

UNIVERSIDAD PRIVADA
DOMINGO SAVIO

PROGRAMACION III

REPOSITORIO

Name: Rodrigo Muñoz Pizarro

Group: Dynamic Quartet

Subject: Programming III

Teacher: Ing. Jimmy Nataniel Requena Ll.

06/2025
Santa cruz- Bolivia

```
onlinegdb.com

main.cpp
1 #include <iostream> // Para std::cout, std::endl
2
3 int main() {
4     int Rodrigo = 20; // Una Rodrigo entera normal
5     int *puntero; // DECLARACIÓN de un puntero a un entero
6
7     // ASIGNACIÓN: 'puntero' ahora guarda la dirección de 'Rodrigo'
8     puntero = &Rodrigo;
9
10    std::cout << "--- Información de 'Rodrigo' ---" << std::endl;
11    std::cout << "Valor de 'Rodrigo': " << Rodrigo << std::endl;
12    std::cout << "Dirección de 'Rodrigo' (&Rodrigo): " << &Rodrigo << std::endl;
13
14    std::cout << "\n--- Información de 'puntero' ---" << std::endl;
15    std::cout << "Contenido de 'puntero' (la dirección que guarda): " << puntero << std::endl;
16    std::cout << "Dirección donde está guardado el propio 'puntero' (&puntero): " << &puntero << std::endl;
17
18    std::cout << "\n--- Accediendo al valor A TRAVÉS del puntero ---" << std::endl;
19    std::cout << "Valor al que apunta 'puntero' (*puntero): " << *puntero << std::endl; // DEREFERENCIA
20
21    // Modificando 'Rodrigo' A TRAVÉS del puntero
22    std::cout << "\n--- Modificando a través del puntero ---" << std::endl;
23    *puntero = 30; // Ve a la dirección que guarda 'puntero' y cambia el valor allí a 30
24    std::cout << "Nuevo valor de 'Rodrigo' (después de *puntero = 30): " << Rodrigo << std::endl;
25    std::cout << "Nuevo valor apuntado por 'puntero' (*puntero): " << *puntero << std::endl;
26    std::cout << "RODRIGO MUÑOZ PIZARRO " << std::endl;
27
28    return 0;
29
30 }
```

input

```
--- Accediendo al valor A TRAVÉS del puntero ---
Valor al que apunta 'puntero' (*puntero): 20

--- Modificando a través del puntero ---
Nuevo valor de 'Rodrigo' (después de *puntero = 30): 30
Nuevo valor apuntado por 'puntero' (*puntero): 30
RODRIGO MUÑOZ PIZARRO

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com

main.cpp
1 #include <iostream>
2 struct Nodo {
3     int dato;
4     Nodo* siguiente;
5
6     Nodo(int valor_dato) : dato(valor_dato), siguiente(nullptr) {} // Constructor conciso
7 };
8
9 int main() {
10    // 1. Crear el primer nodo (cabeza de nuestra mini-lista)
11    Nodo* cabeza = new Nodo(10); // Usamos 'new' porque queremos memoria dinámica
12    std::cout << "Creado primer nodo (cabeza) con dato: " << cabeza->dato << std::endl;
13
14    // 2. Crear un segundo nodo
15    Nodo* segundoNodo = new Nodo(20);
16    std::cout << "Creado segundo nodo con dato: " << segundoNodo->dato << std::endl;
17
18    // 3. ¡ENLAZARLOS!
19    // EL puntero 'siguiente' del primer nodo (cabeza) ahora apunta al segundoNodo
20    cabeza->siguiente = segundoNodo;
21    std::cout << "Enlazando cabeza->siguiente con segundoNodo." << std::endl;
22
23    // 4. Crear un tercer nodo
```

