CHAPTER-1

Introduction to object oriented programming language

What is procedural oriented programming (structured programming)

- ✓ **Procedural Oriented Programming (POP)** is a **programming style** where the program is divided into **well-structured steps and procedures**.
- ✓ It uses **different functions** to perform **different tasks** in a program. Each function carries out a specific job, and the program follows a **step-by-step approach** to solve problems.

For example:

The program is broken down into smaller parts (called steps or functions).

Each part does a specific job (like adding numbers, checking input, or showing output).

These parts are clearly organized, so the program follows a logical order — step by step

- ✓ Procedural programming follows a top-down approach, where a large program is broken into smaller functions and each function is executed step by step

Prepared by Ayush Lamsal | |
layush788@gmail.com

Contd..

- There are no access specifiers like private, public, or protected in procedural programming languages.
- Procedure-oriented programming is effective and saves time because the code is organized into reusable functions.
- The drawback of this type of programming language is that it does not supports data hiding, and managing large program can become difficult as the code grow.
- Procedural programming does not model real-world objects as effectively as object-oriented programming, where we can model real-world objects in the form of classes, which further describe their attributes and behaviors but the functions (in procedural programming) are action oriented that perform the certain actions (like performing calculation)

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics of procedure-oriented programming

Top-Down Approach

The program is broken down into smaller, manageable functions or procedures, starting with the main function and branching out.

Emphasis on Functions

The core of POP is the use of functions that perform specific tasks. Each function is written to handle a particular part of the problem.

Sequential Execution

The program executes line by line, following a specific order of instructions from top to bottom.

Global Data

In POP, data is usually global and used by all functions. There's no data encapsulation as in object-oriented programming which impacts security

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics of procedure-oriented programming

Reusability

Functions can be reused in different parts of the program, which helps reduce redundancy.

Modularity

The program is divided into modules (functions), making it easier to manage and maintain.

No Support for Inheritance or Polymorphism

Unlike object-oriented programming, POP does not support concepts like inheritance, polymorphism, or encapsulation.

Focus on Actions

Functions are designed to perform specific actions or operations on data, rather than modeling real-world objects.

Prepared by Ayush Lamsal | |
layush788@gmail.com

Object oriented programming

- ✓ Object-oriented programming (OOP) is a type of programming language that is based on the concepts of objects and classes.
- ✓ **Classes** define the structure of the objects, providing templates for their attributes and methods.
- ✓ Object is a instance of a class that represents a real world entity, and each object represents a unique entity.

For example, previously we created one object of the Student class, and that particular object holds the attributes of one student (entity).

- ✓ Object-oriented programming also supports important features such as encapsulation, inheritance, polymorphism, and abstraction.

Prepared by Ayush Lamsal | |
layush788@gmail.com

Object oriented programming

- Object-oriented programming (OOP) was introduced to solve some of the problems found in procedural programming.
  - *One major problem in procedural programming is that data is not well protected — it can be accessed and modified by any function in the program.*
  - *To fix this, OOP treats data as a critical element and focuses on protecting it.*
  - *It does this by using classes and objects, where data and functions are bundled together, and access to data is controlled using access specifiers like private, public, and protected.*
- Object –oriented programming (OOP) us a bottom-up approach. This means that:
  - *It starts by designing small parts first (like class and objects)*
  - *Then these small parts are combined to build a complete program*
  - ***The bottom-up approach is used when the details of the sub-problems are known, but the main problem is not yet clearly defined.***

> ***Note:*** *The* ***top-down approach*** *is the opposite. In this approach, we already know the application we want to build, but since it is complex, we divide it into smaller sub-problems to solve step by step.*

