

# ALGORITMOS Y ESTRUCTURAS DE DATOS

## 2023/1

PROFESOR: Manuel Alejandro Moscoso Domínguez  
manuel.moscoso.d@gmail.com

---

### Laboratorio Semana 9

En esta oportunidad realizaremos actividades de introducción al lenguaje de programación C++ y el trabajo con clases.

#### Objetivos

- Resolver ejercicios que involucren la implementación de clases en C++.
- Desarrollar algoritmos que permitan entregar una solución a los problemas entregados.

### Ejercicios

#### Nomenclatura para nombre de archivos fuentes

El nombre del archivo en el cual se almacena el código fuente debe considerar el siguiente formato: labsemanaX.EXT donde; X corresponde a la semana y EXT a la extensión del lenguaje de programación utilizado.

[Ir a la ayuda memoria](#)

---

## Ejercicio número 1

Crear una clase llamada **Rectangle**, considerando dos atributos privados y los métodos públicos. Incluir además un método para calcular el área.

## Ejercicio número 2

Crear una clase llamada **Point**, considerando dos atributos privados y los métodos públicos para su definición y acceso. Implementar el uso de esta clase mediante la definición de objetivos a través del constructor y utilizando los métodos.

## Ejercicio número 3

Crea el algoritmo que permita almacenar un polígono dentro del problema. Para esto debe considerar el uso de Vectores para trabajar con posibilidad de tener más de un objeto de la clase Point.

## Ejercicio número 4

Crear una clase llamada LinkedList, considerando que contenta un atributo privado que apunte al primer nodo de la lista y los métodos públicos para insertar, buscar, eliminar e imprimir. Implementar el uso de las clases y pruebas de todas las funcionalidades.

## Ejercicio número 5

Crear una clase llamada **BinaryTree**, considerando que contenta un atributo privado que apunte a la raíz del árbol y los métodos públicos para insertar, buscar e imprimir. Implementar el uso de las clases y pruebas de todas las funcionalidades.

---

## Ayuda memoria

```
#include <iostream>
#include <vector>

using namespace std;

// Class declaration
class MyClass {
public:
    MyClass(); // Default constructor
    MyClass(int num); // Constructor with parameter
    ~MyClass(); // Destructor

    void myMethod(); // Method declaration
    void setNum(int num); // Setter method
    int getNum(); // Getter method

private:
    int num_; // Private member variable
};

// Default constructor definition
MyClass::MyClass() {
    num_ = 0;
    cout << "Default constructor called." << endl;
}

// Constructor with parameter definition
MyClass::MyClass(int num) {
    num_ = num;
    cout << "Constructor with parameter called." << endl;
}

// Destructor definition
MyClass::~MyClass() {
    cout << "Destructor called." << endl;
}

// Method definition
void MyClass::myMethod() {
    cout << "myMethod() called." << endl;
}

// Setter method definition
void MyClass::setNum(int num) {
    num_ = num;
}

// Getter method definition
int MyClass::getNum() {
    return num_;
}

int main() {
```

---

```
// Object creation using default constructor
MyClass obj1;

// Object creation using constructor with parameter
MyClass obj2(10);

// Method call
obj1.myMethod();

// Setter method call
obj1.setNum(5);

// Getter method call
cout << "num_ value of obj1: " << obj1.getNum() << endl;

// Vector of objects
vector<MyClass> vec;
vec.push_back(obj1);
vec.push_back(obj2);

return 0;
}
```