

**Programación 1**

**Problemas tema 3**

**Información, datos, operaciones y expresiones**

---

Expresiones en C++



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**

# Expresiones

---

- `<declaración> ::=`  
    `<tipo> <variable> “=”`  
    `<expresión> “;”`
- `<asignación> ::=`  
    `<variable> “=” <expresión> “;”`

# Expresiones

---

- Código que es evaluado para obtener un valor
- Combinación de constantes, variables, operadores y llamadas a funciones
  - $\emptyset$
  - grados
  - `sqrt(2.0)`
  - `pesetas / PTAS_POR_EURO`
  - `sin(M_PI * grados / 180.0)`



# Expresiones aritméticas en C++

---

1. Expresión que determine el área de un rectángulo cuyos lados miden ***x*** e ***y***.



# Hoy, estamos interesados solo en expresiones...

$x * y$

## ...no en programas completos

```
#include <iostream>
using namespace std;

/*
 * Programa que solicita al usuario la longitud de los lados de
 * un rectángulo y escribe en la pantalla el área del mismo.
 */
int main() {
    cout << "Introduzca las longitudes de los lados de un "
          << "rectángulo: ";
    double x, y;
    cin >> x >> y;

    cout << "El área del rectángulo es: " << x * y << endl;
}
```



# Ejercicios básicos en Moodle

---

## Ejercicios básicos sobre expresiones numéricas (tema 3)

Si tienes problemas comprendiendo los primeros programas que se han realizado en clase, intenta desarrollar los que se proponen a continuación. Puedes realizarlos en un editor en línea (como [cpp.sh](https://cpp.sh)) o en Visual Studio Code. En este último caso, recomendamos ubicar los ficheros de código fuente C++ correspondientes a los programas solicitados en un directorio denominado «ejercicios-basicos».

1. Escribe un programa que pida al usuario las longitudes de los dos lados de un rectángulo y muestre en la pantalla el valor del área de dicho rectángulo. Tienes una posible solución en la transparencia 4 de la presentación correspondiente a la clase de problemas sobre expresiones.
2. Escribe un programa que pida al usuario dos números enteros, siendo el primero menor o igual que el segundo, y escriba en la pantalla el número de enteros comprendidos entre ambos (inclusive). No es necesario que compruebes que el primero de los enteros introducidos por el usuario sea menor o igual que el segundo; basta con que le indiques esta restricción al usuario cuando se le soliciten.
3. Escribe un programa que pida al usuario la longitud del radio de una circunferencia y escriba en la pantalla la longitud de dicha circunferencia.
4. Escribe un programa que pida al usuario las dimensiones de una caja con forma de ortoedro (paralelepípedo ortogonal), que tendrán que ser todas ellas múltiplos de 10 cm. No es necesario que compruebes las dimensiones son múltiplos de 10 cm; basta con que le indiques esta restricción al usuario cuando se las solicites. El programa debe escribir luego en la pantalla el número de dados de lado 2 cm que pueden almacenarse en dicha caja.
5. Escribe un programa que escriba en la pantalla la longitud de hipotenusa de un triángulo rectángulo cuyos catetos se le han pedido previamente al usuario.
6. Escribe un programa pida al usuario un número de niños y un número de caramelos. A continuación, el programa escribe en la pantalla cuántos caramelos corresponden como máximo a cada uno de los niños, si los caramelos se reparten de forma equitativa entre ellos y cuántos caramelos



## Expresiones aritméticas en C++

---

2. Expresión que determine el número de enteros comprendidos entre los enteros ***x*** e ***y***, inclusive, siendo ***x*** ≤ ***y***.
3. Expresión que determine la longitud de una circunferencia de radio ***r***.



# Constante M\_PI

---

- `<cmath>`
  - `const double M_PI`  
`= 3.14159265358979323846;`
- No forma parte del estándar de C++
  - Está disponible en los compiladores GNU GCC Compiler y MinGW.
    - Podría ser que necesitáramos escribir antes `#define _USE_MATH_DEFINES` antes de `#include <cmath>`.
  - En otros compiladores, podría ser que tuviéramos que definirla nosotros.

