’, c3=’?’, determina el valor numérico de las siguientes expresiones aritméticas. Para resolverlo necesitas saber el valor numérico correspondiente a esos caracteres según la tabla ASCII:

’E’      69

’5’      53

’?’      63

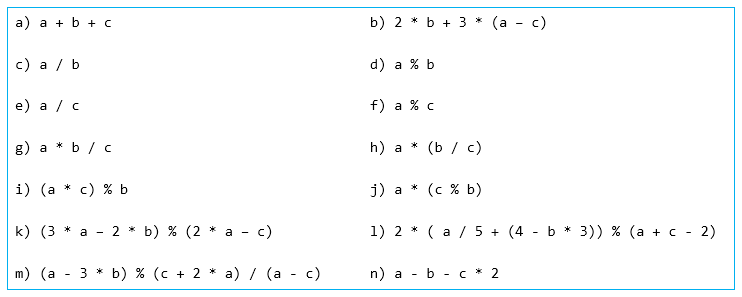
Consulta en la tabla ASCII los valores numéricos del resto de caracteres que aparecen en las operaciones.



1. 70
2. 79
3. 51
4. 3
5. 98
6. 6
7. 100
8. 63
9. 159
10. 2703

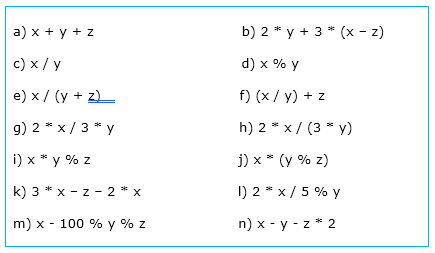
8. Ejercicio sobre operadores aritméticos

**Ejercicio 1:** Si a, b y c son variables enteras con valores a=8, b=3, c=-5, determina el valor de las siguientes expresiones aritméticas:



1. 6
2. 45
3. 2
4. 2
5. -1
6. 3
7. -4
8. 0
9. -1
10. 16
11. 18
12. 0
13. 0
14. 15

**Ejercicio 2:**Si x, y, z son variables de tipo double con valores x= 88, y = 3.5, z = -5.2, determina el valor de las siguientes expresiones aritméticas. Obtén el resultado de cada expresión con un máximo de cuatro decimales.



1. 86.3
2. 286.6
3. 25.1428
4. 0.5
5. 51.76
6. 19.9428
7. 205.3333
8. 16.7619
9. -59.2307
10. .
11. 93.2
12. 0.2
13. 88.182
14. 94.9

**Ejercicio 3.**Si c1, c2 y c3 son variables de tipo char con valores c1=’E’, c2=’5’, c3=’?’, determina el valor numérico de las siguientes expresiones aritméticas. Para resolverlo necesitas saber el valor numérico correspondiente a esos caracteres según la tabla ASCII:

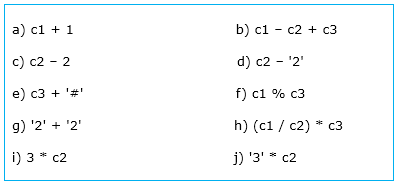
’E’      69

’5’      53

’?’      63

Consulta en la tabla ASCII los valores numéricos del resto de caracteres que aparecen en las operaciones.

1. 70
2. 79
3. 51
4. 3
5. 98
6. 6
7. 100
8. 82.01
9. 159
10. 2703



9. Determinar el tipo del resultado en operaciones aritméticas

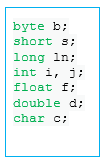
En aquellas expresiones aritméticas en las que intervienen operandos de distinto tipo (int, double, char, etc.) java convierte el tipo de los operandos que intervienen al tipo del operando de mayor precisión y este será el tipo del resultado obtenido en la operación.

Esta conversión de tipos se realiza de forma temporal, solamente para el cálculo de la operación. Los tipos originales de los operandos que intervienen siguen siendo los mismos después de la operación.

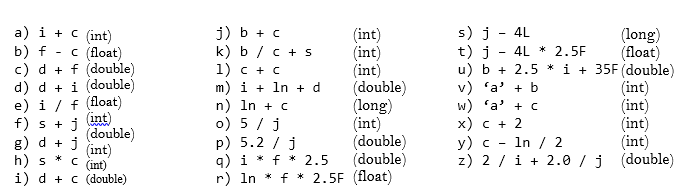
Debemos tener en cuenta que cuando se realiza la conversión temporal de tipos, los operandos de tipo short, byte y char se convierten a int.

