# C++. actividad 1

## Primera parte

* Crear un proyecto de consola con el nombre *Actividades.*
* En *main.cpp*, desarrollar un programa que muestre en la consola el siguiente mensaje:

*Estamos aprendiendo a programar en C++*

*Esta es la primera actividad*

* Incluir un comentario con el enunciado de la actividad.

## Segunda parte

* Dado un código proporcionado por el docente, solucionar los errores y lograr una ejecución correcta del código.

**Objetivos:** Crear un proyecto de consola en C++ y cambiar el mensaje de salida inicial.

# C++. actividad 2

## Primera parte

* Declarar dos variables con valores 72.569 y 160.743.
* Declarar dos variables cuyo contenido será la suma y la resta de las dos variables anteriores respectivamente.
* Mostrar en pantalla el resultado de la suma y la resta entre estas dos variables.

## Segunda parte

* Calcular el IVA de un determinado importe siguiendo esta especificación:
  + Crear una variable para introducir un importe.
  + Crear una constante *IVA* con el valor 0.21.
  + Crear una variable con el resultado del multiplicar el importe por el IVA.
  + Mostrar un mensaje en consola con la información.

**Objetivos:** Utilizar variables en C++

# C++. actividad 3

* Utilizar la función pow() de la librería cmath para hallar el resultado de 23
* Desarrollar una función que muestre un mensaje en pantalla del resultado requerido.
* Llamar a la función para mostrar el resultado.

**Objetivos:** Utilizar funciones y cabeceras en C++ para resolver cálculos

# C++. actividad 4

* Utilizar las secuencias de escape para mostrar este texto en pantalla:

\ESTE\

"TEXTO"

CONTIENE

'SECUENCIAS'

DE

ESCAPE

* Tener en cuenta los saltos de línea y los tabuladores.
* Tener en cuenta que los caracteres \, “ y ‘ hay que escaparlos.

**Objetivos:** Utilizar funciones y cabeceras en C++ para resolver cálculos

# C++. actividad 5

## Primera parte

* Solicitar dos números por consola y mostrar en pantalla la suma, la resta, el producto, la división y el promedio de los números*.*

## segunda parte

* Declarar una variable con el valor 8 y llegar a 100 utilizando los operadores:
  + ++
  + –
  + \*=
  + /=
* Declarar una variable numérica. A ella le sumaremos 1 y la mostraremos a la vez en una sola sentencia. En la consola se mostrará el valor ya incrementado en 1.

**Objetivos:** Utilizar operadores en C++ para resolver cálculos

# C++. actividad 6

* Desarrollar un programa en el que se realice lo siguiente:
  + Declarar un string que contenga el texto *“Estamos aprendiando”*
  + Cambiar las letras *a* que están de forma incorrecta.
  + Añadir *“Phyton”* al final.
  + Sustituir *“Phyton”* por *“C++”.*
  + Eliminar todo lo que está después de *“aprendiendo”.*

**Objetivos:** Manipular strings en C++

# C++. actividad 7

## Primera parte

* Desarrollar un programa que calcule el sueldo de un trabajador dependiendo de las horas trabajadas.
* Estas horas trabajadas serán introducidas por el usuario.
* El trabajador podrá trabajar un máximo de 40 horas semanales, cada una de ellas a 20€/hora.
* Todas las horas extra serán pagadas a 30€/hora.
* Una vez conseguido esto, controlar la entrada de las horas. Si la entrada de las horas es correcta se harán los cálculos. Si no lo es, mostrar un mensaje de error en la consola.

## segunda parte

* Desarrollar un programa que calcule el importe de una multa según una velocidad introducida por consola a la que es interceptado un vehículo según las siguientes condiciones:
  + Si la velocidad es inferior a 0, mostrar un mensaje de error.
  + Si es menor o igual que 50 la multa será 0.
  + Si está entre 51 y 60 la multa será de 50€.
  + Si está entre 61 y 80 la multa será de 100€.
  + Si está entre 81 y 100 la multa será de 10 € por cada km/h que pase de 50.Cualquier otra velocidad será penalizada con la retirada del carnet.
  + Mostrar un mensaje en consola informando de la multa.

## Tercera parte

* Introducir una cifra de ventas por consola.
* Mediante un ternario, calcular las comisiones según estas condiciones:
  + Si superan 1000, será el 20% del importe.
  + Si no superan 1000, será el 10% del importe.
* Mostrar en consola el resultado.

## cuarta parte

* Desarrollar un programa que simule una calculadora. El usuario introducirá dos números por consola y el signo de la operación que se desea realizar.
* Mediante un bloque switch haremos una deriva de casos según el signo para mostrar por consola un mensaje con la suma, la resta, el producto y la división de los números.
* Si el signo no es +, -, \* ni / mostraremos un mensaje con el texto *“Operación no permitida”.*

**Objetivos:** Utilizar sentencias de toma de decisiones en C++

# C++. actividad 8

## Primera parte

* Desarrollar un programa que calcule el promedio de diferentes números introducidos por el usuario.
* El usuario podrá introducir todos los números que quiera y para finalizar el bucle el usuario introducirá el número cero.
* Resolver la actividad utilizando el bucle *while.*

## Segunda parte

* Desarrollar otra versión de la primera parte utilizando el bucle *do while*.

