



Fundamentos de Programación
Grado en Ing. Informática

Guión práctico nº 1

***Tema 2.- Algoritmos. Tipos de
datos, operadores y
expresiones***



1. Diseñe un programa en C++ que muestre por pantalla un mensaje indicando:
Este es mi primer programa en la Universidad de Huelva.
2. Diseñe un programa que lea por teclado un número entero y muestre por pantalla ese número incrementado en 5.
3. ¿Qué mostraría por pantalla la ejecución del siguiente programa?

```
#include<iostream>
using namespace std;
int main () {
int primero, segundo, tercero;
cout << primero << "\n" << segundo << "\n" <<tercero;
return 0;
}
```

4. Diseñe un programa que intercambie el valor de dos variables de tipo entero leídas desde teclado, sacando su valor por pantalla.
5. Diseñe un programa en C++ tal que pida un número entero por teclado que sea una cantidad de bytes y muestre por pantalla cuantos Kbytes son.
6. Diseñe un programa en C++ de tal forma que pedirá la nota de teoría y la nota de prácticas de un examen realizado (se suponen valoradas entre 0 y 10). Y mostrará por pantalla la nota final, sabiendo que la nota de teoría vale el 70% y la nota de prácticas vale el 30%.
7. Dadas las coordenadas por teclado x e y de dos puntos de un plano, muestre por pantalla la distancia que existe entre estos dos puntos.
8. Diseñe un programa que pida dos números enteros por teclado y a continuación calcule su suma, su diferencia, su producto y su cociente y los muestre por pantalla.
9. Determine cuáles de los siguientes identificadores son válidos. Si son inválidos explique por qué (pruébelos en el compilador):

a) registro 1	b) lregistro1	c)archivo_3
d) float	e) \$impuesto	f) _nombre
g) nombre A	h) 123_opcion	i) Belen
j) Resultado_total	k) N_pasaje3	l) Float

10. Diseñe un programa que halle la longitud, la superficie y el volumen de la circunferencia, círculo y esfera. El programa deberá pedir el radio de la circunferencia.

Fórmulas: Longitud = $2 * \text{PI} * R$; Superficie = $\text{PI} * R^2$; Volumen = $4 * \text{PI} * R^3 / 3$

11. Diseñe un programa que pida una cantidad en euros y devuelva su equivalente en dólares americanos y en libras esterlinas. Nota: Utilice constantes para guardar el valor cambio en euros del dólar americano y de las libras esterlinas.

12. Diseñe un programa que lea una temperatura medida en la escala Celsius (grados centígrados) y la convierta a grados Fahrenheit, según la siguiente fórmula:

$$\text{Fahrenheit} = \left(\frac{9}{5}\right) \cdot \text{Celsius} + 32$$

13. Diseñe un programa que resuelva una ecuación de segundo grado $ax^2 + bx + c = 0$. Debe pedir al usuario que introduzca los valores de a, b y c de tipo entero y el programa indicará las dos posibles raíces solución. No considere el caso de raíces imaginarias.

14. Diseñe un programa que lea cinco notas de exámenes con decimales por teclado y calcule la nota media obtenida. Los datos de entrada deben leerse en la misma línea y el resultado deberá mostrarse por pantalla.

15. Diseñe un programa que solicite al usuario un sistema de ecuaciones lineales pidiendo los coeficientes a, b, c, d, e, y f por teclado, y muestre por pantalla los valores de x e y, sabiendo que:

$$ax + by = c$$

$$dx + ey = f$$

$$x = \frac{ce - bf}{ae - bd} \quad y = \frac{af - cd}{ae - bd}$$

16. Diseñe una función que convierta una cantidad positiva de segundos leída desde teclado a su equivalente en horas, minutos y segundos, expresando estos valores por pantalla. Si se lee, por ejemplo, el valor de 3701 seg. debe mostrar por pantalla 1 hora, 1 minuto y 41 segundos.

17. Suponga que observa un presupuesto por arreglar la carretera entre dos pueblos de su provincia de 15000 metros de longitud de acuerdo a la fórmula que se pone a continuación. ¿Darán el mismo resultado las dos asignaciones siguientes en un programa en C++? Diseñe un programa en C++ y muestre por pantalla el resultado.

