C++

Una característica importante es que C++ es tipificado y no Python

C++ se debe tener con la basura, no tiene recolector de basura

Administrar la memoria new(solicitar) y delete(borrar)

Punteros => Referencias

C++ Java

Librería en C++

En C++, se conoce como librerías (o bibliotecas) a cierto tipo de archivos que se pueden importar o incluir. Estos archivos contienen las especificaciones de diferentes funcionalidades ya construidas y utilizables, como por ejemplo leer del teclado o mostrar algo por pantalla entre otras.

Al poder incluir estas librerías con definiciones de diferentes funcionalidades se puede realizar la reutilización de código.



Algunas librerías de C++

Librerías estándar de este lenguaje

* **fstream:** Flujos hacia/desde ficheros. Permite la manipulación de archivos desde el programar, tanto leer como escribir en ellos.
* **iosfwd:** Contiene declaraciones adelantadas de todas las plantillas de flujos y sus typedefs estándar. Por ejemplo ostream.
* **iostream:** Parte del a *STL* que contiene los algoritmos estándar, es quizá la más usada e importante (aunque no indispensable).
* **math:** Contiene los prototipos de las funciones y otras definiciones para el uso y manipulación de funciones matemáticas.
* **memory:** Utilidades relativas a la gestión de memoria, incluyendo asignadores y punteros inteligentes (*auto\_ptr*).

*"auto\_ptr"* es una clase que conforma la librería memory y permite un fácil manejo de punteros y su destrucción automaticamente.

* **Biblioteca new:** Manejo de memoria dinámica
* **numeric:** Parte de la librería numérica de la *STL* relativa a operaciones numéricas.
* **ostream:** Algoritmos estándar para los flujos de salida.
* **Librería stdio:** Contiene los prototipos de las funciones, macros, y tipos para manipular datos de entrada y salida.
* **Librería stdlib:** Contiene los prototipos de las funciones, macros, y tipos para utilidades de uso general.
* **string:** Parte de la *STL* relativa a contenedores tipo string; una generalización de las cadenas alfanuméricas para albergar cadenas de objetos. Muy útil para el fácil uso de las cadenas de caracteres, pues elimina muchas d elas dificultades que generan los char
* **typeinfo:** Mecanismo de identificación de tipos en tiempo de ejecución
* **iterator:** Proporciona un conjunto de clases para iterar elementos.
* **regex:** Proporciona fácil acceso al uso de expresiones regulares para la comparación de patrones.
* **thread:** Útil para trabajar programación multihilos y crear múltiples hilos en nuestra aplicación.
* **time:** Útil para obtener marcas de tiempo durante ejecución. Se usa con frecuencia para conocer el tiempo exacto durante un programa.

Main en C++

Todos los programas de C++ deben tener una función main. La función main es donde el código fuente empieza a ejecutarse.



Usingname space

El uso de *using namespace std;* esta línea ayuda a declarar un espacio de nombre que evita tener que usarlo cada que se accede a una función específica de una librería.

Con el namespace declarado se puede llamar por ejemplo el comando cout >>, que pertenece a la librería iostream, sin embargo sin este namespace sería std::cout >>.



Tipos Primitivos de C++ y sus rangos

Pertene al lenguaje

TDA Tipo Abstracto de Datos No pertenece

* **bool:** El tipo de dato bool, tiene un tamaño de 8 bits y un rango entre 0 y 1, falso o verdadero, este tipo de dato, es comúnmente usado en condicionales o variables que solo pueden tomar el valor de falso o verdadero.
* **int:** El tipo de dato int, tiene un tamaño de 32 bits y un rango entre -2.147.483.648 y 2.147.483.647. Este tipo de dato, es usado para números enteros (sin cifras decimales). M
  + **Modificadores:**
  + **short int:** Tiene un tamaño de 16 bits y un rango entre -32.768 y 32.767.
  + **unsigned short int:** Tiene un tamaño de 16 bits y un rango entre 0 y 65535.
  + **unsigned int:** Tiene un tamaño de 32 bits y un rango entre 0 y 4.294.967.295.
  + **long long int:** Tiene un tamaño de 64 bits y un rango entre -9.223.372.775.808 y 9.223.375.775.807.
  + **unsigned long long int:** Tiene un tamaño de 64 bits y un rango entre 0 y 2exp64.
* **float:** El tipo de dato float tiene un tamaño de 32 bits, es usado comúnmente en números con 6 o menos cifras decimales. Tiene un rango entre 1,17549\*(e^-38) hasta 3,40282\*(e^+38).
* **double:** El tipo de dato double tiene un tamaño de 64 bits, es usado para números de menos de 15 cifras decimales. Tiene un rango entre 2,22507\*(e^-308) hasta 1,79769\*(e^308).
  + **Modificadores:**
  + **long double:** Tiene un tamaño de 96 bits y una precisión de 18 cifras decimales. Tiene un rango entre 3,3621\*(e^-4932) hasta 1,18973\*(e^4932).
* **char:** Las variables del tipo char, son las variables problema del lenguaje C y C++, puesto que tienen una gran cantidad de restricciones y complicaciones. Las variables de tipo char, en C y C++ se consideradas vectores y con un tamaño máximo, entre corchetes "[ ]" lo cual restringe un poco al no saber qué tamaño podría llegar a tener una cadena de caracteres, y aunque hay formas de evadir esto, es bastante complicado.