input

```
Creado primer nodo (cabeza) con dato: 10
Creado segundo nodo con dato: 20
Enlazando cabeza->siguiente con segundoNodo.
Creado tercer nodo con dato: 30
Enlazando segundoNodo->siguiente con tercerNodo.

Recorriendo la mini-lista:
Dato en cabeza: 10
Dato en el segundo nodo (via cabeza->siguiente): 20
Dato en el tercer nodo (via cabeza->siguiente->siguiente): 30

Liberando memoria...
Tercer nodo liberado.
Segundo nodo liberado.
Primer nodo (cabeza) liberado.

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com
Run Debug Stop Share Save Beautify
Language C++

main.cpp
19
20 // 2. Asignar memoria para un arreglo de doubles
21 std::cout << "\n--- Arreglo Dinamico ---" << std::endl;
22 double *p_arreglo_doubles = nullptr;
23 int tamano_arreglo = 5;
24 p_arreglo_doubles = new double[tamano_arreglo]; // Solicita memoria para 5 doubles
25
26 if (p_arreglo_doubles != nullptr) {
27     for (int i = 0; i < tamano_arreglo; ++i) {
28         p_arreglo_doubles[i] = i * 1; // Asigna valores al arreglo
29     }
30
31     std::cout << "Arreglo dinamico creado y llenado:" << std::endl;
32     for (int i = 0; i < tamano_arreglo; ++i) {
33         std::cout << "p_arreglo_doubles[" << i << "] = " << p_arreglo_doubles[i]
34             << " en dir: " << (p_arreglo_doubles + i) << std::endl;
35     }
36
37     delete[] p_arreglo_doubles; // ¡IMPORTANTE! Usar delete[] para arreglos
38     p_arreglo_doubles = nullptr; // Buena práctica
39     std::cout << "Memoria del arreglo dinamico liberada." << std::endl;
40     std::cout << "RODRIGO MUÑOZ PIZARRO" << std::endl;
41 } else {
42     std::cout << "ERROR: No se pudo asignar memoria para p_arreglo_doubles." << std::endl;
43 }
44
45 // Intentar usar un puntero nulo (solo para demostrar, usualmente causa error o comportamiento indefinido)
46 if (p_entero == nullptr) {
47     std::cout << "\np_entero es ahora nullptr." << std::endl;
48     *p_entero = 789; // ¡Esto causaría un error de segmentación! (Descomentar con precaución)
49 }
50
51 return 0;
52 }
53 }

Input
p_arreglo_doubles[1] = 1 en dir: 0x634398bb96e8
p_arreglo_doubles[2] = 2 en dir: 0x634398bb96f0
p_arreglo_doubles[3] = 3 en dir: 0x634398bb96f8
p_arreglo_doubles[4] = 4 en dir: 0x634398bb9700
Memoria del arreglo dinamico liberada.
RODRIGO MUÑOZ PIZARRO

p_entero es ahora nullptr.

...Program finished with exit code 139
Press ENTER to exit console.
```

```
onlinegdb.com
Run Debug Stop Share Save Beautify
Language C++

main.cpp
36 }
37 };
38
39 int main() {
40     // Creamos un vector de punteros a Figura
41     vector<Figura*> figuras;
42     // Agregamos diferentes objetos derivados al vector
43     figuras.push_back(new Circulo());
44     figuras.push_back(new Cuadrado());
45     figuras.push_back(new Triangulo());
46     figuras.push_back(new name());
47     // Recorremos la colección polimórfica
48     for (const Figura* f : figuras) {
49         f->dibujar(); // Llamada polimórfica a través de puntero base
50     }
51     // Liberamos memoria (importante para evitar fugas)
52     for (Figura* f : figuras) {
53         delete f; // Llama al destructor correcto gracias a que es virtual
54     }
55     return 0;
56 }
57 }
58 }

Input
Dibujar Circulo
Dibujar Cuadrado
Dibujar Triangulo
RODRIGO MUÑOZ PIZARRO

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com
main.cpp
class Cuadrado : public Figura {
public:
    void dibujar() const override {
        cout << "Dibujar Cuadrado" << endl;
    }
};

// Clase derivada adicional para contraste
class Rodrigo: public Figura {
public:
    void dibujar() const override {
        cout << "Rodrigo Munoz Pizarro" << endl;
    }
};

int main() {
    // Puntero de tipo base apuntando a objeto derivado
    Figura* figura1 = new Circulo();
    Figura* figura2 = new Cuadrado();
    Figura* figura3 = new Rodrigo();
    // Invocación del método virtual: se ejecuta dinámicamente
    figura1->dibujar(); // Salida: "Dibujar Circulo"
}

Dibujar Circulo
Dibujar Cuadrado
Rodrigo Munoz Pizarro

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com
main.cpp
void mostrar(int valor) {
    std::cout << "Tipo Entero (int): " << valor << std::endl;
}

void mostrar(double valor) {
    std::cout << "Tipo Decimal (double): " << valor << std::endl;
}

void mostrar(const std::string& valor) {
    std::cout << "Tipo Cadena (std::string): \"" << valor << "\"" << std::endl;
}

void mostrar(char valor) {
    std::cout << "Tipo Caracter (char): " << valor << " " << std::endl;
}

void mostrar(const std::vector<int>& miVector) {
    std::cout << "Tipo Vector de Enteros (std::vector<int>): [ ";
    for (size_t i = 0; i < miVector.size(); ++i) {
        std::cout << miVector[i] << (i == miVector.size() - 1 ? " " : ", ");
    }
    std::cout << " ]" << std::endl;
}

--- Demostracion de 'mostrar' sobrecargado ---
Tipo Entero (int): 100
Tipo Decimal (double): 3.14159
Tipo Cadena (std::string): "Hola desde Programacion III!"
Tipo Caracter (char): 'Z'
Tipo Vector de Enteros (std::vector<int>): [ 10, 20, 30, 40, 50 ]
Tipo Cadena (std::string): "Esto es un literal de C-string."

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com
main.cpp
30 }
31 };
32
33 // Función main para probar la clase
34 int main() {
35     std::cout << "=== Demostrando constructores ===" << std::endl;