Precio_total_en_euros = $5000 * (\text{longitud_en_metros} / 5280.0)$;

Precio_total_en_euros = $5000 * (\text{longitud_en_metros} / 5280)$;

18. Diseñe una clase para almacenar las horas normales y extras de un empleado con las operaciones de cargar y calcular nomina. Las horas laborales se facturan a 5 € y las horas extraordinarias a 8 €. Se aplica una retención del 15% independientemente del tipo de facturación. El operador debe entrar como datos desde teclado el número de horas trabajadas normales y extraordinarias. Escriba un programa que haga uso de esta clase.

```
class empleado {
    int horasnormales, horasextras;

public:
    void cargar();
        //Pedirá por teclado las horas normales y las horas extras trabajadas.
    float nomina();
        //Devolverá el sueldo que cobrará después de la retención.
};
```

19. Diseñe una clase rectángulo para almacenar los atributos largo y ancho con las operaciones iniciar (poniendo 2 de ancho y 4 de largo), cargar (poniendo el largo y el ancho que escriba el usuario por teclado), área (que devuelva el valor del área del rectángulo) y cambiar (poner el valor del largo como ancho y el ancho como largo). Escriba un programa que utilice esta clase.

```
class rectangulo {
    int largo;
    int ancho;

public:
    void iniciar ( );
        //poniendo 2 de ancho y 4 de largo
    void cargar ( );
        //poniendo el largo y el ancho que escriba el usuario por teclado
    int area ( );
        //que devuelva el valor del área del rectángulo
    void cambiar( );
        //pondrá el valor del largo como ancho y el ancho como largo
};
```

20. **Movimiento rectilíneo uniformemente acelerado** es aquel en el que un móvil/objeto se desplaza sobre una trayectoria recta estando sometido a una aceleración constante.

La fórmulas que rigen este movimiento son las siguientes:

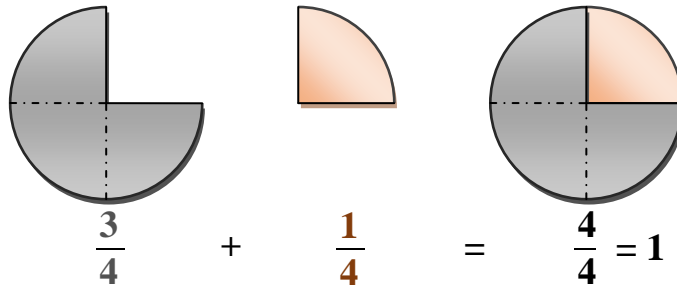
$$V = V_0 + a \times t$$

$$S = S_0 + V_0 \times t + \frac{1}{2} \times a \times t^2$$

Implementar la clase Movimiento que solicite los valores necesarios V_0 , S_0 , a , y t y muestren por pantalla soluciones a problemas como este ejemplo:

Un Formula 1 comienza su vuelta de clasificación con una velocidad inicial de 210m/s, ¿Qué velocidad final tendrá y cuantos metros habrá recorrido si el piloto tiene pisado el pedal de aceleración 10 segundos?

21. **Operaciones con Fracciones.** Una Fracción es la expresión de una cantidad x dividida entre otra cantidad y $\left(\frac{x}{y}\right)$, es decir que representa el cociente no efectuado de números. En la fracción "y" el denominador expresa el número de partes en las que se trocea la unidad, y el numerador "x" indica cuántas partes se toman. Por ejemplo:



Se pide implementar la clase **Fracciones** que solicite los datos de dos fracciones $\frac{a}{b}$ y $\frac{c}{d}$ y muestre por pantalla el resultado de Sumar, Restar, Multiplicar y Dividir ambas fracciones. Dada las limitaciones de la pantalla el ejemplo anterior se mostraría:

$$3/4 + 1/4 = 4/4$$

Las operaciones con fracciones se calculan de la siguiente manera:

$$\text{Sumar: } \frac{a}{b} + \frac{c}{d} = \frac{a \times d + b \times c}{b \times d}$$

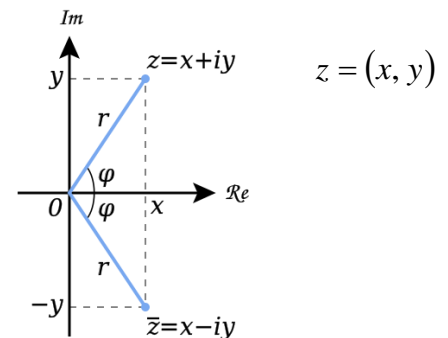
$$\text{Resta: } \frac{a}{b} - \frac{c}{d} = \frac{a \times d - b \times c}{b \times d}$$

$$\text{Multiplicación: } \frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d}$$

$$\text{División: } \frac{a}{b} \div \frac{c}{d} = \frac{a \times d}{b \times c}$$

22. **Operaciones con complejos.** Se define un número complejo como un par ordenado de números reales donde x se define como parte real de z , y el segundo elemento y se define como parte imaginaria de z .