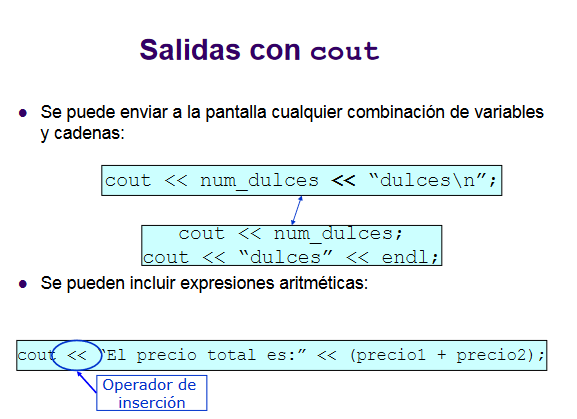
Es recomendable usar las variables de tipo string para las cadenas de caracteres, incluidas en la librería string.h que son bastante fáciles de usar. La sintaxis para declarar un char es la siguiente: char nombre\_char[tamañoMax]; o simplemente char nombre = 'valor' notar que va entre comillas simples.

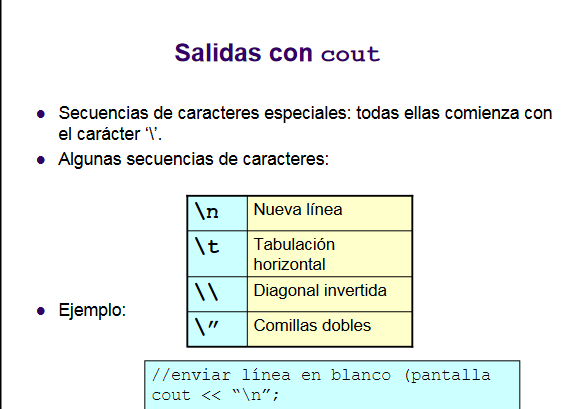
Cout y cin del iostrem

Existen otras librerías que se pueden utilizar.

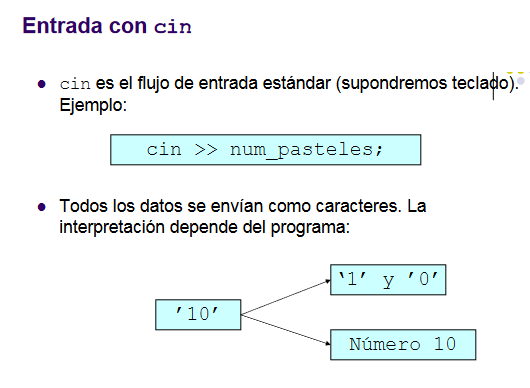
**Cout** permite imprimir mensaje al monitor.

Ejemplos

  
  
