Córdoba, 11 de abril del 2016

Universidad de Córdoba
Escuela Politécnica Superior
Tercero del grado en Ingeniería Informática
Especialidad en Computación
Procesadores de Lenguajes
Curso académico 2015-2016

C++

Descripción del lenguaje de programación

Francisco José Rodríguez Ramírez

Víctor Monserrat Villatoro

Índice

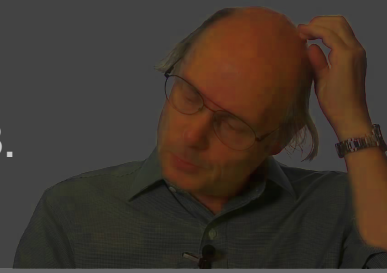
Índice

Índice

Introducción

Volver al índice

- Diseñado por Bjarne **Stroustrup** a mediados de los años **1980**.
- Creado con la intención de permitir la **manipulación de objetos** en C.
- Posteriormente se añadieron facilidades de **programación genérica**.
- Posibilidad de **redefinir los operadores**, y de poder **crear nuevos tipos** que se comporten como tipos fundamentales.
- El nombre **C++** fue propuesto por Rick **Mascitti** en el año **1983**.



Descripción

Volver al índice

- Lenguaje compilado.
- Fuertemente tipado.
- Es un lenguaje orientado a objetos.
- Portable.
- Muy extendido (disponibilidad de librerías).

Descripción

Volver al índice

- Permite reutilizar las funciones de C.
- Permite la sobrecarga de operadores y funciones.
- Soporta espacios de nombres.
- Soporta plantillas.
- Tiene muchísima comunidad.

Elementos del lenguaje

Volver al índice

- Palabras reservadas.
- Símbolos.
- Literales.
- Identificadores.
- Procedimientos y funciones.

Elementos del lenguaje

Volver al índice

Componentes léxicos o tokens

- Palabras reservadas.

  Tokens de caracteres con utilidad especial.

  No se pueden utilizar como nombre de identificador o función.

  main, float, int, for, break, return...

Elementos del lenguaje

Volver al índice

Componentes léxicos o tokens

- Símbolos.

  Operadores básicos como: `+` (suma), `=` (asignación)

  Constructor de sentencias `;`

  Agrupar varias sentencias con `{ }` o expresiones con `( )`

  `/*` Comentarios `*/`

Elementos del lenguaje

Volver al índice

Componentes léxicos o tokens

- Literales.

  Son la especificación de un valor de tipo concreto

  ‘7’, ‘a’, 9, 10.56

Elementos del lenguaje

Volver al índice

Componentes léxicos o tokens

- Identificadores.

  Son case sensitive.

  Empiezan por letra o subrayado.

  No pueden contener espacios, acentos u otros caracteres especiales.

  Máxima longitud: 31

  estado, nElementos, num-elementos

Elementos del lenguaje

Componentes léxicos o tokens

- Procedimientos y funciones.

  Operadores cuya definición está en las bibliotecas que hay que incluir.

  Procedimientos: ejecutan una tarea sin devolver un valor.

  ```
  void process();
  ```

  Funciones: devuelven un valor.

  ```
  int function();
  ```

Tipos de **datos**

Volver al índice

- Entero.
- Real.
- Carácter.
- Lógico.

Tipo de dato **entero**

Volver al índice

Tipo	Memoria	Rango
int	2 bytes	[-32768, 32767]
unsigned int	2 bytes	[0, 65535]
short int	2 bytes	[-128, 127]
long	4 bytes	$[-2^{31}, 2^{31} -1]$
unsigned long	4 bytes	$[0, 2^{32} - 1]$
char	1 byte	[0, 255]

Esta tabla es **orientativa**. El rango depende de cada S.O. y compilador

Tipo de dato **entero**

Volver al índice

Operadores

Binarios	**+**	Suma
	-	Resta
	*****	Multiplicación
	/	División
	%	Módulo
Unarios	**++**	Incremento
	--	Decremento

Tipo de dato **real**

Volver al índice

Tipo	Memoria	Rango
float	4 bytes	7 dígitos
double	8 bytes	15 dígitos
long double	8 bytes	19 dígitos

Esta tabla es **orientativa**. El rango depende de cada S.O. y compilador

Tipo de dato **real**

Operadores

+	Suma
-	Resta
*****	Multiplicación
/	División

Tipo de dato **carácter**

Volver al índice

- En C no existe el tipo carácter como tal. Dos representaciones:
  - Internamente se representan como un número (código ASCII).
  - Como un literal de carácter. Se representan entre comillas simples: ‘!’, ‘A’, ‘a’, ‘5’.

- Se pueden realizar operaciones aritméticas (las aplicables a los enteros).