Los números complejos se pueden representar mediante dos ejes cartesianos donde la componente real está en el eje horizontal x y la parte imaginaria y en el eje vertical.



Se pide implementar la clase **Complejo** que solicite los datos de dos complejos $z_1 = (a, b)$ y $z_2 = (c, d)$ muestre por pantalla el resultado de Sumar, Restar, Multiplicar y Dividir ambos

complejos. Dada las limitaciones de la pantalla un número complejo se mostrará de la forma cartesiana $x + iy$.

Las operaciones con complejos se calculan de la siguiente manera:

Sumar: $(a,b) + (c,d) = (a+c, b+d)$

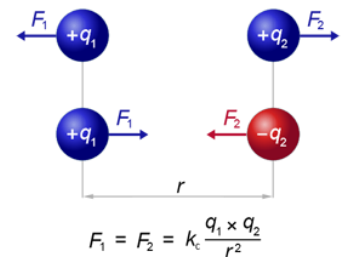
Resta: $(a,b) - (c,d) = (a-c, b-d)$

Multiplicación: $(a,b) \times (c,d) = (ac-bd, ad+bc)$

División: $\frac{(a,b)}{(c,d)} = \frac{(ac+bd, bc-ad)}{(c^2+d^2)} = \left(\frac{ac+bd}{c^2+d^2}, \frac{bc-ad}{c^2+d^2} \right)$

23. El estudio de las fuerzas de atracción o repulsión entre partículas cargadas fue realizado por el físico francés *Coulomb*, cuya ley dice lo siguiente:

Para cargas pequeñas (aproximadamente puntuales) la fuerza de atracción o de repulsión es directamente proporcional al producto de las cargas e inversamente proporcional al cuadrado de la distancia que las separa.



Su expresión matemática es:

$$F = k \frac{q_1 \cdot q_2}{r^2}$$

Donde:

- F es la *fuerza atractiva o repulsiva*.
- q_1 y q_2 son las *cargas eléctricas*.
- r la *distancia* entre ellas
- k es una *constante de proporcionalidad* que depende del sistema de unidades que se emplee y del medio material en el que se encuentran las cargas.

En el vacío y usando el *Sistema Internacional*, en el que F se expresa en *newtons*, q en *culombios* y r en *metros*, k vale aproximadamente:

$$k = 9 \cdot 10^9 \frac{Nm^2}{C^2}$$

Implementar una clase llamada **Cargas** de modo que:

a. Tendrá tres atributos privados: q_1 , q_2 y r . Donde las cargas serán valores en μC (micro C, p.ej. $3\mu C = 3 \times 10^{-6} C$) y la distancia se aportará por el usuario en metros. Hay que tener en cuenta que el usuario indica una carga en μC pero si atendemos a las unidades de la constante k en la fórmula se debe trabajar en C .

b. Tendrá dos métodos públicos:

- a) **void Leer()** que solicitará al usuario los valores de las *cargas* y la *distancia* entre ellas.

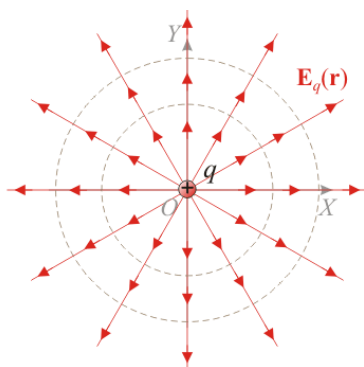
b) **double Fuerza()** que devolverá la *fuerza de atracción/repulsión* entre las cargas.

Codificar un programa que utilice dicha clase.

Ejemplos para comprobar las soluciones obtenidas:

$q_1 = 3\mu\text{C}$	$q_2 = -8\mu\text{C}$	$r = 2\text{m}$	$F = -0.054 \text{ N}$
$q_1 = -0.5\mu\text{C}$	$q_2 = 644\mu\text{C}$	$r = 3.5\text{m}$	$F = -0.237 \text{ N}$
$q_1 = 2.8\mu\text{C}$	$q_2 = 7.5\mu\text{C}$	$r = 0.1374\text{m}$	$F = 10 \text{ N}$
$q_1 = 4\mu\text{C}$	$q_2 = -8\mu\text{C}$	$r = 0.004\text{m}$	$F = -18000 \text{ N}$
$q_1 = 1\mu\text{C}$	$q_2 = 2.5\mu\text{C}$	$r = 0.05\text{m}$	$F = 9 \text{ N}$

24. Si en las proximidades de una carga puntual q , colocamos sucesivamente cargas de prueba q_i , sabemos por la *ley de Coulomb* que dichas cargas se verán sometidas a una fuerza eléctrica. La intensidad del campo eléctrico E , medida en *Newton/Culombio*, creado por dicha carga puntual, se puede expresar del siguiente modo:



$$E = \frac{q}{4\pi\epsilon r^2}$$

Donde:

- q es la *carga puntual* generadora del campo eléctrico.
- ϵ es la *constante dieléctrica* o *permitividad* del medio y cuyo valor en el vacío es:

$$\epsilon_0 = 8.85 \cdot 10^{-12} \frac{\text{C}^2}{\text{Nm}^2}$$

- r es la distancia a la carga puntual.