36
37     // Constructor por defecto
38     Punto p1;
39
40     // Constructor con coordenadas
41     Punto p2(5.0, 3.0);
42
43     // Constructor de copia
44     Punto p3(p2);
45
46     std::cout << "\n=== Mostrando puntos ===" << std::endl;
47     p1.mostrar();
48     p2.mostrar();
49     p3.mostrar();
50
51     return 0;
52 }
Input
=== Demostrando constructores ===
Punto creado en el origen (0,0) por constructor por defecto.
Punto creado en (5,3) por constructor con coords.
Punto copiado de (5,3).

=== Mostrando puntos ===
Punto(0, 0)
Punto(5, 3)
Punto(5, 3)
Punto en (5,3) destruido.
Punto en (5,3) destruido.
Punto en (0,0) destruido.

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com
main.cpp
9 // Área de un rectángulo
10 double calcularArea(double base, double altura) {
11     std::cout << "Calculando área de RECTÁNGULO..." << std::endl;
12     return base * altura;
13 }
14
15 // Área de un cuadrado
16 double calcularArea(int lado) {
17     std::cout << "Calculando área de CUADRADO..." << std::endl;
18     return lado * lado;
19 }
20
21 int main() {
22     double areaCirc = calcularArea(5.0); // Llama a la función de círculo
23     double areaRect = calcularArea(4.2, 6.3); // Llama a la función de rectángulo
24     double areaCuad = calcularArea(7); // Llama a la función de cuadrado
25
26     std::cout << "Área del círculo: " << areaCirc << std::endl;
27     std::cout << "Área del rectángulo: " << areaRect << std::endl;
28     std::cout << "Área del cuadrado: " << areaCuad << std::endl;
29     std::cout << "Rodrigo Muñoz Pizarro" << std::endl;
30
31     return 0;
32 }
Input
Calculando área de CÍRCULO...
Calculando área de RECTÁNGULO...
Calculando área de CUADRADO...
Área del círculo: 78.5397
Área del rectángulo: 26.46
Área del cuadrado: 49
Rodrigo Muñoz Pizarro

...Program finished with exit code 0
Press ENTER to exit console.
```



```
onlinegdb.com

main.cpp
1 #include <iostream> // Para std::cout, std::endl
2 #include <string> // Para std::string
3
4 // Versión 1: Suma dos enteros
5 int sumar(int a, int b) {
6     std::cout << "Ejecutando sumar(int, int)... ";
7     return a + b;
8 }
9
10 // Versión 2: Suma dos números de punto flotante (double)
11 // ¡Sobrecargada! Mismo nombre, diferente tipo de parámetros.
12 double sumar(double a, double b) {
13     std::cout << "Ejecutando sumar(double, double)... ";
14     return a + b;
15 }
16
17 // Versión 3: Concatena dos cadenas (std::string)
18 // ¡Sobrecargada! Mismo nombre, diferente tipo de parámetros.
19 std::string sumar(const std::string& a, const std::string& b) {
20     std::cout << "Ejecutando sumar(const std::string&, const std::string&)... ";
21     return a + b;
22 }
23

input
Suma de enteros (5, 3): Ejecutando sumar(int, int)... 8
Suma de doubles (5.5, 3.3): Ejecutando sumar(double, double)... 8.8
Concatenación de strings ("Hola, ", "Mundo!"): Ejecutando sumar(const std::string&, const std::string&)... Hola, M
undo!
Suma de tres enteros (1, 2, 3): Ejecutando sumar(int, int, int)... 6

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com

ma Upload
File
26
27 int main() {
28     // Probar con int
29     int num = 50;
30     cout << "¿El int " << num << " es válido? " << (validarEntrada(num) ? "Sí" : "No") << endl;
31
32     // Probar con double
33     double dec = 5.5;
34     cout << "¿El double " << dec << " es válido? " << (validarEntrada(dec) ? "Sí" : "No") << endl;
35
36     // Probar con string
37     string texto = "RodrigoMunoz";
38     cout << "¿El string \" " << texto << "\" es válido? " << (validarEntrada(texto) ? "Sí" : "No") << endl;
39
40     // Probar con valores inválidos
41     cout << "\n-- Pruebas con datos inválidos --\n";
42     cout << "¿El int -5 es válido? " << (validarEntrada(-5) ? "Sí" : "No") << endl;
43     cout << "¿El double 12.3 es válido? " << (validarEntrada(12.3) ? "Sí" : "No") << endl;
44     cout << "¿El string \"Rodr!go Muñoz\" es válido? " << (validarEntrada("H01@") ? "Sí" : "No") << endl;
45     return 0;
46 }
47

input
¿El int 50 es válido? Sí
¿El double 5.5 es válido? Sí
¿El string "RodrigoMunoz" es válido? Sí

-- Pruebas con datos inválidos --
¿El int -5 es válido? No
¿El double 12.3 es válido? No
¿El string "Rodr!go Muñoz" es válido? No

...Program finished with exit code 0
Press ENTER to exit console.
```

```
onlinegdb.com

main.cpp
6 std::cout << " factorial(" << n << ") -> Caso Base! Retorna 1." << std::endl;
7 return 1;
8 }
9 // Paso Recursivo
10 else {
11 std::cout << " factorial(" << n << ") -> Paso Recursivo. Llama a factorial(" << n - 1 << ")." << std::endl;
12 long long resultadoRecursion = factorial(n - 1); // ¡Auto-Llamada!
13 long long resultadoFinal = n * resultadoRecursion;
14 std::cout << " factorial(" << n << ") -> Retornando " << n << " * "
15 << resultadoRecursion << " = " << resultadoFinal << std::endl;
16 return resultadoFinal;
17 }
18 }
19
20 int main() {
21 int numero = 10; // Probemos con 4!
22 std::cout << "Iniciando calculo del factorial de " << numero << "." << std::endl;
23 long long resultado = factorial(numero);
24 std::cout << "\nEl factorial de " << numero << " es: " << resultado << std::endl;
25 std::cout << "Rodrigo Muñoz Pizarro" << std::endl;
26
27 return 0;
28 }
```