**Formatos de las salidas**

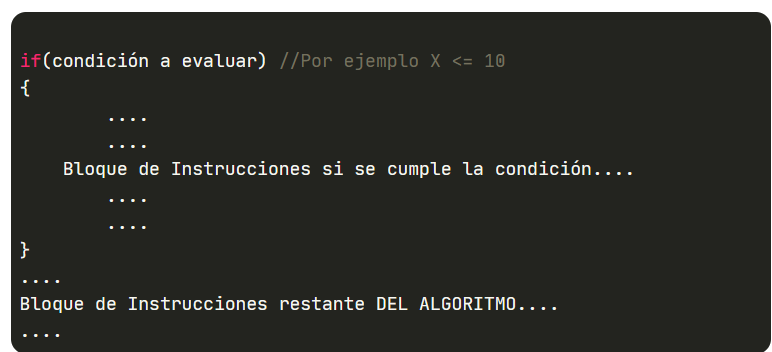


**Cin :** captura de información de teclado

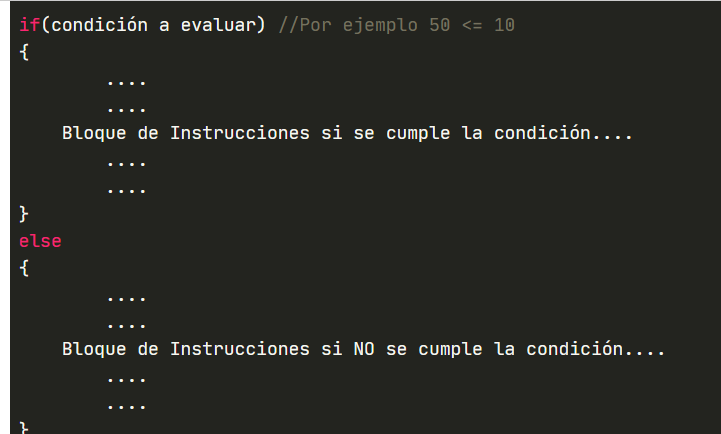


En este link se encuentra mas información importante del cout y el cin. Esta lo básico <http://informatica.uv.es/iiguia/FP/EntradaSalida.pdf>

Condicional if



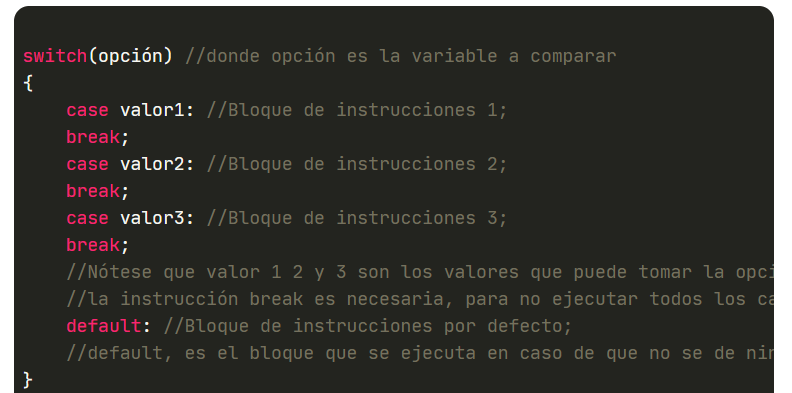
Condicional if-else



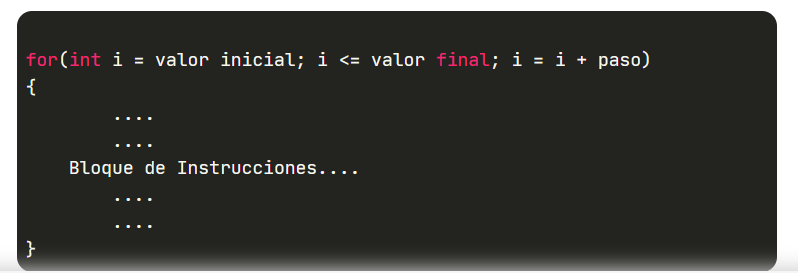
Switch

Los condicionales Switch, son una estructura de control condicional, que permite definir múltiples casos que puede llegar a cumplir una variable cualquiera, y qué acción tomar en cualquiera de estas situaciones, incluso es posible determinar qué acción llevar a cabo en caso de no cumplir ninguna de las condiciones dadas.

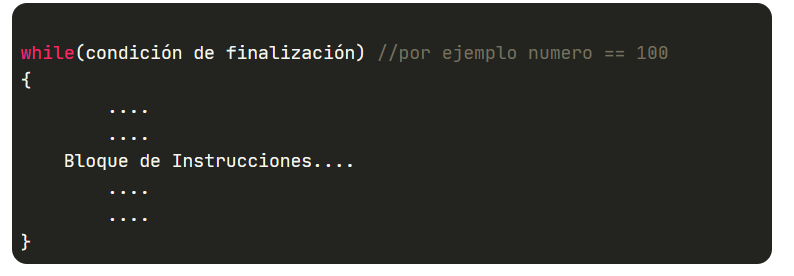
Ejemplo de un Switch



Sintaxis Ciclo For



Sintaxis Ciclo while



Tipos de errores:

* Sintaxis son los más sencillos, no se sigue la sintaxis del lenguaje

Python x:5

* Lógica los más complicados
* Linkeo que es enlazar el código fuente con una librería

1. Administrar de la memoria

Nodo

Class Nodo

Int valor;

Nodo \*siguiente;

Estructura de datos que puede utilizarse para la implementación de nuevas [estructuras](https://definicion.de/estructura) (tales como las colas, las pilas y listas,etc) y está formada por una serie de nodos que almacenan, además de la información deseada, un enlace, un puntero o una referencia al nodo que lo precede, al posterior, o bien uno a cada uno.

