**C++ OOP Basics**

**What is OOP?**

OOP stands for Object-Oriented Programming. An object-oriented programming language uses objects in its programming. Programming with object-oriented concepts aims to emulate real-world concepts such as inheritance, polymorphism, abstraction, etc, in a program.

C++ language was designed with the main intention of adding object-oriented programming to C language. As the size of the program increases, the readability, maintainability, and bug-free nature of the program decrease. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

This was the major problem with languages like C which relied upon functions or procedures (hence the name procedural programming language). As a result, the possibility of not addressing the problem adequately was high. Also, data was almost neglected, and data security was easily compromised Using classes solves this problem by modeling the program as a real-world scenario.

**Difference between Procedure Oriented Programming and Object-Oriented Programming**

**Procedure Oriented Programming**

* Consists of writing a set of instructions for the computer to follow
* The main focus is on functions and not on the flow of data
* Functions can either use local or global data
* Data moves openly from function to function

**Object - Oriented Programming**

* Works on the concept of classes and object
* A class is a template to create objects
* Treats data as a critical element
* Decomposes the problem in objects and builds data and functions around the objects

Basically, procedural programming involves writing procedures or functions that manipulate data, while object-oriented programming involves creating objects that contain both data and functions.

**Basic Elements in Object-Oriented Programming**

* Classes - Basic template for creating objects. This is the building block of object-oriented programming.
* Objects – Basic run-time entities and instances of a class.
* Data Abstraction & Encapsulation – Wrapping data and functions into a single unit
* Inheritance – Properties of one class can be inherited into others
* Polymorphism – Ability to take more than one form
* Dynamic Binding – Code which will execute is not known until the program runs
* Message Passing – message (Information) call format

**Benefits of Object-Oriented Programming**

Object-oriented programming has many advantages. Listed below are a few.

* Programs involving OOP is faster and easier to execute.
* By using objects and inheritance, it provides a clear structure for programs and improves code reusability.
* It makes the code easier to maintain, modify and debug.
* Principle of data hiding helps build secure systems
* Multiple Objects can co-exist without any interference
* Software complexity can be easily managed so that even the creation of fully reusable software with less code and shorter development time is possible.