Teniendo en cuenta todo esto, realiza el siguiente ejercicio.

**Ejercicio**: A partir de las siguientes declaraciones de variables:

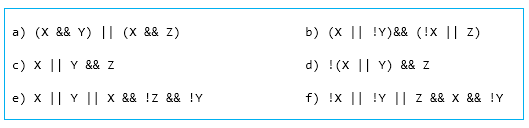


Determina cuál es el tipo de dato del resultado de las siguientes expresiones aritméticas:



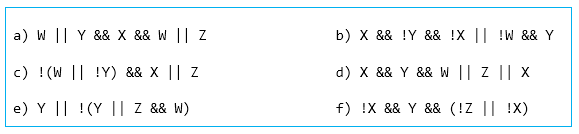
10. Ejercicio sobre operadores lógicos

**Ejercicio 1:** Si X, Y y Z son variables de tipo boolean con valores X = true, Y = false, Z = true, determina el valor de las siguientes expresiones lógicas:



1. true
2. true
3. true
4. false
5. true
6. true

**Ejercicio 2:**Si W, X, Y y Z son variables de tipo boolean con valores W = false, X = true, Y = true, Z = false, determina el valor de las siguientes expresiones lógicas:



1. falso
2. true
3. true
4. true
5. true
6. falso

Para resolver ejercicios con operadores lógicos tienes que conocer las tablas de verdad de estos operadores:

|  |  |  |
| --- | --- | --- |
| Operador AND | Operador OR | Operador NOT |
|  |  |  |

Como puedes observar:

* El operador lógico && (AND) da como resultado **true** cuando ambos operandos son true, en cualquier otro caso el resultado es false.
* El operador || (OR) da como resultado **false** cuando ambn cualquier otro caso el resultado es true.
* El operador ! (NOT) cambia el valor lógico del operando.

También debemos tener en cuenta para resolver estos ejercicios cuál es la precedencia de los operadores lógicos en Java:

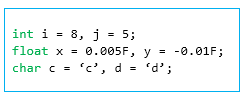
|  |
| --- |
| Precedencia de los operadores lógicos |
|  |

11. Ejercicios sobre operadores relacionales

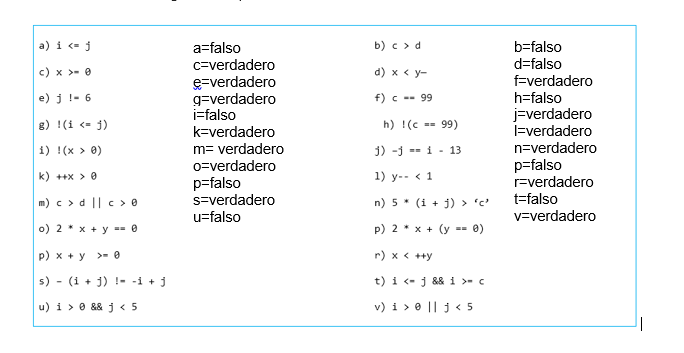
En estos ejercicios practicaremos con los operadores Java relacionales. En algunos de los ejercicios, los operadores relacionales aparecerán en expresiones en los que también intervienen el resto de operadores Java: operadores aritméticos, lógicos y los operadores incremento y decremento.

**Ejercicio:**

A partir de las siguientes declaraciones de variables:



Determina el valor de las siguientes expresiones:

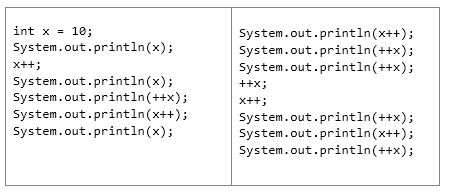


**12. Ejercicio con los operadores incremento y decremento: ++ y --.**

**Ejercicio 1:** Si x es una variable de tipo int con valor 10, determina qué se muestra por pantalla cuando se ejecutan las siguientes instrucciones:

x++ : imprime la variable y después incrementa

++X : incrementa y después imprime



int x = 10;

System.out.println(x); 10

x++;

