

Introducción a C++

Instructor: Jhon Oblitas

5 de Julio, 2024



§1. Introducción

Mucha gente piensa que C++ es la mejor opción para un programador competitivo, y C++ está casi siempre disponible en los sistemas de concurso. Los beneficios de usar C++ son que es un lenguaje muy eficiente y su biblioteca estándar contiene una gran colección de estructuras de datos y algoritmos.

§1.1. Plantilla de código en C++

```
#include <bits/stdc++.h> // Con esto llamamos a todas las librerías de C++.

using namespace std;
int main(){
    // a partir de aqui podemos escribir código
    return 0;
}
```

§1.2. Entrada y salida de datos

En la mayoría de los concursos, se usan flujos estándar para leer input y escribir output. En C++, los flujos estándar son **cin** para input y **cout** para output. Además, se pueden usar las funciones de C `scanf` y `printf`.

■ Ejemplo de Input (cin):

```
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Ingrese un número: ";
    cin >> num;
    cout << "El número ingresado es: " << num << endl;
    return 0;
}
```

■ Ejemplo de Output (cout):

```
#include <iostream>
using namespace std;

int main() {
    int a = 5, b = 7;
    cout << "La suma de " << a << " y " << b << " es " << (a + b) << endl;
    return 0;
}
```

§2. Variables

Las variables en C++ se usan para almacenar datos que pueden ser utilizados y manipulados en el programa. Aquí hay un ejemplo de cómo declarar e inicializar variables:

```
#include <iostream>
using namespace std;

int main() {
    int entero = 10;
    double decimal = 3.14;
    char caracter = 'A';
    string cadena = "Hola, Mundo!";
    cout << entero << ", " << decimal << ", " << caracter << ", " << cadena << endl;
    return 0;
}
```

§3. Operaciones Matemáticas

En C++, podemos realizar varias operaciones matemáticas usando operadores aritméticos, a continuación veremos algunos ejemplos:

```
#include <iostream>
using namespace std;

int main() {
    int a = 10, b = 3;
    cout << "Suma: " << (a + b) << endl;
    cout << "Resta: " << (a - b) << endl;
    cout << "Multiplicacion: " << (a * b) << endl;
    cout << "Division: " << (a / b) << endl;
    cout << "Modulo: " << (a % b) << endl;
    return 0;
}
```

§4. Declaración de Condicionales

Las estructuras condicionales nos permiten tomar decisiones en el código. Las más comunes son `if`, `else if`, y `else`. Aquí hay un ejemplo:

```
#include <iostream>
using namespace std;

int main() {
    int numero = 10;

    if (numero > 0) {
        cout << "El numero es positivo." << endl;
    } else if (numero < 0) {
        cout << "El numero es negativo." << endl;
    } else {
        cout << "El numero es cero." << endl;
    }

    return 0;
}
```

§5. Ciclos

Los ciclos en C++ se usan para repetir un bloque de código múltiples veces. Los más comunes son `for`, `while`. Aquí hay ejemplos de cada uno:

§5.1. Ciclo for

```
#include <iostream>
using namespace std;

int main() {
    for (int i = 0; i < 5; i++) {
        cout << "i = " << i << endl;
    }
    return 0;
}
```

§5.2. Ciclo while

```
#include <iostream>
using namespace std;

int main() {
    int i = 0;
    while (i < 5) {
        cout << "i = " << i << endl;
        i++;
    }
    return 0;
}
```

§6. Arreglos o Vectores

Los arreglos y vectores son estructuras de datos que almacenan múltiples elementos. Los arreglos tienen un tamaño fijo, mientras que los vectores pueden cambiar de tamaño dinámicamente.

§6.1. Arreglos

```
#include <iostream>
using namespace std;

int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    for (int i = 0; i < 5; i++) {
        cout << "arr[" << i << "] = " << arr[i] << endl;
    }
    return 0;
}
```

§6.2. Vectores

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec = {1, 2, 3, 4, 5};
    for (int i = 0; i < vec.size(); i++) {
        cout << "vec[" << i << "] = " << vec[i] << endl;
    }
    return 0;
}
```