

Alfredo Vellido : www.lsi.upc.edu/~avellido

Fonaments d'Informàtica

Semana 2. Primeros programas



Escola d'Enginyeria de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Conceptos básicos

Elementos básicos de C++

- En este curso aprenderemos las bases de la programación en C++: (mega-lenguaje, del que sólo veremos un pequeño subconjunto)
 - ¿Cuáles son los componentes básicos de un programa en C++?
 - Elementos para **guardar la información (en memoria)** y operar con ella:
 - **Variables, tipos de datos y expresiones**
 - **Instrucciones que se van ejecutando secuencialmente y que manipulan las variables**
- (**Recordad:** C++ es un lenguaje de programación imperativo)

Conceptos básicos

Tipos de datos básicos de C++

- **Tipos de datos** = cjto. de valores + operaciones
- **Literales**: Manera de representar los valores

int: (emulan a los **enteros**)

- Ejemplos de **literales**: -345, -7, 0, 56 ...
- **Valores**: -345, -7, 0, 56
- Operaciones: - (unario/cambio signo), +, - (binario/resta), *, /, %

float, double: (emulan a los **reales** –diferente precisión: núm. de decimales-)

- Ejemplos de **literales**: 3.45, -3.89999, 3.1e-5
- **Valores**: 3,45 -3,89999 0,000031
- Operaciones: - (unario/cambio signo), +, - (binario/resta), *, /

Conceptos básicos

Tipos de datos básicos de C++ (2a)

■ **char:** (emula los caracteres alfanuméricos)

- Ejemplos de literales: `'A'`, `'a'`, `'1'`, `':'`, `' '`, `endl`, ...
- Valores: letras mayúsculas y minúsculas, dígitos, signos puntuación, `'\n'`, salto de línea
- Operaciones: `+` `-` `/` (Aritmética de caracteres: `char('a'+'A')`)




■ **bool:** (emula los booleanos)

- Literales: **true**, **false**
- Valores: cierto, falso
- Operaciones: **and** (`&&`), **or** (`||`), **not** (`!`)

Conceptos básicos

Tipos de datos básicos de C++ (2b)

■ **bool:** (emula los **booleanos**)

 and	<p>The "and" operation takes two inputs and produces one output. If both inputs are true, the output is true; in all other cases, the output is false. It can be interpreted as follows: <i>"I will return true if input 1 and input 2 are true."</i></p>
 or	<p>The "or" operation takes two inputs and produces one output. If either of the inputs are true, the output is true; otherwise (i.e., if neither input is true), the output is false. It can be interpreted as follows: <i>"I will return true if either input 1 or input 2 is true."</i></p>
 not	<p>The "not" operation takes one input and produces one output. If the input is true, the output is false. If the input is false, the output is true. In other words, the "not" operation takes the input and returns its opposite.</p>

Conceptos básicos

Tipos de datos básicos de C++ (2c)

- **bool**: (emula los booleanos)
- **Tabla de la verdad** de los operadores booleanos

a	b	!a	a && b	a b
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false

Conceptos básicos

Tipos de datos básicos de C++ (3)

- Excepcionalmente, también usaremos el tipo `string` (que no es básico)
 - Literales: `"hola"`, `"hola, que tal?"`, `" "`, ...
 - **NOTA:** Necesita `#include <string>`
- **Operaciones relacionales** (comunes a todos los tipos):
 - `<` , `>` , `<=` , `>=` (`false < true`; `'a' < 'b'` ... código ASCII),
`=` , `!=`

Interludio: ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Conceptos básicos

Variables y expresiones

- **Variable = tipo de datos + nombre**
- (Wikipedia dixit) *Una variable es un elemento de un algoritmo. Cada variable esta asociada a un valor que puede variar durante la ejecución.*
- Los **valores** de una variable deben ser **de un tipo determinado**.
- **Variable**: *nombre* al que asociamos un *valor* del *tipo* correspondiente; el valor puede ir cambiando durante la ejecución del algoritmo. **Variable como “contenedor”**.

Conceptos básicos

Variables y expresiones

- **Variable = nombre + tipo de datos**
- El nombre, que nos lo inventamos nosotros, será una cadena alfanumérica (letras minúsculas y números). **NOTA:** Puede contener el carácter '_'.

Conceptos básicos

Expresión: construcción formada por variables, literales, operaciones, paréntesis (y quizá alguna cosa más: e.g. llamadas a funciones) sintácticamente correcta.

