

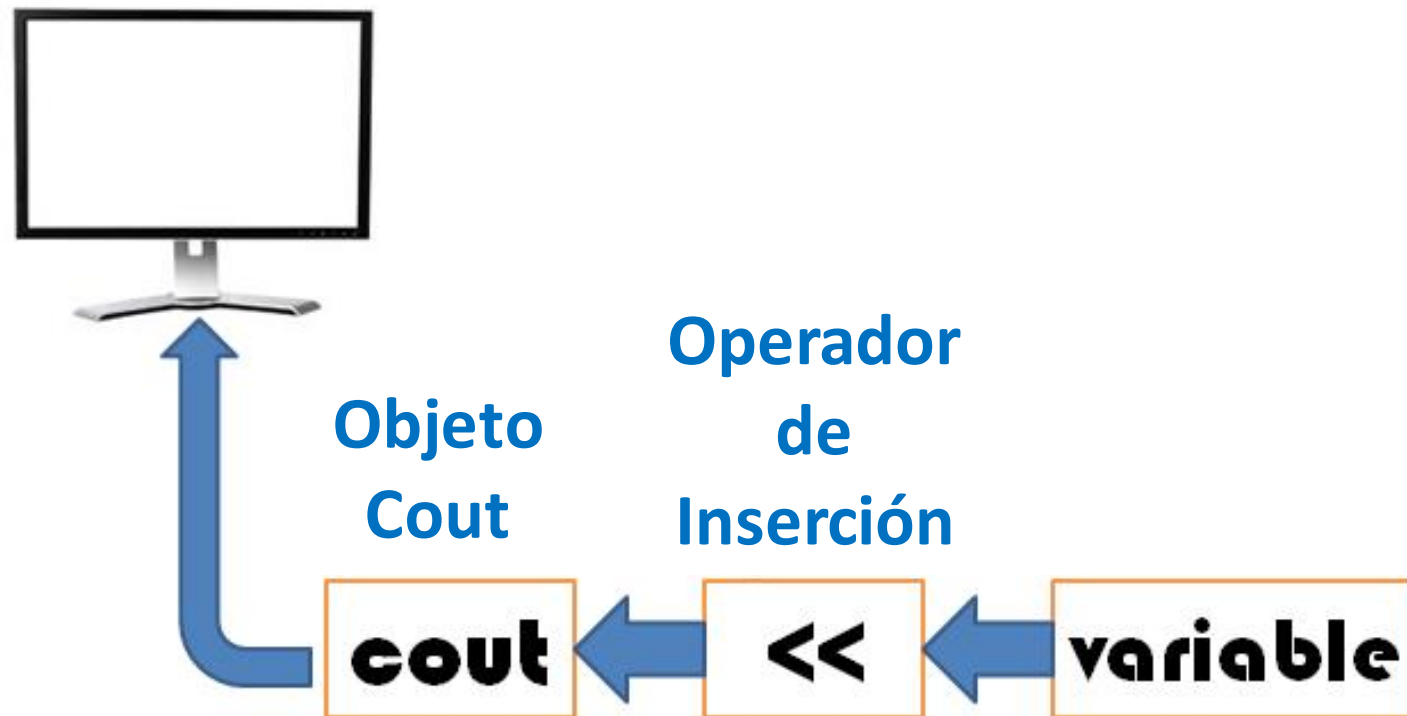
PROGRAMACIÓN ESTRUCTURADA EN C++

Conceptos y Operaciones Básicas

JONATHAN JEAN PAUL ARANCIBIA SANABRIA

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hola Mundo!" << endl;
7     return 0;
8 }
```

cout : console output



cout : console output

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hola 1";
7     cout << "Hola 2" << endl;
8
9     return 0;
10 }
```

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hola 1" << "Hola 2" << endl;
7     cout << "Hola 1\n" << "Hola 2" << endl;
8     cout << "Hola 1\b" << "Hola 2" << endl;
9     cout << "Hola 1\r" << "Hola 2" << endl;
10    cout << "Hola 1\t" << "Hola 2" << endl;
11
12    return 0;
13 }
```

CONCEPTOS BÁSICOS

COMENTARIOS EN EL CÓDIGO

