C++

**OBJECT ORIENTED PROGRAMING**

**Why Object-Oriented Programming?**

Before we discuss object-oriented programming, we need to learn why we need object-oriented programming?

* C++ language was designed with the main intention of adding object-oriented programming to C language
* As the size of the program increases readability, maintainability, and bug-free nature of the program decrease.
* This was the major problem with languages like C which relied upon functions or procedure (hence the name procedural programming language)
* As a result, the possibility of not addressing the problem adequately was high
* Also, data was almost neglected, data security was easily compromised
* Using classes solves this problem by modeling program as a real-world scenario

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

**Procedure Oriented Programming**

* Consists of writing a set of instruction 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

**Basic Concepts in Object-Oriented Programming**

* **Classes -**Basic template for creating objects
* **Objects –**Basic run-time entities
* **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 forms
* **Dynamic Binding –**Code which will execute is not known until the program runs
* **Message Passing –**message (Information) call format

**Benefits of Object-Oriented Programming**

* Better code reusability using objects and inheritance
* Principle of data hiding helps build secure systems
* Multiple Objects can co-exist without any interference
* Software complexity can be easily managed