System.out.println(x); 11

System.out.println(++x); 12

System.out.println(x++); 12

System.out.println(x); 13

System.out.println(x++); 13

System.out.println(++x); 15

System.out.println(++x); 16

++x;

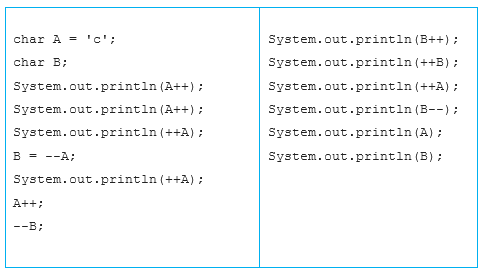
x++;

System.out.println(++x); 19

System.out.println(x++); 19

System.out.println(++x); 21

**Ejercicio 2:**Dadas las variables A y B de tipo char,calcula qué se muestra por pantalla cuando se ejecutan las siguientes instrucciones**:**



char A = 'c';

char B;

System.out.println(A++); c

System.out.println(A++); d

System.out.println(++A); f

B = --A;

System.out.println(++A); g

A++;

--B;

System.out.println(B++); d

System.out.println(++B); f

System.out.println(++A); h

System.out.println(B--); f

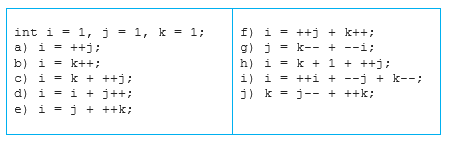
System.out.println(A); h

System.out.println(B); e

**Ejercicio 3:**Un programa Java contiene las siguientes declaraciones y asignaciones iniciales:

**int i = 1, j = 1, k = 1;**

Determina el valor final de las tres variables en cada una de las siguientes instrucciones. Las instrucciones son independientes unas de otras, es decir, el valor inicial de las variables en cada instrucción es i = 1, j = 1, k = 1.



int i = 1, j = 1, k = 1;

a) i = ++j; i=2 j=2 k=1

b) i = k++; i=1 j=2 k=2

c) i = k + ++j; i=3 j=2 k=1

d) i = i + j++; i=2 j=2 k=1

e) i = j + ++k; i=3 j=1 k=2

f) i = ++j + k++; i=3 j=2 k=2

g) j = k-- + --i; i=0 j=1 k=0

h) i = k + 1 + ++j; i=4 j=2 k=1

i) i = ++i + --j + k--; i=3 j=0 k=0

j) k = j-- + ++k; i=1 j=0 k=3

**13. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println – I**

Relación Nº 1: Ejercicios 1 y 2  
  
Empezamos con una serie de ejercicios para aquellos que están comenzando desde cero con el lenguaje Java y tampoco tienen experiencia con ningún otro lenguaje de programación.

En estos ejercicios básicos inicales se realizan las siguientes instrucciones: declarar variables, asignarles un valor, operar con ellas y mostrar resultados por pantalla.

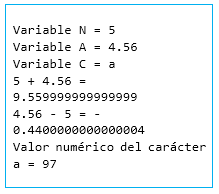
Estos ejercicios son todos de estructura secuencial, es decir, no hay condiciones ni bucles. Tampoco se lee nada por teclado. El objetivo es familiarizarse con la declaración de variables y practicar la salida por consola utilizando los métodos print y println.

**Ejercicio básico inicial 1**:

Escribe un programa Java que realice lo siguiente: declarar una variable N de tipo int, una variable A de tipo double y una variable C de tipo char y asigna a cada una un valor. A continuación muestra por pantalla:

* El valor de cada variable.
* La suma de N + A
* La diferencia de A - N
* El valor numérico correspondiente al carácter que contiene la variable C.