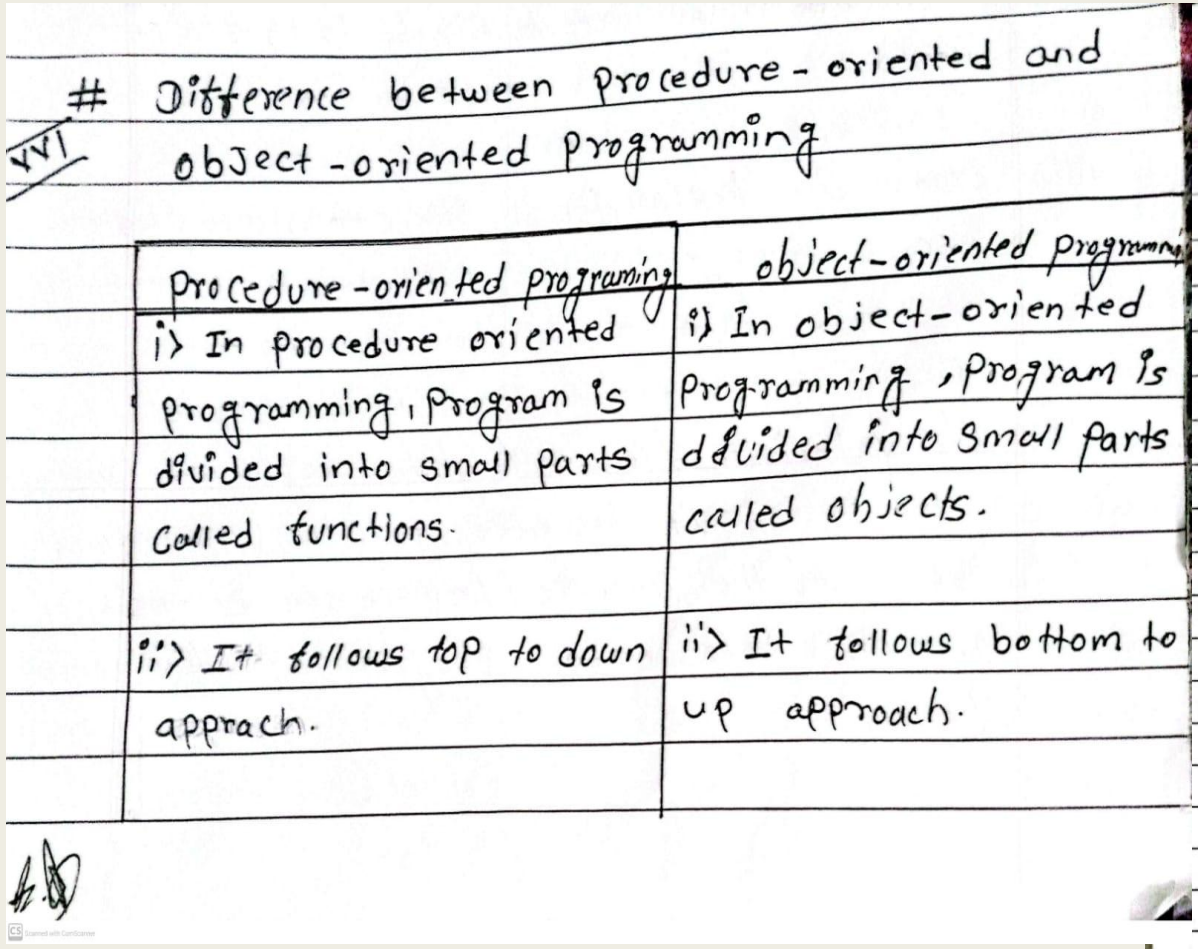
Prepared by Ayush Lamsal ||
layush788@gmail.com

Feature of Object oriented programming

- Follows a button-up approach in program design
- Program are divided into parts known as object
- Access specifiers are used.
- Data is hidden and cannot be accessed by external function.
- New data and functions can be easily added when ever necessary
- It supports inheritance that allows one class(sub class) to inherit properties and method form another class (super class)
- Object of one class may interact with object of another class by sending a message to access their data and method
- object-oriented programming models real-world concepts more effectively

Prepared by Ayush Lamsal | |
layush788@gmail.com

Difference between procedure –oriented and object –oriented programming



Difference between procedure-oriented and object-oriented programming

Procedure-oriented programing	object-oriented programing
i) In procedure oriented programming, program is divided into small parts called functions.	i) In object-oriented Programming, program is divided into small parts called objects.
ii) It follows top to down apprach.	ii) It follows bottom to up approach.
iii) There is no access specifier in procedural programming	iii) Programming have access specifies like private, public and protected.
iv) Adding new data and function is not easy.	iv) Adding new data and function is easy.
v) It does not have any proper way for hiding data so it is less secure	v) It provides access specifier private for hiding data so it is more secure.
vi) overloading is not possible in procedure oriented.	vi) overloading is possible in object-oriented.
vii) It is based on unreal world	vii) It is based on real world.
viii) Examples: C, FORTON, Pascal etc.	vii) C++, Java, python, C# etc are its examples.

Prepared by Ayush Lamsal ||
layush788@gmail.com

Characteristics / term of object oriented language

It is necessary to understand some of the concept /term used extensively in object oriented programming language which are as follows:

1. Objects:

- ✓ Objects are the basic runtime entities in an object-oriented system. They are created at runtime and represent real-world entities more closely
- ✓ A class itself does not occupy memory, but when an object is created, memory is allocated because the object stores the values defined by the class.
- ✓ An object is an instance of a class, which means it belongs to a specific class and cannot exist independently. An object is created from the class, and its properties and behavior are defined by the class.