## Tercera parte

* Desarrollar un programa que muestre por consola la tabla de multiplicar de un número utilizando el bucle *for*.
* Se nos pedirá un número en pantalla entre 1 y 10 mediante consola.
* En el caso de que el número sea correcto aparecerá en pantalla desarrollada su tabla de multiplicar. Si no lo es, mostraremos un mensaje por consola.
* Por cada iteración sacar el resultado con la especificación clásica. Por ejemplo, para la tabla del 5 la primera iteración devolvería 5 x 1 = 5, la siguiente 5 x 2 = 10…. Hasta 5 x 10 = 50

**Objetivos:** Utilizar bucles en C++

# C++. actividad 9

## Primera parte

* Crear un array e introducir en él los días de la semana.
* Solicitar un número por teclado del 1 al 7.
* Mostrar en un mensaje el día de la semana elegido. (Por ejemplo, si se ha elegido 2 debe aparecer en el mensaje martes).

## segunda parte

* Crear un array vacío con 5 nombres.
* Pedir los nombres por consola*.*
* Pedir por pantalla el número de nombre queremos vaciar.
* Mostrar el mensaje “Eliminada la persona *nombre*“.

**Objetivos:** Utilizar arrays en C++

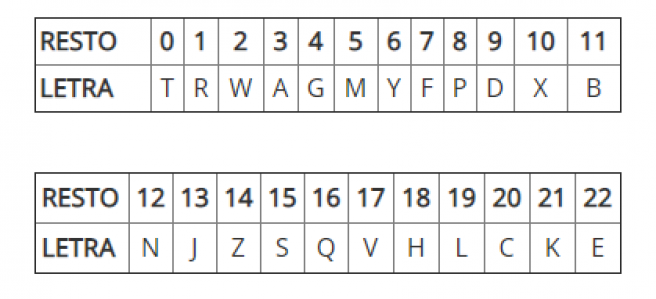
# C++. actividad 10

## Primera parte

* Desarrollar un programa que mediante una función calcule la suma de 5 números enteros.
* La función aceptará como mínimo dos argumentos, siendo los otros tres opcionales con el valor por defecto 0.
* Invocar a la función y comprobar si el resultado es el esperado.

## Segunda parte

* Desarrollar una función que devuelva la letra de un DNI basándonos en el número.
* Para hacer esto, calcularemos el reto de dividir el número entre 23.
* Una vez hallado el resto, la equivalencia entre ese número y la letra correspondiente será esta:



* Invocar a la función y comprobar si el resultado es el esperado.

**Objetivos:** Utilizar funciones en C++

# C++. actividad 11

## Primera parte

* Crear una clase con el nombre *CuentaCorriente*. Tendrá las propiedades públicas: *titular, saldo e interés*.
* Crear un método *getInteresMensual* que devuelva el resultado de la fórmula: saldo\*interés/100/12
* Crear un método *getInteresAnual* que devuelva el resultado de la fórmula: saldo\*interés/100
* Crear un objeto de la clase y probar el funcionamiento de los métodos.

## segunda parte

* Para esta parte utilizaremos la clase *CuentaCorriente*.
* Crear dos constructores. Uno sin argumentos y otro con los tres argumentos que pasen a las propiedades. Utilizar el puntero *this* para pasar los valores a las propiedades. En este segundo constructor.
* Crear métodos *get* y *set* para acceder a las propiedades.
* En el método *set* para el saldo, comprobar que el saldo no sea negativo. Utilizar este método *set* en el segundo constructor.

**Objetivos:** Utilizar clases en C++

# C++. actividad 12

Suponemos que una aplicación de gestión de un almacén necesita una clase para representar los artículos disponibles en el inventario.

* Definir una clase *Articulo* que tenga tres propiedades: *nombre (string), existencias(int)* y *precio(double).*
* Declarar un constructor con los valores por defecto.
* Declarar un segundo constructor con las tres propiedades utilizando el puntero *this.*
* Declarar los métodos *get* y *set* necesarios para acceder a las propiedades. El precio no podrá ser negativo, siendo cero si no se cumple la especificación.
* Crear un método *stockValorado* que devuelva el producto de las *existencias* por el *precio.*
* Crear un método *pedirAProveedor* que devuelva un valor booleano. Si las existencias son inferiores a 100 devolverá *true* y si no lo son devolverá *false.*
* Crear una segunda versión del método *pedirAProveedor* que acepte por parámetro un umbral de existencias. Si las existencias son inferiores a ese valor devolverá *true* y si no lo son devolverá *false.*
* Crear una propiedad estática *promedioPrecios* que vaya calculando el promedio de los precios según los objetos que se instancien.
* Instanciar algunos objetos de la clase para comprobar el funcionamiento de la misma.

**Objetivos:** Utilizar clases en C++

# C++. actividad 13

## Primera parte

* Crear un array e introducir en él los días de la semana.
* Solicitar un número por teclado del 1 al 7.
* Mostrar en un mensaje el día de la semana elegido. (Por ejemplo, si se ha elegido 2 debe aparecer en el mensaje martes).

## segunda parte

* Suponemos que vivimos en un bloque de pisos y queremos almacenar los vecinos en un array:
  + Crear un array vacío con 5 vecinos. Por cada uno crearemos un objeto con las propiedades nombre y edad.
  + Agregar cada vecino al array*.*
  + Pedir por pantalla el piso del vecino que queremos eliminar.
  + Mostar el nombre y la edad ese vecino.

**Objetivos:** Utilizar arrays en C++