MD NABEEL
29/22/23

# OBJECTS, ATTRIBUTES AND BEHAVIOUR

**Objective:** The objective of this lab assignment was to delve into various aspects of object-oriented programming in C++, including access modifiers, static variables, const members, and arrays of objects.

**Learnings:**

- **Access Modifiers:** Explored the significance of **public** and **private** access modifiers within a class structure. Learned that private members can only be accessed within the class itself, while public members are accessible externally.
- **Class Constructors and Destructors:** Studied the implementation and role of default constructors, parameterized constructors, copy constructors, and destructors within a C++ class.
- **Static Variables and Const Members:** Understood the usage of static variables within a class to track object counts and const members to maintain constants within the class structure.
- **Arrays of Objects:** Investigated the creation and manipulation of arrays containing objects and visualized these objects on a 2D plane.

**Challenges:**

- Understanding the fine nuances of access modifiers initially presented some challenges, especially in distinguishing their specific scopes within the class.
- Working with static variables and const members required a careful understanding of their behavior within the class and their interaction with object instances.

**Key Notes:**

- Private members in a class are only accessible within the class itself, whereas public members allow interaction from external scopes.
- Static variables track class-wide information, such as object counts, while const members maintain constants within the class structure.

**Conclusion:** This lab assignment provided a comprehensive understanding of fundamental concepts in C++ object-oriented programming. It facilitated practical knowledge in managing class members, constructors, and working with arrays of objects.

MD NABEEL
29/22/23

# OBJECTS, ATTRIBUTES AND BEHAVIOUR

---

## CODE NOTES:

- Private members are only accessible within the class or by their friend classes. They are exclusive to the class.

- Public members are accessible anywhere the object is visible, being the least restrictive access level.

- Member functions have a scope limited to the class they belong to. They can access all the class members or its objects.

- The **const** keyword ensures that the member function does not modify the object on which it is called. Functions that don't modify the object are called accessor functions, while those that do are known as mutator functions. Accessor functions are often declared as **const**.

- Declaring functions as **const** prevents modification of the data members, indicating they are read-only and won't change the object's state. These functions can be called on **const** objects of the class.

- Static members are initialized before any object of the class is created. They aren't associated with a specific object and can be called using the class name with the scope resolution operator. Memory for static members is allocated only once when the class is defined and is shared by all objects.

- Static **const** members can also be initialized outside the class.

- The statement that **const** member functions cannot change the state of an object is false. While they cannot change the state of the object they are called on, they can alter the state of other objects of the same class.

- The commands for compilation and execution for question with graphics library:

  ```
  g++ -c access_modifier.cpp -I /usr/include/SFML/
  g++ access_modifier.o -o sfml-app -lsfml-graphics -lsfml-window -lsfml-system
  ./sfml-app
  ```