



Ejercicio 7

Docente: Jimmy Nataniel Requena Llorentty

Materia: Programación III

Carrera: Ingeniería En Sistemas

Estudiantes: Joaquin Marcos Maita Flores

Santa Cruz – Bolivia

2025

CONSTRUYENDO OBJETOS DE MÚLTIPLES MANERAS

Constructores

- Métodos especiales con el mismo nombre que la clase.
- Se llaman automáticamente al crear un objeto.
- Su propósito es inicializar el objeto.

¡Los Constructores TAMBIÉN se pueden Sobrecargar!

- Esto permite crear objetos de la misma clase de diferentes maneras, proveyendo diferentes conjuntos de información inicial.

Ejemplo Conceptual: Clase Punto

```
# #include <iostream> // ¡Esta línea es la que faltaba!

class Punto {
public:
    double x, y;

    // Constructor 1: Por defecto (en el origen)
    Punto() : x(0.0), y(0.0) {
        std::cout << "Punto creado en el origen (0,0) por constructor por defecto." << std::endl;
    }

    // Constructor 2: Con coordenadas específicas
    Punto(double coord_x, double coord_y) : x(coord_x), y(coord_y) {
        std::cout << "Punto creado en (" << x << ", " << y << ") por constructor con coords." << std::endl;
    }

    // Constructor 3: Copia (se genera uno por defecto, pero podemos hacerlo explícito)
    Punto(const Punto& otroPunto) : x(otroPunto.x), y(otroPunto.y) {
        std::cout << "Punto copiado de (" << otroPunto.x << ", " << otroPunto.y << ")." << std::endl;
    }

    // Destructor (opcional, para demostrar el ciclo de vida)
    ~Punto() {
```

```
        std::cout << "Punto en (" << x << ", " << y << ") destruido." <<
std::endl;
    }

    // Método para mostrar las coordenadas
    void mostrar() const {
        std::cout << "Punto(" << x << ", " << y << ")" << std::endl;
    }

};

// Función main para probar la clase

int main() {
    std::cout << "=== Demostrando constructores ===" << std::endl;

    // Constructor por defecto
    Punto p1;

    // Constructor con coordenadas
    Punto p2(5.0, 3.0);

    // Constructor de copia
    Punto p3(p2);

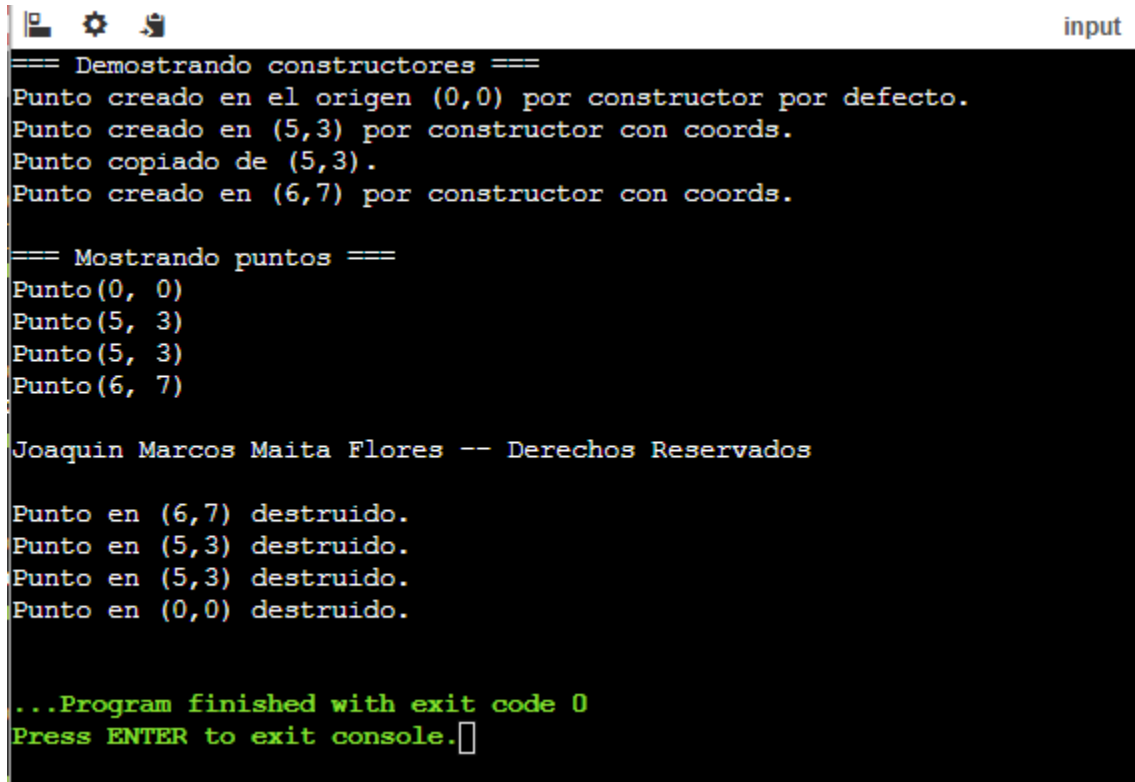
    //intentar crear un puntero, funcionara, pues pongamoslo a prueba
    Punto p4(6.0, 7.0);

    std::cout << "\n=== Mostrando puntos ===" << std::endl;
    p1.mostrar();
    p2.mostrar();
    p3.mostrar();
    p4.mostrar();

    std::cout<<" \nJoaquin Marcos Maita Flores -- Derechos
Reservados\n"<<std::endl;

    return 0;
}
```

CÓDIGO EJECUTADO



```
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).
Punto creado en (6,7) por constructor con coords.

=== Mostrando puntos ===
Punto(0, 0)
Punto(5, 3)
Punto(5, 3)
Punto(6, 7)

Joaquin Marcos Maita Flores -- Derechos Reservados

Punto en (6,7) destruido.
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.
```

Principales funciones del programa

- Sobrecarga de constructores: Diferentes formas de crear un objeto.
- Gestión de memoria: El destructor muestra cuándo se libera un objeto.