GOVERNMENT ARTS COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMMERCE (CA)

CLASS: I M.COM (CA) **SEMESTER II**

SUBJECT CODE: 18MCC23C

OBJECT ORIENTED PROGRAMMING WITH C++

Unit 1:

Evaluation of programming paradigm- Elements of Object Oriented programming- Data Encapsulation and Abstraction classes- Inheritance-Derived classes Polymorphism- Operator overloading – Friends functions- Polymorphism-virtual functions- Merits and demerits of OOP – Popular OOP languages- C++ at a glance- Applications of C++- C++ statements – Structure of C++ program.

Reference Books:

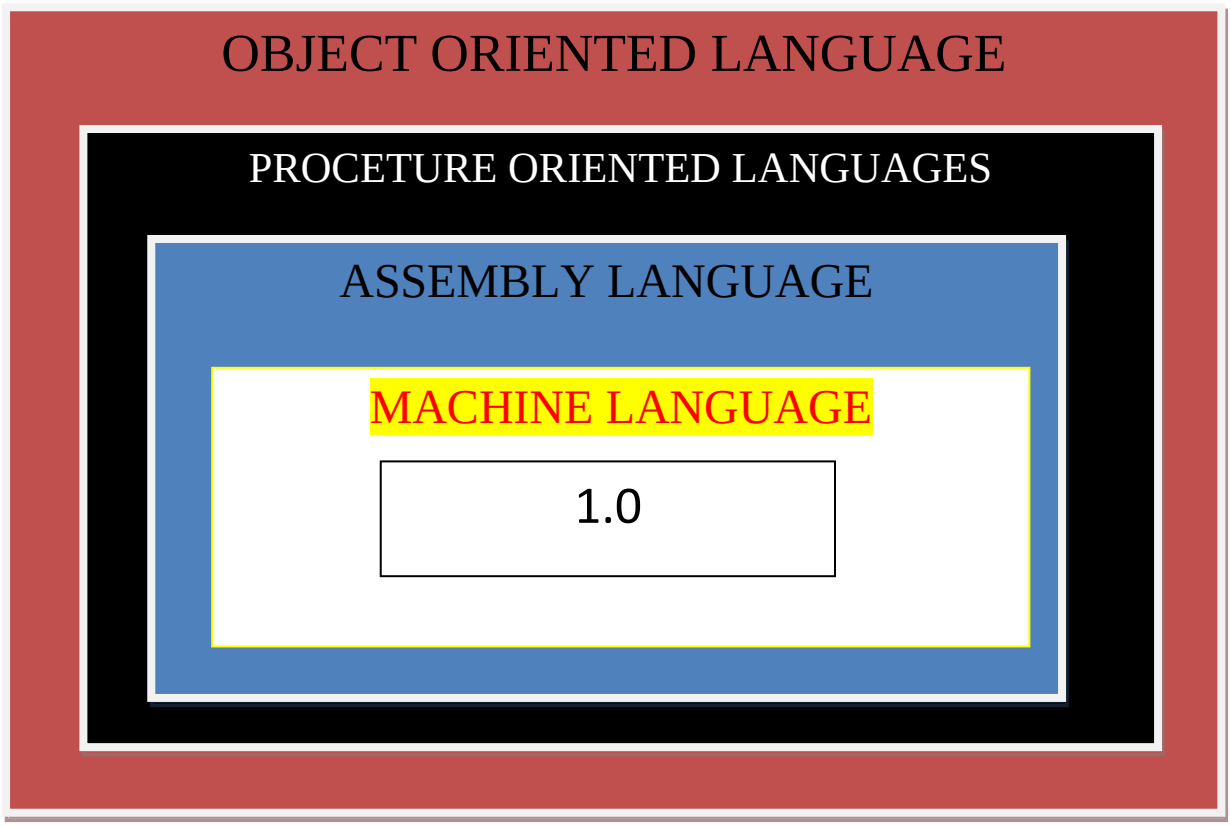
E. Balagurusamy, “Object Oriented Programming with C++”, TataMcGraw Hill Publishing Company Ltd.