Nodos

Int String

Int String

primero

NULL

Direccion

siguiente

Int valor

primero

primero->valor primero->siguiente

Clase

Plantilla o molde . formada por atributos y métodos.

Una clase es en general un modelo, receta o plantilla que define el estado y comportamiento de cierto tipo de objetos. Una clase puede pensarse como una colección de variables (atributos o propiedades) y funciones (métodos) que permiten representar un conjunto de datos y especificar las operaciones o procedimientos que permiten manipular tales datos. Se puede inclusive entender una clase como un tipo de dato personalizado, similar a las estructuras (structs C).

Ejemplo :

Estudiante

Atributos

identificación, nombre,tel,dir

Los atributos, también llamados datos o variables miembro son porciones de información que un objeto posee o conoce de sí mismo.

Métodos

matricular,desmatricular,cambiar teléfono

Los métodos de una clase son funciones que sirven para manipular las variables de la clase, de ahí viene la primera característica relevante de la programación orientada a objetos que es el encapsulamiento, ya que en lo posible se va a tratar de que solo pueda accederse a una variable de clase a través de un método de la clase.

Constructor

permite inicializar los atributos. Python init

C++ mismo nombre de la clase. Nombre es el mismo de la clase, pueden existir varios y se diferencia de acuerdo con:

Cantidad de parámetros, tipos de parámetros o bien orden de los parámetros

Un constructor es un método que se ejecuta de manera automática al instanciar un objeto de una clase. El constructor tiene como finalidad la inicialización de las variables de la clase y posiblemente ejecutar algunos de los métodos de la clase. Una clase puede tener tantos constructores (sobrecargas) como sea necesario. La característica más sobresaliente de los constructores es que su nombre es el mismo que el de la clase, es decir, son métodos de la clase que se nombran igual que la clase, y que además no tienen valor de retorno. Las distintas sobrecargas del constructor de una clase van a depender de las distintas combinaciones de parámetros de entrada de la función.

Deben tener diferente cantidad de parámetros, diferente orden o tipo si la misma cantidad de parámetros.

Clases de autoreferencia

Class nodo

private: // atributos

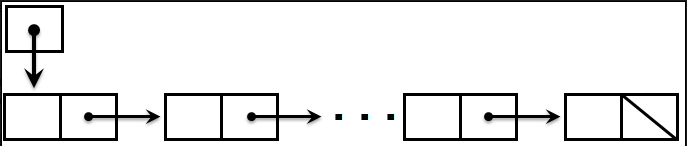
int valor; // tipo entero. Ej clase estudiante nombre,dir,tel,siguente

nodo \*siguiente; // tipo de la clase, clase autoreferencia. Direccion [2,3,4]

 En la Programación orientada a objetos, la autorreferencia es la capacidad que tienen los objetos de una clase de almacenar explícitamente una referencia a objetos de su misma clase, es decir, a objetos de su mismo tipo.  
  
  [Ejemplo Nodo](https://drive.google.com/open?id=0B2VcsvtisQAaOEJkR0p3S2JUaGc), el cual muestra una clase autorreferida (Nodo) con dos atributos:

* dato: el atributo que representa el elemento a almacenar en el nodo.
* siguiente: el cual es una referencia a un objeto cuya clase es la misma que lo define (Nodo), y por lo tanto, es una autorreferencia.

primero



Conjunto de Nodos

NULL

Lista de simple : no es secuencial. Lista Enlazada es una secuencia de elementos donde, a cada uno de estos se les llama nodos.

Una lista enlazada es un tipo de dato autorreferenciado porque contienen un puntero o enlace a otro dato del mismo tipo. Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante (suponiendo que dicho punto está previamente identificado o localizado), pero no permiten un acceso aleatorio.



primero

Puntero : dirección de memoria

Un puntero es una variable que almacena la dirección de memoria de un objeto. Los punteros se usan ampliamente en C y C++ para tres propósitos principales:

* para asignar nuevos objetos en el montón,
* para pasar funciones a otras funciones
* para recorrer en iteración los elementos de matrices u otras estructuras de datos.

2.C++ tipificado. Tipos primitivos int, double, char

3.No indentación, los bloques de instrucciones se { }

4.instrucciones ;

6. No existe elif