Tipo de dato **lógico**

Volver al índice

- Es un tipo de dato muy común en los lenguajes de programación que se utiliza para representar valores verdadero y falso que suelen estar asociados a una condición (por ejemplo ser un número par).

```
bool variable1 = true;
```

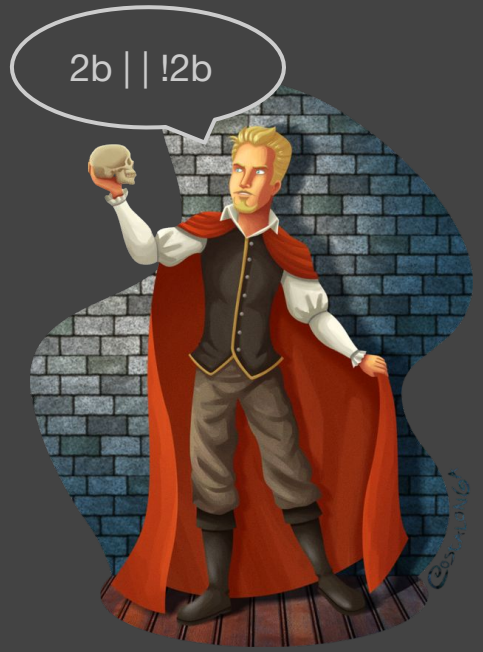
```
bool variable2 = false;
```

Tipo de dato **lógico**

Volver al índice

Operadores lógicos

&&	Y
||	O
!	No



Tipo de dato **lógico**

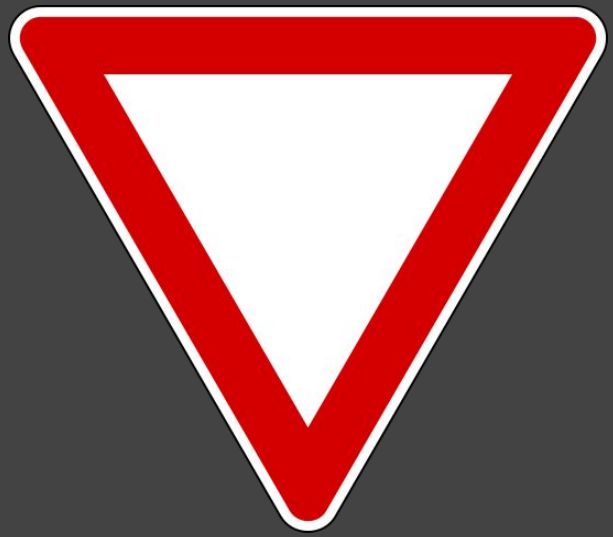
Operadores relacionales

==	Igual
!=	Distinto
<, <=	Menor, menor o igual que
>, >=	Mayor, mayor o igual que

Precedencia de **operadores**

Volver al índice

( )
!
* / %
+ -
< <= > >=
== !=
&&
||



Clases

Volver al índice

- Son construcciones que se utilizan como modelos para crear objetos de algún tipo.
- Ayudan a la programación mediante la abstracción.
- Cada **instancia** de una clase se denomina objeto.
- Tienen una parte pública y otra privada. En algunas ocasiones, puede incluso tener una parte protegida.

Ejemplo de **clases**

Volver al índice

```
class Object {

    public:

        Object() { _isInstantiated = true; }


    private:

        bool _isInstantiated;

};
```

Sobrecarga de **funciones**

Volver al índice

- Permite utilizar distintas funciones con el mismo nombre pero distinto prototipo.
- Esto ayuda a tener más claro y ordenado el código, facilitando al programador el nombramiento y uso de funciones.

Ejemplo de **sobrecarga de funciones**

Volver al índice

```
int mayor(const int &a, const int b) {
    if(a > b) return a;
    else return b;
}

int mayor(const int &a, const int &b, const int &c) {
    return mayor(mayor(a, b), c);
}
```

Sobrecarga de **operadores**

Volver al índice

- Permite decirle al compilador cómo queremos que actúe nuestro programa ante distintos operadores y para distintas clases.
- Se establecen en las clases.
- Nos da facilidades y claridad en el código.

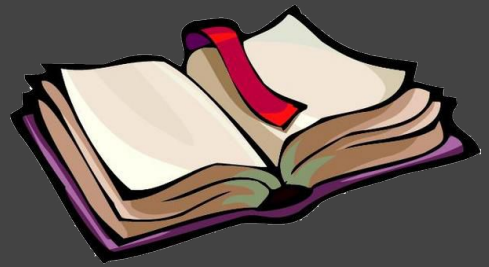
Ejemplo de **sobrecarga de operadores**

Volver al índice

```
int& operator +(const Object &x, const Object &y) {

    return x.size() + y.size();

}



ostream& operator <<(ostream &o, const Object &x) {

    o << x.name() ;

    return o;

}
```

Bibliografía

Volver al índice



- **Wikipedia.**
  - Colaboradores de Wikipedia. C++ [en línea]. Wikipedia, La enciclopedia libre, 2016 [fecha de consulta: 11 de abril del 2016]. Disponible en **https://es.wikipedia.org/**
- **Cplusplus.**
  - Colaboradores de Cplusplus. C++ [en línea]. Cplusplus, 2016 [fecha de consulta: 11 de abril del 2016]. Disponible en **http://www.cplusplus.com/**