Implementar una clase llamada **CampoElec** de modo que:

- Tendrá dos atributos privados:** q (valor en μC) y r (valor en m). Al igual que en el ejercicio anterior, hay que tener en cuenta que el usuario indicará una carga en μC pero si atendemos a las unidades de la constante ϵ_0 en la fórmula se debe trabajar en C .
- Tendrá dos métodos públicos:**
 - void Leer()** que solicitará al usuario los valores de la *carga* y la *distancia*.
 - double Intensidad()** que devolverá la *intensidad del campo eléctrico* creado por dicha carga puntual a la distancia r .

Codificar un programa que utilice dicha clase.

Ejemplos para comprobar las soluciones obtenidas:

$q = 0.001\mu\text{C}$	$r = 0.02\text{m}$	$E = 2.25 \times 10^4 \text{ N/C}$
$q = 0.005\mu\text{C}$	$r = 0.3\text{m}$	$E = 499.5 \text{ N/C}$
$q = 0.04\mu\text{C}$	$r = 2\text{m}$	$E = 89.9 \text{ N/C}$

25.- La **energía cinética**, en la *mecánica clásica*, de una masa puntual depende de su *masa* y su *velocidad*. La **energía potencial gravitatoria** se define como la energía que poseen los cuerpos por el hecho de poseer *masa* y estar situados a una determinada *distancia* de la superficie terrestre. La **energía mecánica** es la suma de la energía cinética más la energía potencial.

A continuación se muestra cómo se expresan matemáticamente dichas energías y en qué unidad de miden:

Energía cinética: $E_c = \frac{1}{2}mv^2$ Donde: <ul style="list-style-type: none"> m es la <i>masa</i> expresada en kg. v es la <i>velocidad</i> expresada en m/s 	Energía potencial: $E_p = m \cdot g \cdot h$ Donde: <ul style="list-style-type: none"> m es la <i>masa</i> expresada en kg. g es la <i>gravedad</i> cuyo valor es de 9.8 m/s². h es la <i>altura</i> a la que se encuentra, expresada en m. 	Unidad de medida: La unidad de medida de la energía, en el S.I. es el Julio (J) . Un Julio equivale a: $J = \frac{\text{kg} \cdot \text{m}^2}{\text{s}^2}$
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Implementar una clase llamada **Energía** con las siguientes características:

- Tendrá 3 atributos privados:** m (expresada en **kg**), v (expresada en **km/h**) y h (expresada en **km**). El alumno deberá **tener en cuenta en qué unidad se almacenan** los datos, y en qué unidad deben estar expresados para hacer las operaciones correctamente, de modo que el programa deberá **realizar las conversiones oportunas**.
- Tendrá tres métodos públicos:
 - void Leer()** que solicitará al usuario los valores de *masa*, *velocidad* y *altura* de un determinado cuerpo en las unidades antes indicadas.
 - float ECinetica()** que devolverá la energía cinética.
 - float EPotencial()** que devolverá del energía potencial.

Codificar un programa que utilice dicha clase y muestre por pantalla la **energía cinética**, la **energía potencial** y la **energía mecánica** del cuerpo indicado por el usuario.

Ejemplos para comprobar las soluciones obtenidas:

$m = 75\text{kg}$ $v = 32.4\text{km/h}$ $h = 0.0025\text{km}$	$E_c = 3037.5\text{J}$ $E_p = 1837.5\text{J}$ $E_m = 4875\text{J}$
$m = 2500\text{kg}$ $v = 720\text{km/h}$ $h = 3\text{km}$	$E_c = 5 \times 10^7\text{J}$ $E_p = 7.35 \times 10^7\text{J}$ $E_m = 1.235 \times 10^8\text{J}$
$m = 200\text{kg}$ $v = 10.8\text{km/h}$ $h = 0.03\text{km}$	$E_c = 900\text{J}$ $E_p = 58800\text{J}$ $E_m = 59700\text{J}$