```
1 #include<iostream>
2 using namespace std;
3 /*
4     Comentario
5     de múltiples
6     líneas
7 */
8 int main() //Función principal
9 {
10     cout << /*Comentario incómodo*/"Hola 1";
11     cout << "Hola 2" << endl;
12     //En la consola se imprimirá el
13     //siguiente mensaje
14     //Hola1Hola2
15
16     return 0;
17     //El main() retorna un 0 al sistema operativo
18 }
```

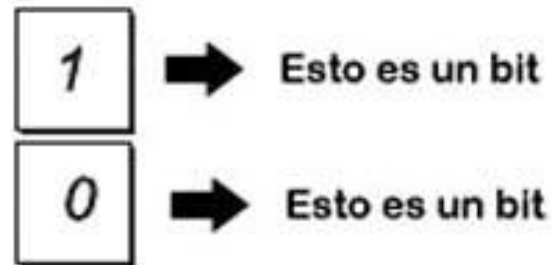
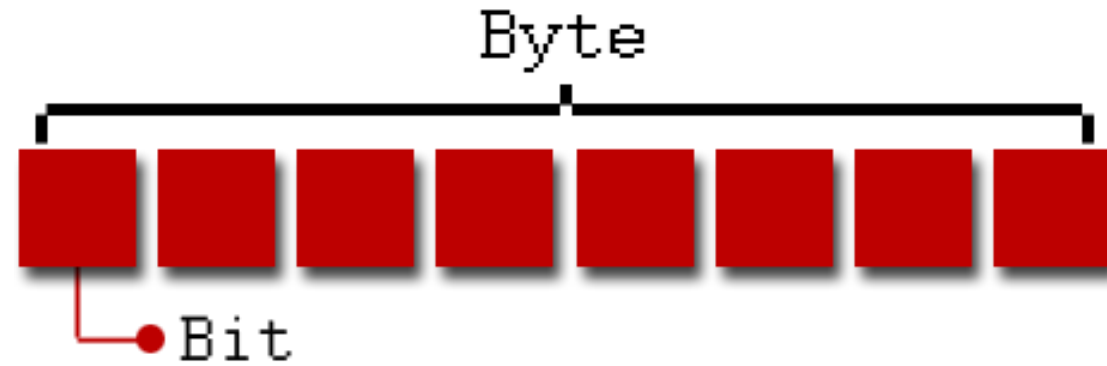
int : enteros
char : caracteres
float : reales
double : reales
bool : booleanos

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int edad;
7     char letra;
8     bool respuesta;
9     float talla;
10    double distancia;
11
12    return 0;
13 }
```

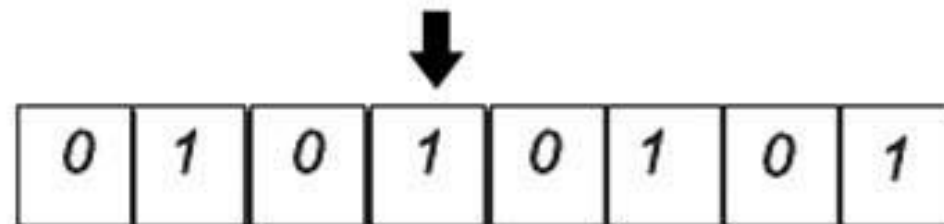
ASIGNACIONES DE VARIABLES

El símbolo = es el Operador de Asignación
Asigna de derecha a izquierda

```
1#include<iostream>
2using namespace std;
3
4int main()
5{
6    int edad;
7    char letra;
8    bool respuesta;
9    float talla;
10   double distancia;
11   //Las asignaciones se hacen de derecha a izquierda
12   edad = 17;        //Válido (Asignación de un entero)
13   20 = edad;        //Inválido
14
15   letra = 'A';      //Asignación de un caracter
16   respuesta = true;  //Asignación de un booleano
17   talla = 1.8;      //Asignación de un flotante
18   distancia = 108.2; //Asignación de un double
19   return 0;
20}
```

Esto es un byte



CHAR

Una variable de tipo CHAR ocupa **1 byte** de espacio en la memoria
Cuenta con 8 bits, por lo tanto con $2^8 = 256$ posibles valores.

