OOP's with C++

Page No.
Date

Unit I

1) Introduction to programming concepts
2) Data types, Data input, output, and operators
3) Decision making, loops, arrays & strings
4) Unified modelling language.
5) classes, abstraction, encapsulation

Unit II

1) constructors and destructors
2) working with objects
3) polymorphism
4) modeling relationships in class diagram

Unit III

1) Inheritance
2) modeling relationships
3) Runtime polymorphism
4) pointers
5) file handling
6) Applying OOP to solve real life applications

Books:

1) Object Oriented programming with C++
author - Balguruswamy & 8th edition, McGraw Hill

4/12/22

Unit I

Page No.
Date

1. Introduction to programming Concepts

Programming techniques

1) machine level programming language
2) Assembly level language programming
3) high level programming

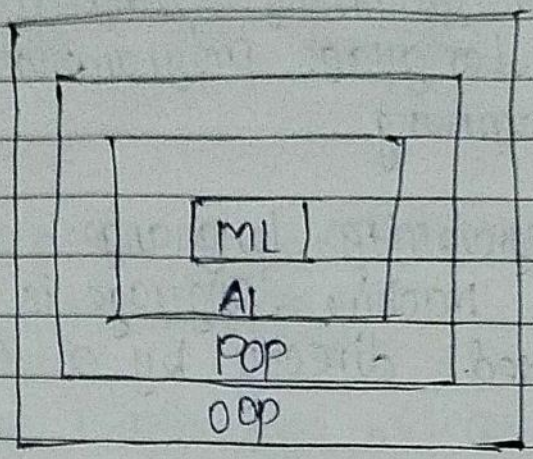
1) Machine level programming language
Machine code or machine language is a set of instructions executed directly by a computer's CPU

2) Assembly language
- It is a low level programming language for a computer or other programming device.
- There is very strong generally one-one correspondence between the language and the architectures machine code instruction.
- assembly language is converted into executable machine code or instruction by a utility program refered to as assembler
- The conversion is known as assembling

3) High level language.
- High level language is any programming language that enables development of a program in much simpler programming context
- It focuses more on programming logic rather than underlying hardware components memory addressing and register utilization

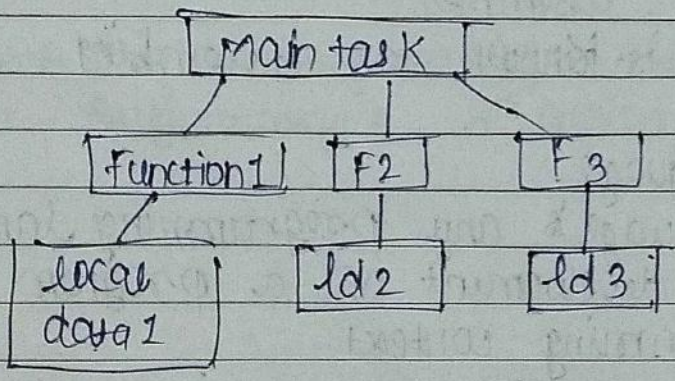
High level language is broadly classified into 2 types

1) Procedure oop (POP)
2) Object oriented program ming (OOP)



3/12/22

1) Procedure Oriented programming
- It is conventional programming which uses high level languages such as COBOL, FORTAN, C
- In POP the problem is viewed as a sequence of things to be done ie reading, calculating & printing
- A no. of functions are written to accomplish this task, so the primary focus is on functions



disadvantages:
- It does not model real world problems very well
- there is no data hiding

Page No.
Date

Analysis of POP

- concentration is on the development of functions, very little attention is given to the data that is being used by function
- Data moves freely between functions where there is chance for the intruder / hacker to hack the data.
- Critical application can't be design using this approach

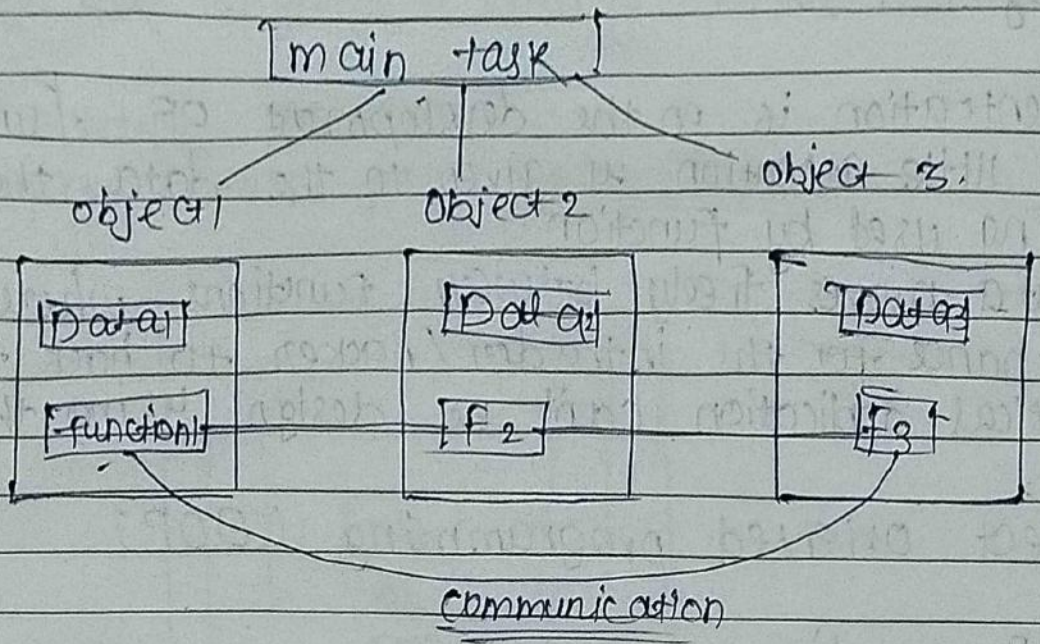
2) Object oriented programming (OOP)