Ejemplos de expresiones

`5==6`

`5*8`

`6/4+ ---> incorrecta`

`true or false`

Conceptos básicos

Ejemplos de expresiones

- Si **x**, **y** son variables de tipo entero ...

x-y

x*x

xy ----> **incorrecta!**

Conceptos básicos

Cómo se evalúa una expresión:

Precedencias:

- (unario/cambio signo), ! (not) ---> los primeros en evaluarse

*, /, %

+, -

<, >, <=, >=,

==, !=

&& (and), || (or) ---> los últimos en evaluarse

Ejemplo:

$3 + 2 * 2 - 15$ ---> se evalúa a -8

$32 \% 4 + 12 - 4 * 3$ ---> se evalúa a 0

$3 + 7 / -5 < 8 * 1 - 2$ ---> se evalúa a true

Conceptos básicos

Ejemplos de expresiones

- A **igualdad de prioridad**, de izquierda a derecha:

5 - 2 + 4 ---> se evalua a ?

5 * 2 / 4 ---> se evalua a ?

3 + 7 / 3 * 2 - 15 ---> se evalua a ?

5 < 6 == not true ---> se evalua a ?

Conceptos básicos

Ejemplos de expresiones

- Se pueden poner **paréntesis** para romper el orden:

Ejemplo:

$(3 + 2) * 2 - 15$

---> se evalúa a **-5**

$32 \% (4 + 12) - 4 * 3$

---> se evalúa a **-12**

$3 + 7 / -5 < 8 * (1 - 2)$

---> se evalúa a **false**

Conceptos básicos

Ejemplos de expresiones (3b)

- Cuidado con los tipos en C++ !!!

`int x = 5.0 * 2 / 4` ----> `x=2` (aviso!: convirtiendo a `int`
de `double`)

`double x = 5.0 * 2 / 4` ----> `x=2.5`

`int x = 5 * 2 / 4` ----> `x=2`

Conceptos básicos

De vuelta a los elementos básicos de C++

- Esquema base de un programa en C++

```
<Inclusión de librerías>
<Definición de constantes>
<Definición de tipos>
int main (void)
{
    <Declaración de variables>
    <instrucción1>
    <instrucción2>
    ...
    <instrucciónN>
}
```

Conceptos básicos

De vuelta los elementos básicos de C++

- Un ejemplo sencillo: leer de teclado, escribir en pantalla

```
#include <iostream> // Inclusión de la biblioteca de entrada/salida
using namespace std; /* Directiva: usamos el espacio de nombres de la
                        biblioteca estándar*/

int main(void)
{int entero;
 char character;
 cout << "Entra un entero: ";
 cin >> entero;
 cout << "Entra un caracter: ";
 cin >> character;
 cout << "El entero que has entrado es el: " << entero << endl;
 cout << "El caracter que has entrado es: " << character << endl;
 system ("pause"); return 0;
}
```

Conceptos básicos

Declaración de variables

- **Función:** Dar nombre y tipo a una variable. Opcionalmente, también valor.
- **Sintaxis:** `tipo nombrevariable [= expresion (valor inicial)];`
- **Efecto:** A partir de ese punto, se puede usar una variable de nombre "nombrevariable" y tipo "tipo"

Conceptos básicos

Declaración de variables

- **Sintaxis:** `tipo nombrevariable [= expresion (valor inicial)];`

- Ejemplos:

```
int n;  
int m = 0;  
int m = 3*n;  
bool existe;  
double prod;  
char letra;  
string palabra;
```

- **Declaración de una variable:** la primera vez que aparece, se indica su tipo.
- **IMPORTANTE!!!:** En el nombre **NO** valen espacios ni guiones '-' (sí '_'). **Tampoco** los nombres pueden **comenzar por una cifra**.

Conceptos básicos

Asignación

- **Función:** asignar a una variable el resultado de evaluar una expresión.
- **Sintaxis:** `nombrevariable = expresion;`
(`<nombrevariable>` es una variable de tipo T,
 `<expresion>` es una expresión de tipo T)
- **Efecto:** asigna a la variable de nombre "*nombrevariable*" el resultado de evaluar "*expresion*";

Conceptos básicos

Asignación

- **Sintaxis:** `nombrevARIABLE = expresion;`

- **Ejemplos:**

```
int i,j;  
...  
j = i;  
j = 3*i + 20%(i-1);  
  
int m=0,n=1;  
bool a, b=false;  
...  
a = (m < n) or b;  
a = m != n;
```