INT

Una variable de tipo INT ocupa **4 bytes** de espacio en la memoria
Cuenta con 32 bits, por lo tanto con $2^{32} = 4,294,967,296$ posibles valores.
1 bit para el signo + 31 bits para el número, por lo tanto el rango de valores es:
-2,147,483,648 hasta el 2,147,483,647

MODIFICADORES:

short : reduce el espacio de un int a 2 bytes (16 bits) | -32,768 a 32,767 | 65,536 valores

long : aumenta el espacio de un int a 4 bytes (32 bits) | para arquitecturas que trabajan el int con 2 bytes

signed: permite que la variable contenga valores tanto negativos como positivos

unsigned: permite que la variable contenga solo valores positivos | en un int de 4 bytes tendría el siguiente rango: 0 a 4,294,967,295

FLOAT

Una variable de tipo FLOAT ocupa **4 bytes** de espacio en la memoria

El rango de valores es:

1.17e-38 a 3.40e38

DOUBLE

Una variable de tipo DOUBLE ocupa **8 bytes** de espacio en la memoria

El rango de valores es:

2.22e-308 a 1.80e308

BOOL

Una variable de tipo BOOL ocupa **1 byte** de espacio en la memoria

Los valores posibles para este tipo de dato son: **true** o **false**

Trabajar con valores como 10900000 puede resultar tedioso al momento de escribir código.

$$10900000 \equiv 1.09 \times 10^7$$

$1.09 \times 10^7 \equiv 1.09e7$ es una representación mas sencilla

```
1#include<iostream>
2using namespace std;
3
4int main()
5{
6    int valor;
7
8    valor = 1.09e7;
9
10   cout << valor << endl;
11
12   return 0;
13}
```

cin : console input



cin : console input

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int edad;
7
8     cout << "Ingrese su edad: ";
9     cin >> edad;
10
11     cout << "Usted tiene " << edad << " años." << endl;
12     return 0;
13 }
```

CONCEPTOS BÁSICOS

PREGUNTAS DE REPASO

¿Qué son las variables?

- 1 Son espacios en la memoria que contienen información. Su contenido no puede variar a lo largo de la ejecución del programa. Tiene un tipo determinado correspondiente al tipo de información que puede guardar.
- 2 Son espacios en la memoria que contienen información. Su contenido puede variar a lo largo de la ejecución del programa. Puede almacenar una información de cualquier tipo y no es necesario indicarle un tipo.
- 3 Son espacios en la memoria que contienen información. Su contenido puede variar a lo largo de la ejecución del programa. Tiene un tipo determinado correspondiente al tipo de información que puede guardar.

CONCEPTOS BÁSICOS

PREGUNTAS DE REPASO

Indique lo correcto sobre los tipo char (caracter).

- 1 Puede almacenar valores enteros positivos o negativos.
- 2 El peso en la memoria de un char es de 1 bit.
- 3 Un caracter almacena una letra, símbolo o dígito.

CONCEPTOS BÁSICOS

PREGUNTAS DE REPASO

Indique lo **correcto** con respecto a los tipo float y double.

- 1 Los tipos floats almacenan solo enteros y los double almacenan reales.
- 2 Los tipo floats y double almacenan reales y tienen el mismo peso en la memoria.
- 3 Un tipo float y double almacenen reales. Los tipo double ocupan el doble de espacio en la memoria.

CONCEPTOS BÁSICOS

PREGUNTAS DE REPASO

Indique el valor que representa lo siguiente : $8.5E-2$

1 0.085

2 0.85

3 850

OPERACIONES BÁSICAS

OPERADORES ARITMÉTICOS