OOP paradigm

1) The measure motivating factor in the invention of OOP approach is to remove sum of flaws encountered in the procedure approach
2) OOP treats data as a critical element in the program development and doesn't allow data to flow freely around the system
3) It ties the data more closely to the functions that operates on it and protects the data from accidental modification from outside functions
4) It provides data hiding features

Analysis of OOP's / working of OOP

- OOP allows decomposition of a problem into no. of entities called objects and then builds data and functions around this objects
- Data of an object can be accessed only by the functions associated with that object
- However functions of one object can access the functions of other objects.



5/12/22

C++

- It was developed by ~~then~~ Bjarne Straustrup at bell laboratories in 1979
- since C++ is an attempt to add oop features in C, earlier it was called as 'c with objects'.
- later in 1983 Stroustrup named it as 'C++'

Advantages of C++

1) Portability
C++ provides this feature of portability by allowing us to develop codes without ~~carrying~~ caring about the hardware

2) mid-level programming language
It can treat a low level and high level language which help to develop games and desktop application Whereas features of low level language helps to make

iostream / conio } libraries ← It consist some functions

Paradigm :- Systematic manner of writing something

Page No.
Date

kernels and drives.

3) Object oriented
The oop concepts like polymorphism, encapsulation, inheritance and abstraction gives c++ the biggest advantage is over other programming languages

4) Multi-paradigm programming language
Paradigm refers to planning involved in programming. It is concern with the logic, the style and the way how we proceed with the program. C++ is a multi-paradigm programming language as it follows 3 paradigm: 1) generic : using a single idea that serves multiple purpose
2) imperative : using steps that change the state of the program
3) object oriented : using methods and class for reusability and modularity

5) wide range of application
C++ is useful to make GUI's as well as games also it is useful to develop graphics and real time algebraic simulation. Hence C++ is beneficial in every stream

// ← used to comment a line

/* ... */ ← used to comment a paragraph.

Page No.
Date

Syntax of C++

```
#include <iostream.h>
#include <conio.h>
using namespace std;
int main()  (main function)
{
  Cout << "Hello Fycs";
  return 0;
}
```

Object — operator

6/12/22

Basic concepts of OOP's

(Void Function) procedure call

1) Objects
2) class
3) Data abstraction
4) Data encapsulation
5) Inheritance
6) Polymorphism (operator overloading, function overloading)
7) Dynamic binding
8) Message passing

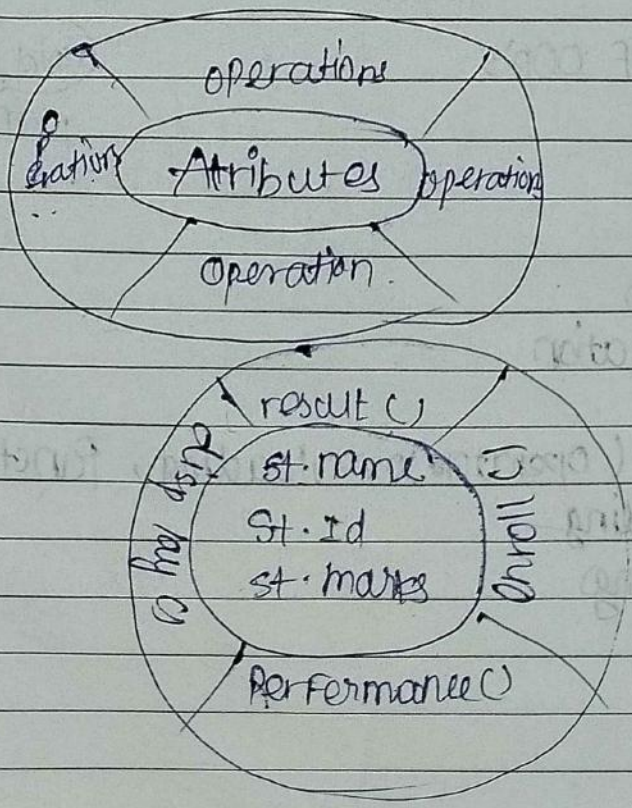
8/12/22

Page No.
Date

Basic concepts of OOP's

1) Object

- Objects are the basic runtime entities in an object oriented system
- They maybe represented as a person, a place, a bank account, a table of data or any item that the program must handle
- The fundamental idea behind object oriented approach is to combine both data and function into a single unit and this units are called objects
- The term objects means a combination of data and program that represent some real world entity