Si por ejemplo le hemos dado a N el valor 5, a A el valor 4.56 y a C el valor ‘a’, se debe mostrar por pantalla:



public class App {

public static void main(String[] args) throws Exception {

int N = 5;

double A = 4.56;

char c = 'a';

double suma = N + A;

double diferencia = N - A;

System.out.println("El valor de N: "+ N);

System.out.println("El valor de A: " + A);

System.out.println("El valor de c: " + c);

System.out.print("La suma es: ");

System.out.println(suma);

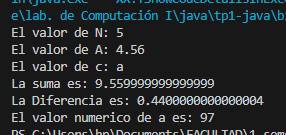
System.out.print("La Diferencia es: ");

System.out.println(diferencia);

System.out.print("El valor numerico de a es: " + (int)c);

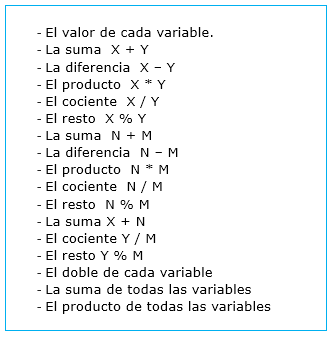
}

}

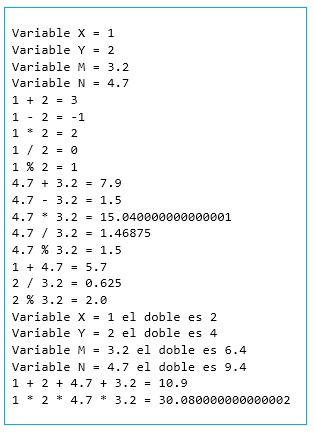


**Ejercicio básico inicial 2**:

Escribe un programa Java que realice lo siguiente: declarar dos variables X e Y de tipo int, dos variables N y M de tipo double y asigna a cada una un valor. A continuación muestra por pantalla:



Si por ejemplo le hemos dado a X el valor 1, a Y el valor 2, a M el valor 3.2 y a N el valor 4.7 se debe mostrar por pantalla:



public class App {

public static void main(String[] args) throws Exception {

int X = 1, Y = 2;

double N =4.7 , M = 3.2;

System.out.println("Variable X = "+ X);

System.out.println("Variable Y = "+ Y);

System.out.println("Variable N = "+ N);

System.out.println("Variable M = "+ M);

System.out.println(X +" + "+ Y + " = "+ (X+Y));

System.out.println(X +" - "+ Y + " = "+ (X-Y));

System.out.println(X +" \* "+ Y + " = "+ (X\*Y));

System.out.println(X +" / "+ Y + " = "+ X/Y);

System.out.println(X +" % "+ Y + " = "+ X%Y);

System.out.println(N +" + "+ M + " = "+ (N+M));

System.out.println(N +" - "+ M + " = "+ (N-M));

System.out.println(N +" \* "+ M + " = "+ (N\*M));

System.out.println(N +" / "+ M + " = "+ (N/M));

System.out.println(N +" % "+ M + " = "+ N%M);

System.out.println(X + " + " + N + " = " + (X+N));

System.out.println(Y + " / " + M + " = " + Y/M);

System.out.println(Y + " % " + M + " = " + Y%M);

System.out.println("Variable X = " + X + " el doble es " + 2\*X);

System.out.println("Variable Y = " + Y + " el doble es " + 2\*Y);

System.out.println("Variable M = " + M + " el doble es " + 2\*M);

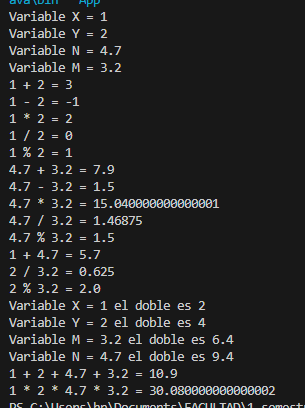
System.out.println("Variable N = " + N + " el doble es " + 2\*N);

System.out.println(X + " + " + Y + " + " + N + " + " + M + " = " + (X+Y+M+N));

System.out.println(X + " \* " + Y + " \* " + N + " \* " + M + " = " + (X\*Y\*M\*N));

}

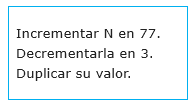
}



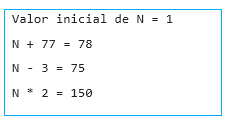
**14. Ejercicios para practicar la salida de datos por pantalla: System.out.print y System.out.println – II**

**Ejercicio básico inicial 3**

Escribe un programa Java que declare una variable entera N y asígnale un valor. A continuación escribe las instrucciones que realicen los siguientes:



Si por ejemplo N = 1 la salida del programa será:



int n=5;

n+=77;