```
1#include<iostream>
2using namespace std;
3
4int main()
5{
6    //Suma
7    cout << 7 + 5 << endl;    //Imprimirá: 12
8    //Resta
9    cout << 6 - 3 << endl;    //Imprimirá: 3
10   //Multiplicación
11   cout << 2 * 2 << endl;    //Imprimirá: 4
12   //División
13   cout << 8 / 4 << endl;    //Imprimirá: 2
14   //Residuo
15   cout << 7 % 5 << endl;    //Imprimirá: 2
16   return 0;
17 }
```

OPERACIONES BÁSICAS

DIVISIÓN ENTERA Y DIVISIÓN REAL

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << 7 << endl;    //¿Existe alguna diferencia
7     cout << 7.0 << endl; //entre ambas expresiones?
8
9     //¿Que expresiones imprimiran lo mismo en la consola?
10    cout << 7 / 3 << endl;
11    cout << 7.0 / 3.0 << endl;
12    cout << 7.0 / 3 << endl;
13    cout << 7 / 3.0 << endl;
14    return 0;
15 }
```

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     //En la siguiente expresión hay 4 operadores
6     //¿En que orden se efectúan?
7     cout << 14-8/2*3+1 << endl;    //Imprimirá 3
8     //1: Paréntesis
9     //2: Multiplicación, División y Residuo
10    //3: Suma y Resta
11    //mismo peso: se resuelve de izquierda a derecha
12    //Probemos con las siguientes expresiones
13    cout << (14-8)/2*3+1 << endl;    //Imprimirá 10
14    cout << 14-(8/2*3)+1 << endl;    //Imprimirá 3
15    cout << 14-8/2*(3+1) << endl;    //Imprimirá -2
16    cout << (14-8)/2*(3+1) << endl;    //Imprimirá 12
17    return 0;
18 }
```

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     bool ans1 = 5 > 3;    // mayor que
6     bool ans2 = 7 < 1;    // menor que
7     bool ans3 = 4 == 4;   // igual que
8     bool ans4 = 5 <= 6;   // menor o igual que
9     bool ans5 = 8 >= 8;   // mayor o igual que
10    return 0;
11 }
```

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     bool var1 = (5>3) && true; // && operador lógico AND
6     bool var2 = false || true; // || operador lógico OR
7     bool var3 = 3 != 6;        // != operador "diferente de"
8
9     cout << var1 << endl;
10    cout << var2 << endl;
11    cout << var3 << endl;
12
13    return 0;
14 }
```

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 int main()
5 {
6     const double PI = 3.1415;
7     double raiz_cuadrada = sqrt(15);
8     double potencia = pow(10,4);
9     double raiz_decima = pow(12,0.1);
10    double seno = sin(PI/2.0);
11    double coseno = cos(PI);
12    double tangente = tan(PI/6);
13    return 0;
14 }
```


Marque lo correcto con respecto a los operadores aritméticos.

- ☐ 1 La suma, resta y multiplicación tienen la misma precedencia.
- ☐ 2 El operador módulo halla el residuo entre dos valores reales.
- ☐ 3 El operador multiplicación y división tienen la misma precedencia.

¿Cuál sería el valor de la variable **resultado** al final de las siguientes operaciones?

```
float resultado;
```

```
int a;
```

```
a= 15%2;
```

```
resultado = a/3;
```

☒ 1

0

☐ 2

0.3333

☐ 3

1

¿Para qué sirve la función `sqrt(valor)`?

1

Nos devuelve el cuadrado de un valor ingresado entre parentesis.

2

Nos devuelve la raíz cuadrada de un valor ingresado entre paréntesis.

3

Nos devuelve la tangente de un valor ingresado entre paréntesis.

¿Qué hace la siguiente línea de código?

```
const int valor = 100;
```

1

Se declara una variable de tipo entero con valor 100.

2

Se declara una constante de tipo entero con valor 100.

Indique qué significan los siguiente operadores en el orden correspondiente.

<, >, <=, >= , ==

1

Mayor, menor, menor igual, mayor igual, igualdad.

2

Menor, mayor, menor igual, mayor igual, igualdad.

3

Menor, mayor, menor igual, mayor igual, asignación.

PROGRAMACIÓN ESTRUCTURADA EN C++

GRACIAS 😊

JONATHAN JEAN PAUL ARANCIBIA SANABRIA