

Laboratori Programació I, 2019-20  
**Exercicis d'Operadors i expressions**  
Grau d'Enginyeria Informàtica



UNIVERSITAT DE  
BARCELONA

Nom i cognoms: Oscar De Caralt Roy

Grup de laboratori: MC

Aquests exercicis tracten sobre expressions, operadors i conversió. Les expressions aritmètiques i lògiques són força utilitzades, per sumar, restar, comprovar si alguna condició és certa o falsa... Aquestes expressions utilitzen operadors, i els tenim que conèixer. Els operadors tenen un ordre de precedència – és un ordre d'importància, en el sentit que un operador es realitza primer o després que un altre operador. Això vol dir que una divisió és més important que una suma? Més que important, una es computa primer que una altra. Aquest ordre no sempre és intuïtiu, i ens pot donar resultats inesperats! Ens toca estudiar la precedència dels operadors. Davant el dubte amb la precedència, sempre s'aconsella utilitzar parèntesis. Som-hi!!

**Aquest llistat de problemes l'heu de lliurar (pel Campus Virtual) el dia abans del dia de la prova P1. La data de lliurament varia dependent del grup.**

Aquells alumnes que vulguin imprimir els nombres formatats (amb, per exemple, només dos decimals), poden llegir la documentació:

<https://docs.oracle.com/javase/tutorial/java/data/numberformat.html> i modificar el codi donat amb noves crides a mètodes.

Per realitzar aquests exercicis et resultarà molt útil consultar (i estudiar) la taula que pots trobar en aquesta pàgina web:

<http://introc.cs.princeton.edu/java/11precedence/>

**Exercici 1.** A continuació es mostra un programa JAVA a on es realitzen diferents operacions amb variables de diferents tipus.

```
//Expressions1.java
public class Expressions1{
    public static final float EPSILON = 0.000001 f;

    public static void main(String[] args) {
        int a, b, c, d, e, i;
        float x, z, u, f, g, h;
        boolean sonIgualsF, sonIgualsD;
        char car1 = 'b', car2;

        a = 10 / 3 + 2;
        b = 3 * 10 / 7;
        c = 3 / 10 * 7;

        System.out.println("a=" + a + "b=" + b + "c=" + c);

        z = 10e3;
        u = 10e-3;
        x = 1000 * (2 / 3 + 1 / 3 - 1);
        System.out.println("z=" + z + " u=" + u + " x=" + x);

        b = b + 1 / b;
        x = 1 / x;
        a = (int) x;
        z = 1. / a;
        System.out.println("b=" + b + " x=" + x + " a=" + a + "
z=" + z);

        a = 50 % b;
        b = 678 % 10;
        c = 67 % 10;
        d = 6 % 10;
        e = 1 / 0;

        System.out.println("a=" + a + " b=" + b + " c=" + c + "
d=" + d + " e=");

        f = 0.0f;
        f += 0.02f;
```

```

        f += 0.03f;

        sonIgualsF = f == 0.05f;
        System.out.println("sonIgualsF = " + sonIgualsF);

        sonIgualsF = Math.abs(f - 0.05f) < EPSILON;
        System.out.println("sonIgualsF = " + sonIgualsF);

        g = 1.000001;
        h = 0.000001;

        sonIgualsD = (g-h) == 1.0;
        System.out.println("sonIgualsD = " + sonIgualsD);

        sonIgualsD = Math.abs((g-h) - 1.0) < EPSILON;
        System.out.println("sonIgualsD = " + sonIgualsD);

        car2 = (char) (car1 + 1);
        System.out.println("car2 = " + car2);

        i = car2 + 1;
        System.out.println("i=" + i + " i= " + (char) i);
    }
}

```

a) Abans d'executar el programa, examina-ho atentament i omple els resultats que penses que mostrarà el programa quan s'executi. Toma't el teu temps, és un exercici una mica llarg.

a = 5   b = 4   c = 0

z = 10000.0   u = 0.01   x = -1000.0

b = 4   x = 0.0   a = 0   z = ∞

a = 2   b = 8   c = 7   d = 6   e = error d'execució

sonIgualsF = False

sonIgualsF = True

sonIgualsD = False

sonIgualsD = True

car2 = no sé

i = no sé   i = no sé

b) Executa el programa (si n'hi han errors de compilació, ja saps el que et toca fer, efectivament, corregir-los) i escriu els resultants obtinguts del programa. Analitza les possibles diferències amb el que has escrit en l'apartat anterior.