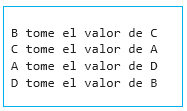
n-=3;

n\*=2;

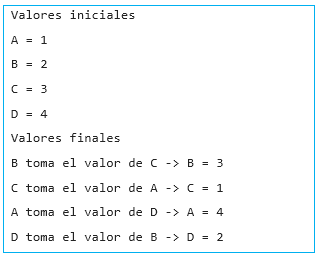
System.out.print("El numero es: "+n);

**Ejercicio básico inicial 4**

Programa java que declare cuatro variables enteras A, B, C y D y asígnale un valor a cada una. A continuación realiza las instrucciones necesarias para que:



Si por ejemplo A = 1, B = 2, C = 3 y D = 4 el programa debe mostrar:



int a=1;

int b=2;

int c=3;

int d=4;

int suma=a+b+c+d;

System.out.print("el valor inicial de a es: "+a+" el valor inicial de b es: "+b+" el valor inicial de c es: "+c+" el valor inicial de es: "+d);

a=d;

d=b;

b=c;

c=(suma)-a-d-b;

System.out.print("el valor de a es: "+a+" el valor de b es: "+b+" el valor de c es: "+c+" el valor de es: "+d);

15. Ejercicios con el operador condicional ( ? : ) - I

En esta entrada vamos a ver tres ejemplos de utilización del operador condicional ? :  
Se trata de usar el operador condicional en lugar de la instrucción condicional if para mostrar por pantalla un mensaje u otro dependiendo de una condición.  
  
**Ejercicio básico inicial 5**

Escribe un programa java que declare una variable A de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si A es par o impar. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo A = 14 la salida será



Si fuese por ejemplo A = 15 la salida será:



int a=24;

System.out.print(a + (a%2==0? "es par": "es impar"));

**Ejercicio básico inicial 6**

Escribe un programa java que declare una variable B de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si el valor de B es positivo o negativo. Consideraremos el 0 como positivo. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo B = 1 la salida será



Si fuese por ejemplo B = -1 la salida será:



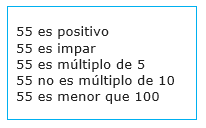
int a=-24;

System.out.print(a + a>=0? "es positivo": "es negativo");

**Ejercicio básico inicial 7**

Escribe un programa java que declare una variable C de tipo entero y asígnale un valor. A continuación muestra un mensaje indicando si el valor de C es positivo o negativo, si es par o impar, si es múltiplo de 5, si es múltiplo de 10 y si es mayor o menor que 100. Consideraremos el 0 como positivo. Utiliza el operador condicional ( ? : ) dentro del println para resolverlo.

Si por ejemplo C = 55 la salida será



int a=120;

System.out.println(a + (a>=0? " es positivo": " es negativo"));

System.out.println(a + (a%2==0? " es par": " es impar"));

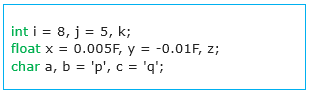
System.out.println(a + (a%5==0? " es multiplo de 5": " no es multiplo de 5"));

System.out.println(a + (a%10==0? " es multiplo de 10": " no es multiplo de 10"));

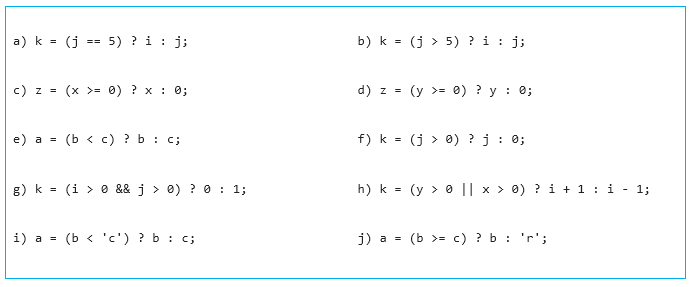
System.out.println(a + (a>100? " es mayor que 100": " es menor que 100"));

16. Ejercicios con el operador condicional ( ? : ) – II

**Ejercicio 1:**Un programa contiene las siguientes declaraciones y asignaciones iniciales de variables:



Determina el valor de cada una de las siguientes expresiones:

