

60. Actividad

De forma individual investiga:

1. Palabras reservadas de C++

Palabras reservadas Lenguaje C

Utiliza ciertos nombres para designar funciones, dichos nombres se conocen como palabras reservadas. Una variable no puede llamarse igual que una palabra reservada. C cuenta con 48 palabras reservadas, estas son:

1	auto	Modificador que indica que una vble local se crea al inicio de la ejecución de la función y se destruye al final
2	break	Provoca que se termine la ejecución de una iteración o para salir de la sentencia <i>switch</i>
3	case	Sentencia que hace parte de cada uno de los casos del <i>switch</i>
4	char	Tipo de dato carácter (normalmente 1Byte)
5	const	Define variables cuyo valor debe permanecer constante durante toda la ejecución del programa
6	continue	Provoca que se comience una nueva iteración, evaluándose la expresión de control
7	default	Es el caso por defecto que se ejecuta si dentro del <i>switch</i> no concuerda ninguno de los casos definidos
8	do	Variación del <i>while</i> donde primero se ejecuta y después se procede a evaluar la expresión de control
9	double	Tipo de dato flotante (real) (normalmente 8Bytes)
10	else	Es la segunda parte de un condicional <i>if</i> , se ejecuta cuando no se cumpla la primer condición
11	enum	Permite declarar valores de datos que se ajustan a series ordenadas en las cuales un elemento sigue, o precede, a otro.
12	extern	Define que existe una variable global que está definida en otro archivo fuente
13	float	Tipo de dato flotante (real) (normalmente 4Bytes)
14	for	Sentencia de control iterativa, que permite inicializar los controles de un ciclo mediante la estructura: <i>for</i> (Inicialización; Condición; Actualización)
15	goto	Instrucción de control de salto que permite realizar saltos en el flujo de control de un programa, es decir, permite transferir el control del programa, alterando bruscamente el flujo de control del mismo.
16	if	Sentencia de control básica en la que se evalúa una expresión condicional
17	int	Tipo de dato entero con signo (normalmente 2 o 4Bytes)
18	long	Tipo de dato entero largo con signo (normalmente 4Bytes)
19	register	Modificador es aplicable únicamente a variables locales e indica al compilador que esta variable debe ser almacenada permanentemente en un registro del procesador del ordenador
20	return	Indica a una función el valor que debe devolver
21	short	Tipo de dato entero corto con signo (normalmente 2Bytes)
22	signed	Modificador que indica que el tipo dato con signo
23	sizeof	Operador que nos permite obtener el tamaño de un tipo o de una variable.
24	static	Modificador que indica que una variable local no se destruye al finalizar la función donde fue declarada.

2. Reglas para la construcción de Identificadores en C++

1. Debe empezar por una letra o carácter de subrayado.

`natural` `_var2` `alumno_e11` `x1`

A pesar de estar permitido, es recomendable no usar el subrayado al inicio, pues es utilizado por los desarrolladores de bibliotecas y compiladores y, eventualmente, podemos generar una colisión.

2. Las mayúsculas son caracteres distintos que las minúsculas.

`Interes` e `Interes` son variables distintas.

3. No pueden utilizarse las palabras reservadas del lenguaje, como `double`, `false`, etc.

4. No deben llevar comas, puntos, acentos, espacios en blanco, la letra `ñ`, etc.

`tamaño` `var 1` `interés` `día.24` no son válidos.

3. Operadores de C++

I. De asignación

II. Aritméticos

III. Lógicos

¿Qué representan las siguientes expresiones regulares?

$L=[a-zA-Z_]+$

Selecciona caracteres de a-z y de A-Z, además selecciona el carácter “_”. Coincidencias de 1 o más.

$D=[0-9]^+$

Selecciona uno o más números del 0-9.

$E=[, \backslash t, \backslash r, \backslash n]^+$

Selecciona los espacios, las comas, tab, retorno de carro y salto de línea

Notas: Las llaves “{ }” además de utilizarse con cuantificadores se utilizan para hacer referencia a una expresión regular previamente definida. Los paréntesis “()” permiten agrupar y las comillas permite agregar un símbolo que no está considerado

$(L)(\{L\}\{D\})^*$

Encuentra una letra (a-z A-Z), después puede encontrar 0 o más dígitos del 0 al 9, o una letra a-z A-Z.

$(\"-\"{D}^+)|\{D\}^+$

Encuentra uno (o más) números negativos o positivos.

“+”

Encuentra un símbolo de +

while

Busca la palabra “while”

Utiliza las expresiones regulares para llenar la siguiente tabla:

Token	Lexema	Patrón	Reservada (SI/NO)
Letras	Hola, mundo	[a-zA-Z_]+	No
Número	0,1,2,33	[0-9]+	No
Espacios	“ ”	[,\t,\r,\n]+	No
Variable	A4	(L)({L} {D})*	No
Número	-6,90	("-"{D}+) {D}+	No
Operador suma	+	“+”	Sí
Condicional (while)	While	while	Sí