K.R.Venugopal, Raj kumar, T.Ravishanker., “Mastering C++”,TataMcGraw Hill Publishing Company Ltd.

Prepared by

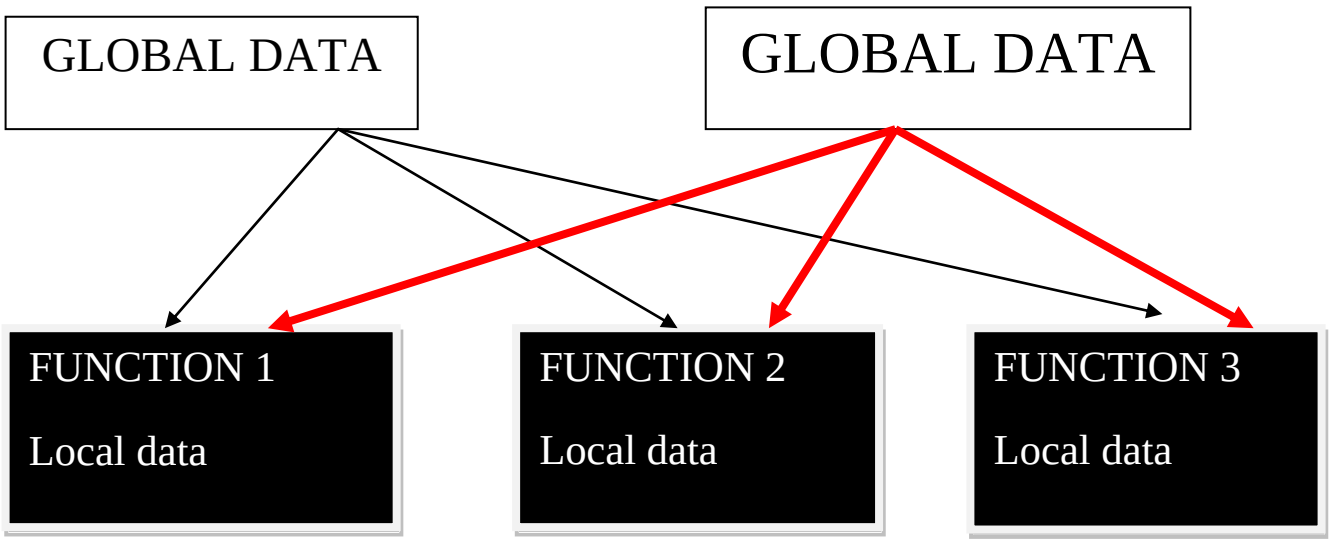
Dr. S. Vasantha

LAYER OF COMPUTER SOFTWARE



Computer programming using high level languages such as COBOL, FORTRAN and C. These are commonly known as procedure oriented programming languages.

- These programs are based on algorithms.
- Larger programs are divided into smaller programs known as functions.
- Employs top down approach in program design.