input

```
factorial(5) -> Paso Recursivo. Llama a factorial(4).
Calculando factorial(4)...
factorial(4) -> Paso Recursivo. Llama a factorial(3).
Calculando factorial(3)...
factorial(3) -> Paso Recursivo. Llama a factorial(2).
Calculando factorial(2)...
factorial(2) -> Paso Recursivo. Llama a factorial(1).
Calculando factorial(1)...
factorial(1) -> Caso Base! Retorna 1.
factorial(2) -> Retornando 2 * 1 = 2
factorial(3) -> Retornando 3 * 2 = 6
factorial(4) -> Retornando 4 * 6 = 24
factorial(5) -> Retornando 5 * 24 = 120
factorial(6) -> Retornando 6 * 120 = 720
factorial(7) -> Retornando 7 * 720 = 5040
factorial(8) -> Retornando 8 * 5040 = 40320
factorial(9) -> Retornando 9 * 40320 = 362880
factorial(10) -> Retornando 10 * 362880 = 3628800

El factorial de 10 es: 3628800
Rodrigo Muñoz Pizarro

...Program finished with exit code 0
Press ENTER to exit console.
```

SERIE FIBONACCI

```
onlinegdb.com

main.cpp
1 #include <iostream>
2
3 // Función recursiva para calcular el número de Fibonacci en la posición n
4 int fibonacci(int n) {
5     if (n <= 0) {
6         return 0;
7     }
8     if (n == 1) {
9         return 1;
10    }
11    // Paso recursivo
12    return fibonacci(n - 1) + fibonacci(n - 2);
13 }
14
15 int main() {
16     int terminos = 10; // Calcular hasta F(6)
17
18     std::cout << "Secuencia de Fibonacci (primeros " << terminos << " terminos):" << std::endl;
19
20     for (int i = 0; i < terminos; ++i) {
21         std::cout << fibonacci(i) << " ";
22     }
23
24     std::cout << std::endl;
25
26     std::cout << "RODRIGO MUÑOZ PIZARRO" << std::endl;
27     return 0;
28 }
```