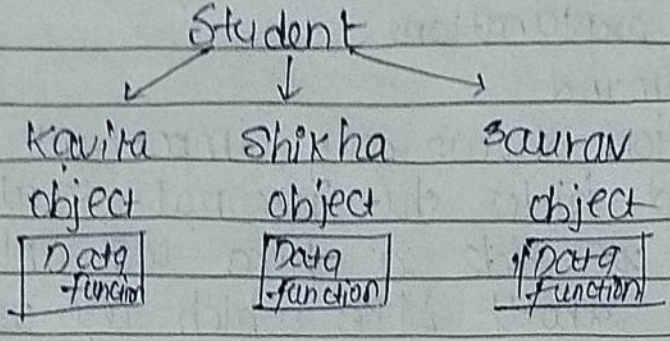


2) Classes

- It is a group of object that share common properties for data part and some program part are collectively called a class

- In c++ a class is a new data-type that contains member-functions and member variables

eg:-



Object	class
1) It is an instance OF a class	1) It is a blueprint or template from which objects are created
Objects	
2) In real world entity are pen, laptop, mobile	2) It is a group of similar Object.
3) It is a physical entity	3) It is a logical entity
4) Object is created through new keyword eg: Student S1 = new student;	4) Class is declared by using class keyword eg: class A
5) Objects allocate memory when it is created	5) Class doesn't allocate memory when it is created.

3) Data abstraction

- It refers to the act of representing essential features without including the background details or explanations
eg:- ~~Background~~
- Abstraction shows only important things to the user and hides the internal details
eg: When we ride a bike we only know some features about bike which are used for riding, but the internal details of functionality is hidden (not known by us)

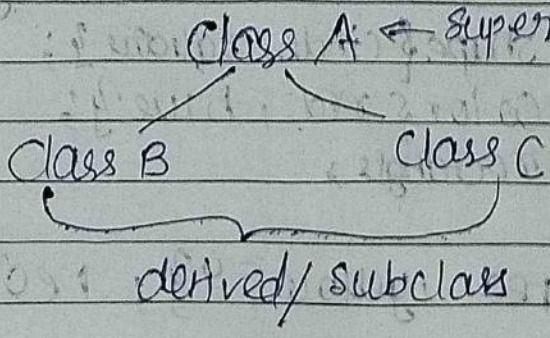
4) Data encapsulation

- The wrapping of data and functions into a single unit (called class) is known as encapsulation.
- The data is not accessible to the outside world and only those functions which are wrapped in the class can access it
- This functions provide interface between the objects data and program
eg:-

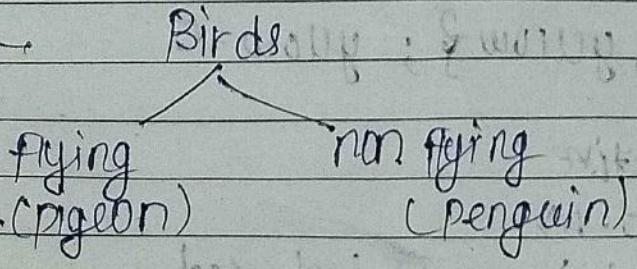
class	student
attribute :	st.name, st.Id, st.marks
functions :	result(), display() enroll(), performance()

5) Inheritance

- It is the process by which object of one calass acquires the properties of another class.
- here, inheritance provides the idea of reusability. this means that we can add additional features to an existing class without modifying it.
- This is possible by designing a new class which will have the combined features of both the classes.
- The existing class is called as base class or super class and the new class is called as sub class or derived class



eg :-



Application OOP's

1) Object oriented Database (OODBMS)
2) real time system design
3) Simulation & modeling System
4) Hypertext and Hypermedia
5) AI expert ~~system~~ media
6) decision support and

18/12/22

Page No.
Date

Tokens

1) Reserved ⟶ Predefined
2) Identifiers ⟶ user defined
3) literals
→ A sequence of charaders
i) integer ii) character iii) Float iv) string
4) Constant
keyword ⟶ const
eg:- const a = 10;
5) enum (enumeration)
eg:- enum (x, y, z);
x = 0, y = 1, z = 2 By default it starts From 0 and if we want we can assign value also
eg:- enum shape { circle, square };
enum color { red, blue };
shape rectangle;

enum { red, yellow = 9 }; red = 0

enum { red = 4, yellow }; yllow = 5

preprocessor directive
eg:- # include <conio.h>
here semicolon (;) is not included

4) Operators

Special operators

1) :: scope variable member declaration
2) :: * pointer to ~~word destination~~
3) → * pointer to ~~word~~ member operator
4)

→ It is used to give newline

Page No.
Date

5) endl. line feed operator
6) new

1) :: scope variable
Scope resolution operator is used to recover a hidden value

5) Separators