## Expresiones aritméticas en C++

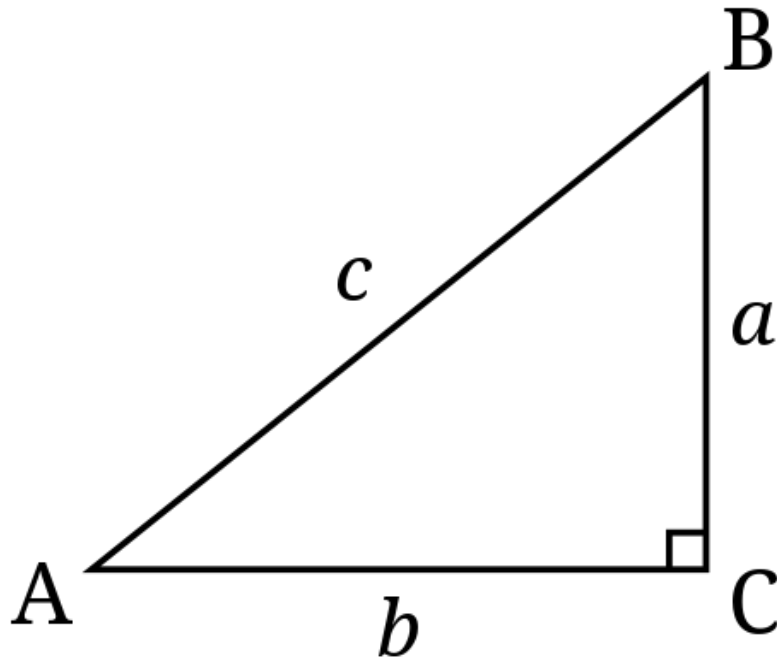
---

4. Expresión que determine cuántos dados de lado 2 cm pueden almacenarse en una caja con forma de ortoedro (paralelepípedo ortogonal) cuyas dimensiones son  **$x$** ,  **$y$** ,  **$z$** , todas ellas múltiplos de 10 cm.
5. Expresión que determine la longitud de hipotenusa de un triángulo rectángulo cuyos catetos miden  **$a$**  y  **$b$** .

# Triángulos rectángulos

$$\square c^2 = a^2 + b^2$$

$$\square c = \sqrt{a^2 + b^2}$$



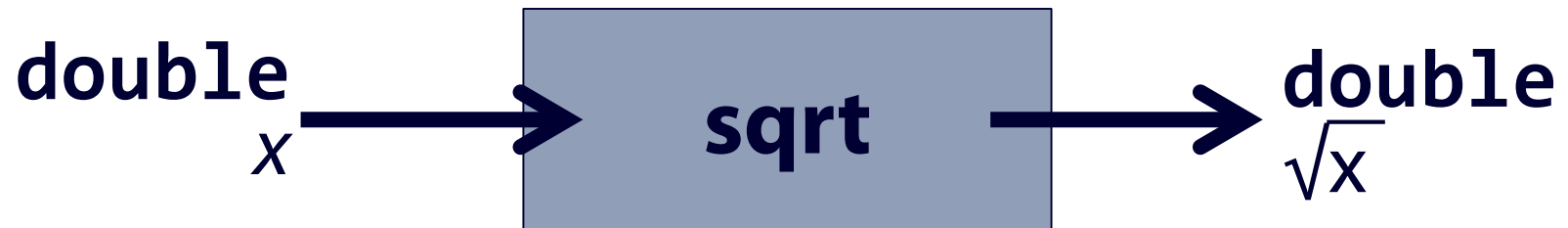
# Función sqrt

---

- `<cmath>`
- **`double sqrt (double x);`**
  - Compute square root
  - Returns the *square root* of *x*.
- Parameters
  - *x*
    - Value whose square root is computed.
    - If the argument is negative, a *domain error* occurs.
- Return Value
  - Square root of *x*.
  - If *x* is negative, a *domain error* occurs:
- Fuente
  - <http://www.cplusplus.com/reference/cmath/sqrt/>

# Función sqrt

---



# Función pow

---

- `<cmath>`
- **`double pow (double base, double exponent);`**
  - Raise to power
  - Returns *base* raised to the power *exponent*:  $\text{base}^{\text{exponent}}$
- Parameters
  - `base`
    - Base value.
  - `exponent`
    - Exponent value.
- Return Value
  - The result of raising base to the power exponent.
  - ...
- Fuente
  - <http://www.cplusplus.com/reference/cmath/pow/>