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

6_simple_class_object.cpp > main()

```cpp
#include<iostream>
using namespace std;

class person{

    // These are the data member  which is kept private by default and we cannot access form outside the class
    char name[20];
    int id ;

    // These are the member function that which is kept public and can be accessed from outside the class
    // member function are awlays(mostly) kept public
    public:
    void get_details()
    {

    }
};
int main (){
    person p1; // p1 is the object created at a runtime and memory is allocated to it
}

```

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

2. class:

A class is a blue print or a template for creating object. It defines the structure and behavior of objects that will be created from it.

Class in an object oriented programing is a user defined data type that allows us to group variable of a different type under a single name. These variable are also known as a member variable.

Ones the class has been defined we can create any number of the object of that class so that the class is a reusable component. So we can also call the class as a collection of object of a similar type (i.e. all the object belongs to same class in a collection).

The syntax of creating a class is:

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

We can use private, public and protected access specifiers according to out need in program.

If we don't provide these access specifier to the data member than by default these data member will be private

```
#include<iostream>
using namespace std;


class class_name{
    private:
    // data member and member function;

    public:
    //data member and member function

    protected:
    // data maember and member function
}
```

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

Data abstraction (Hiding) and encapsulation:

- The wrapping up (or combining) of data and function into a single unit is known as encapsulation.
- The data is not accessible to the outside world and only these function which are wrapping in the class can access it.
- The insulation of data from direct access is called data hiding or information hiding

Inheritance:

- Inheritance is the process by which the object of one class acquire the properties of object of another class . It helps to share the common characteristics with the class from which it is derived
- The concept of inheritance provides the idea of reusability. This means that we can add additional feature to an existing class without modifying it

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

Polymorphism (Or overloading):

The word **"Polymorphism"** comes from the Greek words **"poly" and "morphē",** where "poly" means "many" and "morphē" means "form".

Polymorphism means ability to make more than one form.

Any operation show different behavior in different needs

For example

Think about the + (plus) sign:

- *2 + 3 ⟶ gives 5 : (It adds numbers)*
- *"Hello" + "World" ⟶ gives "HelloWorld“ :(It joins strings)*

- Same operator, but different behavior depending on the data type.

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

Dynamic Binding:

Dynamic binding is the process of deciding **which method to call while the program is running** (at runtime).

Dynamic binding is especially used in **inheritance**, when a **child class overrides a method** of the **parent class**. It allows the program to decide **at runtime** which method (parent's or child's) should be called.

For example:

- When both the parent and child class have a method with the same name (e.g., sound()), and the method is called using a parent class reference or pointer, then:

➡ Dynamic binding means:

- The program decides at runtime whether to call the parent's method or the child's method.

Prepared by Ayush Lamsal | |
layush788@gmail.com

Characteristics / term of object oriented language

Message passing:

Message passing is a characteristic of object-oriented programming where **one object communicates with another object to share information or request actions**.

- **While defining message passing, we need to specify:**
- **The object name:** The object that will receive the message
- **The method (function) name:** The function or method the object should execute
- **The information (arguments):** Any data the object needs to pass is sent as arguments

Prepared by Ayush Lamsal | |
layush788@gmail.com

Benefits / Advantage of OOP

- **Eliminates Redundant Code:**
  Through inheritance, OOP helps remove duplicate code and extends the use of existing classes.
- **Data Security:**
  Encapsulation helps programmers build secure programs by protecting data from unauthorized access by other parts of the program.
- **Easy Work Partitioning:**
  Work can be divided into classes and objects, making teamwork and project management easier.
- **Manages Software Complexity:**
  The modular approach divides programs into smaller, manageable parts, simplifying complexity.
- **Easy to Upgrade:**
  Object-oriented systems can be scaled up smoothly from small to large systems.
- **Saves Development Time:**
  OOP increases programmer productivity by reducing development time.
- **Models Real-World Problems:**
  OOP helps solve real-world problems by representing real-world objects and scenarios through classes and objects, making programs easier to understand and design.

Prepared by Ayush Lamsal | |
layush788@gmail.com

Application area of OOP

- Used for developing a real time system.
- OOP is widely used in game development
- OOP is also used in object-oriented databases
- OOP is used in Artificial Intelligence (AI) and Machine Learning:
- OOP is widely used in simulation and modeling, as it represents the real world scenarios in more better way
  - *Note: **Simulation** means analyzing and mimicking how real-world objects or systems work, to study their behavior in different situations without experimenting in the real world.*
- Used for developing large enterprise application like CRM system.
- OOP is also used in web development

Prepared by Ayush Lamsal | |
layush788@gmail.com