input

```
Secuencia de Fibonacci (primeros 10 terminos):
0 1 1 2 3 5 8 13 21 34
RODRIGO MUÑOZ PIZARRO

...Program finished with exit code 0
Press ENTER to exit console.
```

OnlineGDB

online compiler and debugger for c++

code compile, run, debug, share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Sign Up

Login

main.cpp

```
1 #include <iostream>
2 #include <vector>
3 // Suma los elementos de 'arr' desde el índice 'idx' hasta el final
4 int sumarArreglo(const std::vector<int>& arr, int idx) {
5     // Caso Base: Si el índice está fuera de los límites del vector,
6     // significa que no hay más elementos que sumar.
7     if (idx >= arr.size()) {
8         return 0;
9     }
10    // Paso Recursivo: Suma el elemento actual (arr[idx])
11    // con la suma del resto del arreglo (desde idx + 1).
12    else {
13        return arr[idx] + sumarArreglo(arr, idx + 1);
14    }
15 }
16 int main() {
17     std::vector<int> misNumeros = {10, 5, 15, 20, 50}; // Suma = 100
18     int sumaTotal = sumarArreglo(misNumeros, 0);
19     // Empezar desde el índice 0
20     std::cout << "La suma recursiva del arreglo es: " << sumaTotal << std::endl;
21
22     std::cout << "RODRIGO MUÑOZ PIZARRO" << std::endl;
23
24     return 0;
25 }
```

Input


La suma recursiva del arreglo es: 100
RODRIGO MUÑOZ PIZARRO

...Program finished with exit code 0
Press ENTER to exit console.

Language C++

AD

close ad [x]

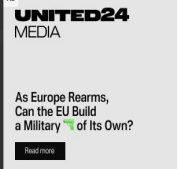


UNITED24
MEDIA

Los recursos minerales de Ucrania, valorados en 15 billones de dólares, son un objetivo prioritario para Rusia

Read more

AD



UNITED24
MEDIA

As Europe Rearms, Can the EU Build a Military of Its Own?

Read more

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy

© 2016 - 2025 GDB Online