a = 5   b = 4   c = 0

z = error, és un float i hauria de ser un double  
ser un double   x = -1000.0

u = error, és un float i hauria de

b = 4   x = 0.0   a = 0   z = infinity

a = 2   b = 8   c = 7   d = 6   e = command execution failed

sonIgualsF = False

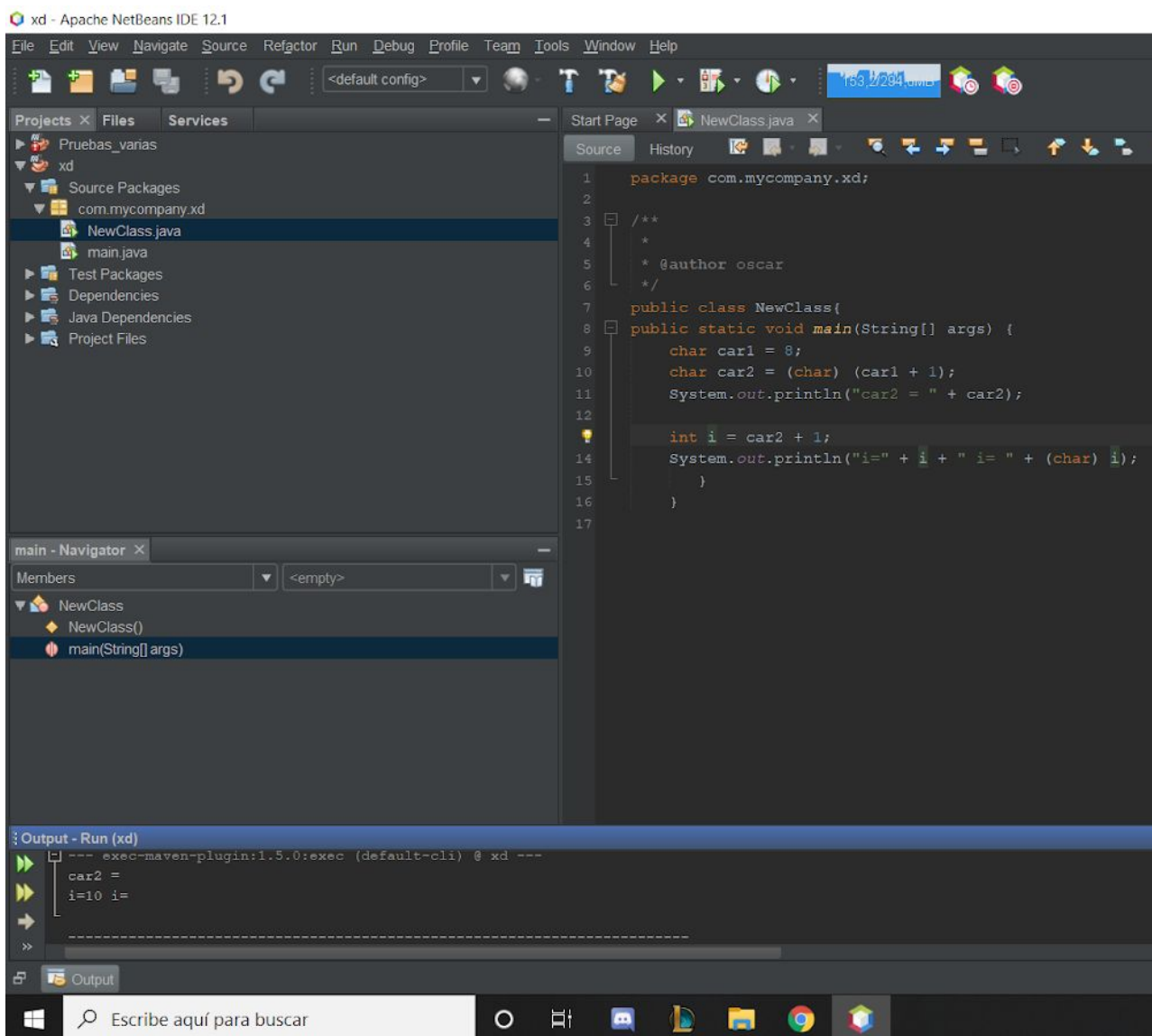
sonIgualsF = True

sonIgualsD = False

sonIgualsD = True

car2 =

i = 10   i =



**Exercici 2.** A continuació s'indiquen diversos grups d'instruccions en les quals una expressió aritmètica s'assigna a una variable. Algunes d'elles tenen una sintaxis incorrecte, altres retornen valors que poden ser inesperats. Et proposem que, sense incorporar les instruccions a un programa, emplenis a mà la taula següent preveient els valors que finalment quedaran dins de les variables que es demanen. Després ho comprovaràs executant el programa!

Instruccions	Correctes (Si/No)	Valor previst
int a = 41, b = 3, c = 2, d; d=a/b/c;	Si	d= 6
int a = 41, b = 3, c = 2, d; d = a / (b / c) ;	Si	d= 41
double x = 41, y = 3, z = 2, u; u = x / ( y / z) ;	Si	u = 27.3
int i = 2, j = 2; i = i * (i + j) ;	Si	i = 8
int i = 3; float x = -1.75; i=i-x;	No	i = Error
int i = 8, k=11; i = i % (k - 3);	Si	i = 0
int a, b = 10; a = (b/ 3) + 2;	Si	a = 5
int a, b = 10; a = b + 3 / 2;	Si	a = 11
double x = 1.5, y = 1.3, z; z = (int) x%(int) y;	Si	z = 0
int a, b; a = 10; b = (a == 1)	No	b = Error

Ara es tracta de veure si el que heu pensat és correcte i entendre el que passa. Escriu un programa amb les instruccions, seguint, si voleu, el següent esquema, i executa'l. Fes això per a cada fila de la taula anterior.

//Expressions2.java

```
public class Expressions2{
    public static void main (String[] args){
        int a = 41, b = 3, c = 2, d;
        d=a/b/c;
        System.out.println("El valor és: " + d);
    }
}
```

```

        //...
    }
}

```

**Exercici 3.** A la següent taula es presenten expressions JAVA que s’han de qualificar com: “Sintàcticament incorrecte”, “Sintàcticament correcte, però semànticament incorrecte”, “No avaluable”, “No assignable”, “Correcte”. Els valors de les variables que s’utilitzen a les expressions són:

**int** i = 10, j = 0;

**char** c = 'H';

**float** f = 15.5f;

**boolean** b = true;

Fes l’anàlisi de cada fila de la taula de manera separada.

Expressió	Qualificació
c = a’;	Sintàcticament incorrecte. Falta una cometa abans de la a.
f = i;	Correcte.
b = i && f;	Sintàcticament correcte, però semànticament incorrecte.
b = (i > j) && b;	Correcte
i = c + 2;	Correcte
c = c + “a”;	No assignable
i = f * i;	Sintàcticament incorrecte
i = i / j;	No assignable
b = ! (i=10)	Sintàcticament incorrecte
c = (char) (c-2)	Correcte