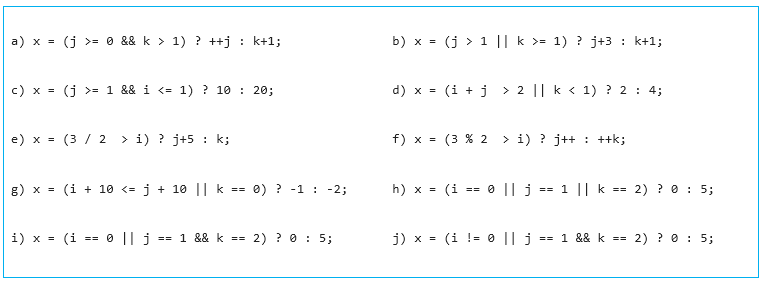


1. **k-8**
2. **k-5**
3. **z-0.005f**
4. **z-0**
5. **a-p**
6. **k-5**
7. **k-0**
8. **k-9**
9. **a-q**
10. **‘r’**

**Ejercicio 2:**Un programa contiene las siguientes declaraciones y asignaciones iniciales de variables:

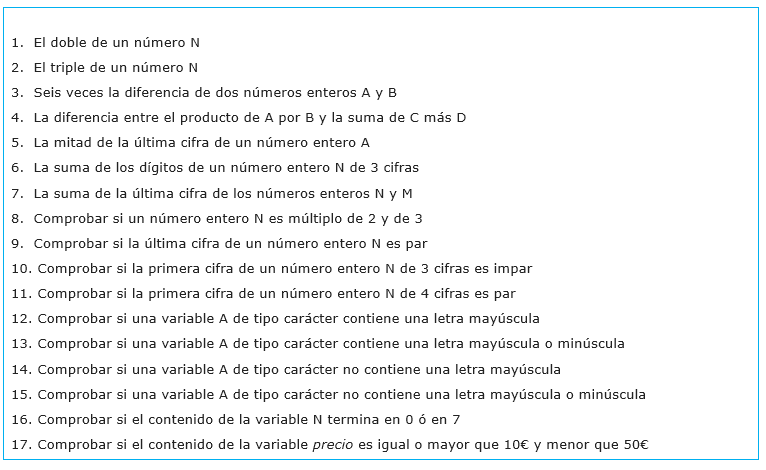


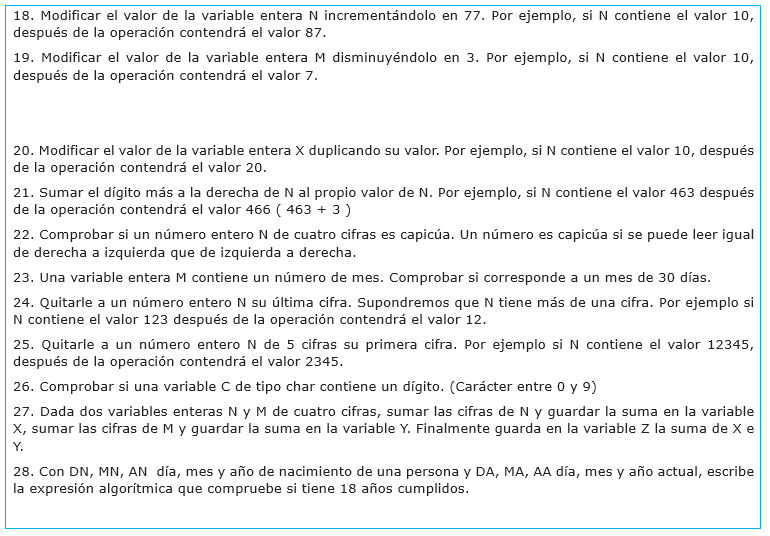
Determina el valor que toma la variable x al ejecutarse cada una de las siguientes expresiones. Las instrucciones son independientes unas de otras, es decir, el valor inicial para las variables i, j, k en cada instrucción es i = 1, j = 1, k = 1.