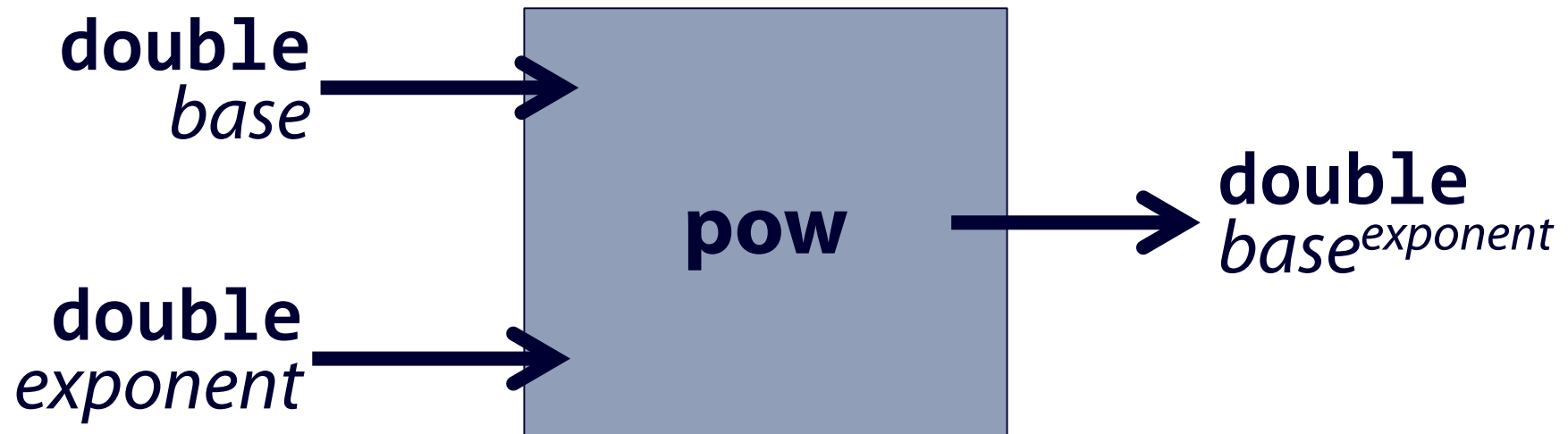
# Función pow

---

- ❑ `<cmath>`
- ❑ **`double pow (double base, double exponent);`**
- ❑ Return Value
  - ❑ The result of raising base to the power exponent.
  - ❑ If the base is finite negative and the exponent is finite but not an integer value, it causes a domain error.
  - ❑ If both base and exponent are zero, it may also cause a domain error on certain implementations.
  - ❑ If base is zero and exponent is negative, it may cause a domain error or a pole error (or none, depending on the library implementation).
  - ❑ The function may also cause a range error if the result is too great or too small to be represented by a value of the return type.
- ❑ Fuente
  - <http://www.cplusplus.com/reference/cmath/pow/>

# Función pow

---







# Expresiones aritméticas en C++

---

6. Expresión que determine el valor de la suma de los enteros comprendidos en el intervalo  $[x, y]$ , siendo  $x \leq y$ .

# Expresiones aritméticas en C++

7. Cuántos caramelos corresponden como máximo a cada uno de los  $n$  niños de un grupo, si repartimos un total de  $t$  caramelos entre ellos de modo equitativo.
8. Cuántos caramelos sobrarían tras hacer el reparto anterior.





# División entera

---

1 3 7 2 3 8

5 6 2

7 6 3

3 4 8

2 4

8 1

1 6 9 4



# Expresiones aritméticas en C++ con conversión de tipos

---

9. Expresión que determine el porcentaje de alumnos aprobados con respecto a los matriculados en una asignatura, siendo ***aprobados*** el número de alumnos aprobados y ***matriculados*** el número de alumnos matriculados en dicha asignatura. ¿Y la tasa de aprobados?
10. Expresión que redondee el valor de un dato real positivo ***x*** al entero más próximo.