1. x-2
2. x-4
3. x-10
4. x-4
5. x-1
6. x-2
7. x- -1
8. x-0
9. x-5
10. x-5

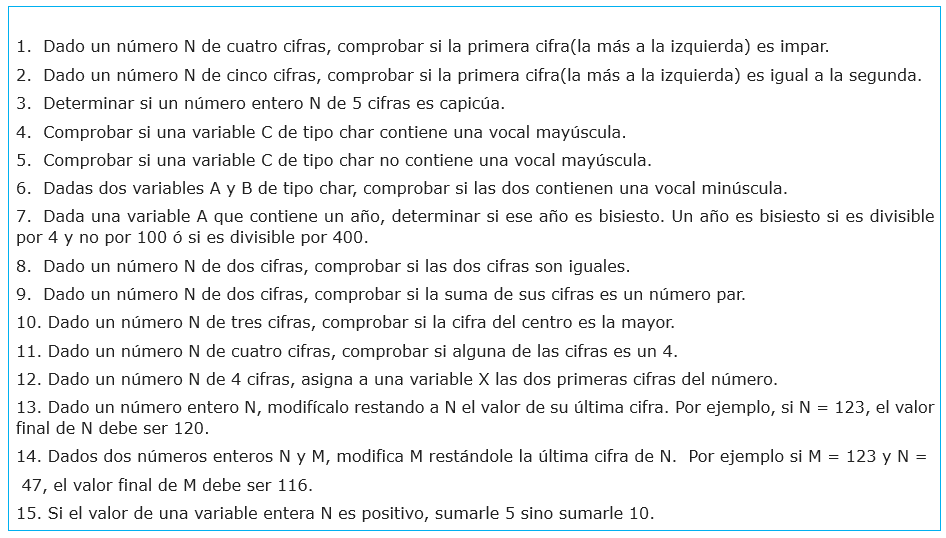
17. Ejercicios sobre expresiones algorítmicas





1. 2\*n
2. 3\*n
3. 6 \* (a-b)
4. a\*b - (c+d)
5. a%10 /2
6. n/100 + n%10 +n/10%10
7. n%10 + m%10
8. n%2 == 0 && n%3==0
9. n%10%2 ==0
10. n/100 % 2!=0
11. n/1000%2==0
12. A>=’A’ && A<= ‘Z’
13. A>=’A’ && A<=’Z’ || A>=’a’ && A<=’z’
14. A<’A’|| A>’Z’
15. (A < ’A’ || A > ’Z’) && (A < ’a’ || A > ’z’)
16. N%10 == 0 || N%10 == 7
17. precio >= 10 && precio < 50
18. N = N +77 ó N+=77
19. N = N – 3 ó N-=3
20. N = N \* 2 ó N\*=2
21. N = N + N%10 ó N+=N%10
22. N/1000 == N%10 && N/100%10 == N/10%10
23. m == 4 || m == 6 || m == 9 || m == 11
24. n = n/10
25. n = n % 10000
26. c >= ‘0’ && c <= ‘9’
27. X = n/1000 + n/100%10 + n/10%10 + n%10; Y = m/1000 + m/100%10 + m/10%10 + m%10; Z = X + Y;
28. AA – AN > 18 || AA – AN == 18 && MA > MN || AA – AN == 18 && MA == MN && DA >= DN

18. Ejercicios sobre expresiones algorítmicas – II

Escribe las expresiones algorítmicas equivalentes para cada uno de los siguientes enunciados:

18.1) n / 1000 % 2 == 0

18.2) n / 10000 == n / 1000 % 10

18.3) n / 10000 == n % 10 && n / 1000 % 10 == n / 10 % 10

18.4) C == ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’

18.5) !(C= ‘A’ || C == ‘E’ || C == ‘I’ || C == ‘O’ || C == ‘U’)

18.6) (A == ‘a’ || A == ‘e’ || A == ‘i’ || A == ‘o’ || A == ‘u’) && (B == ‘a’ || B == ‘e’ || B == ‘i’ || B == ‘o’ || B == ‘u’)

18.7) A % 4 == 0 && A % 100 != 0 || A % 400 == 0

18.8) N / 10 == N % 10

18.9) (N / 10 + N % 10) % 2 == 0

18.10) N / 10 % 10 > N / 100 && N / 10 % 10 > N % 10

18.11) N / 1000 == 4 || N / 100 % 10 == 4 || N / 10 % 10 == 4 || N % 10 == 4

18.12) X = N / 100

18.13) N = N – N % 10 o también N -= N % 10

18.14) M = M - N % 10 o también M -= N % 10

18.15) N = N + (N > 5 ? 5 : 100)