# Función round

---

- `<cmath>`
- **`double round (double x);`**
  - **Round to nearest**
  - Returns the integral value that is nearest to  $x$ , with halfway cases rounded away from zero
- **Parameters**
  - $x$ 
    - Value to round
- **Return Value**
  - The value of  $x$  rounded to the nearest integral (as a floating-point value).
- **Fuente**
  - <https://www.cplusplus.com/reference/cmath/round/>

# Expresiones de relación y booleanas en C++

---

11. Expresión cuyo valor sea *cierto* si y solo si el valor de ***x*** es mayor o igual que el valor de ***y***.
12. Expresión cuyo valor sea *cierto* si y solo si el valor de ***a*** no es nulo.
13. Expresión cuyo valor sea *cierto* si y solo si los valores de ***a*** y ***b*** son iguales.

# Expresiones de relación y booleanas en C++

---

14. Expresión cuyo valor sea *cierto* si y solo si el valor de ***x*** es estrictamente mayor que la suma de los valores de ***a*** y ***b***.
15. Expresión cuyo valor sea *cierto* si y solo si el valor de ***x*** está comprendido en el intervalo [***a***, ***b***], con ***a*** ≤ ***b***.
16. Expresión cuyo valor sea *cierto* si y solo si el valor de ***x*** está fuera del intervalo [***a***, ***b***], con ***a*** ≤ ***b***.



# Expresiones de relación y booleanas en C++

---

17. Expresión cuyo valor sea *cierto* si y solo si el valor de ***x*** es par, sabiendo que  $x \geq 0$ .
18. Expresión cuyo valor sea *cierto* si y solo si el valor de ***b*** es impar, sabiendo que  $b \geq 0$ .
19. Expresión cuyo valor sea *cierto* si y solo si el valor de ***a*** es múltiplo de 7, sabiendo que  $a \geq 0$ .



# Expresiones de relación y booleanas en C++

---

20. Expresión cuyo valor sea *cierto* si y solo si el entero ***d*** es par, positivo y divide al entero ***n***.
21. Expresión cuyo valor sea *cierto* si y solo si un triángulo cuyos lados miden ***a***, ***b*** y ***c***, con ***a* > *b*** y ***b* ≥ *c*** es un triángulo rectángulo.

## Más expresiones

---

22. Expresión que, dado un número de habitación  **$h$**  de un hotel, indique la planta en la que se encuentra la habitación. Supondremos un hotel con el sistema de numeración habitual, donde el número de centenas del número de habitación representa la planta.

## Más expresiones

---

23. Una normativa de evaluación anterior a la actualmente vigente permitía otorgar, en cada asignatura, hasta una matrícula por cada 20 estudiantes o fracción.
- Si ***m*** es el número de estudiantes mayor que 0 matriculados en un asignatura, escribe una expresión en C++ que permita obtener el número máximo de matrículas que se podían otorgar con la antigua normativa.

# Soluciones

Necesita la declaración previa de M\_PI en la biblioteca <cmath> o la siguiente declaración  
**const double M\_PI = 3.141592653589793;**

1.  $x * y$
2.  $y - x + 1$
3.  $2.0 * M\_PI * r$
4.  $(x * y * z) / 8$
5.  $\text{sqrt}(a * a + b * b)$   
o, alternativamente,  $\text{sqrt}(\text{pow}(a, 2) + \text{pow}(b, 2))$
6.  $(y - x + 1) * (x + y) / 2$
7.  $t / n$
8.  $t \% n$
9. Porcentaje:  $100.0 * \text{aprobados} / \text{matriculados}$   
Tasa: **double**(aprobados) / **double**(matriculados)

# Soluciones

---

10. `int(round(x))`

11. `x >= y`

12. `a != 0`

13. `a == b`

14. `x > a + b`

15. `a <= x && x <= b`

16. `!(a <= x && x <= b)`

o, alternativamente: `x < a || b < x`

17. `x % 2 == 0`

# Soluciones

---

18.  $b \% 2 \neq 0$

19.  $a \% 7 == 0$

20.  $(d \% 2 == 0) \ \&\& \ (d > 0) \ \&\& \ (n \% d == 0)$

21.  $a * a == b * b + c * c$

o, alternatively,

$$\text{pow}(a, 2) == \text{pow}(b, 2) + \text{pow}(c, 2)$$

22.  $h / 100$

23.  $(m - 1) / 20 + 1$