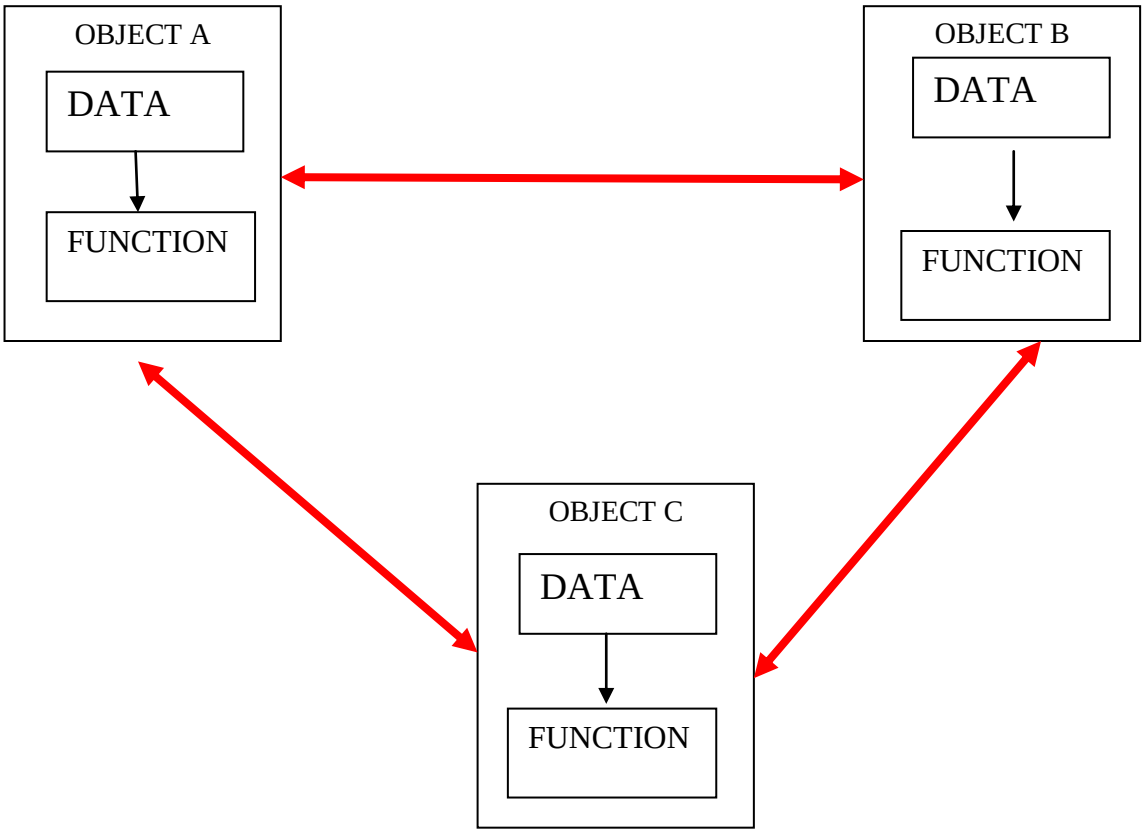


Object Oriented Programming

OOP allows decomposition of a problem into a number of entities called objects and builds data and functions around these objects.

FEATURES OF OOP

- Programs are divided into what are known as objects.
- Objects may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Flows bottom up approach in program design.



BASIC CONCEPTS OF OBJECT ORIENTED PROGRAMMING

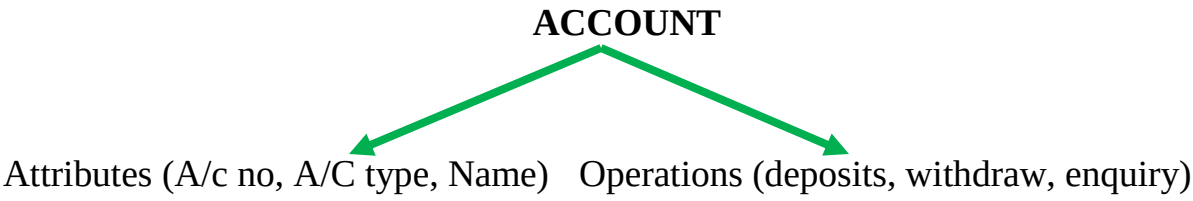
- Objects
- Classes
- Data abstraction and encapsulation
- Inheritance
- Polymorphism
- Dynamic binding
- Message passing
-

Object

Object is a real time entity. An object can be a person, place, student, employee inventory etc.

Every object will have data structure data structure called attributes and behavior called operations.

EXAMPLE



Classes:

Collection of object is called class

The objects with the same data structure attributes and behaviors (operations) are grouped into a class.

EXMPLE

Student_class

Attributes: Name, Age, Sex.

Operations: Speak (), Listen(), Walk(), Read()

SYNTAX:

Class class name

{

Private: Data members;

Public: Member functions;

};

EXMPLE

```
Class Account
{
Private: char name [20];
         int A/C type;
         int A/C Number;
         float balance;
Public: void deposit ();
        Void withdraw ();
        Void enquire ();
};
```

ENCAPSULATION AND DATA ABSTRACTION

Encapsulation is a Mechanism that associates the code and the data it manipulated and keeps them safe from external interference and misuses.

DATA ABSTRACTION

Creating new data types using encapsulated items that are will suited to an application to be programmed.

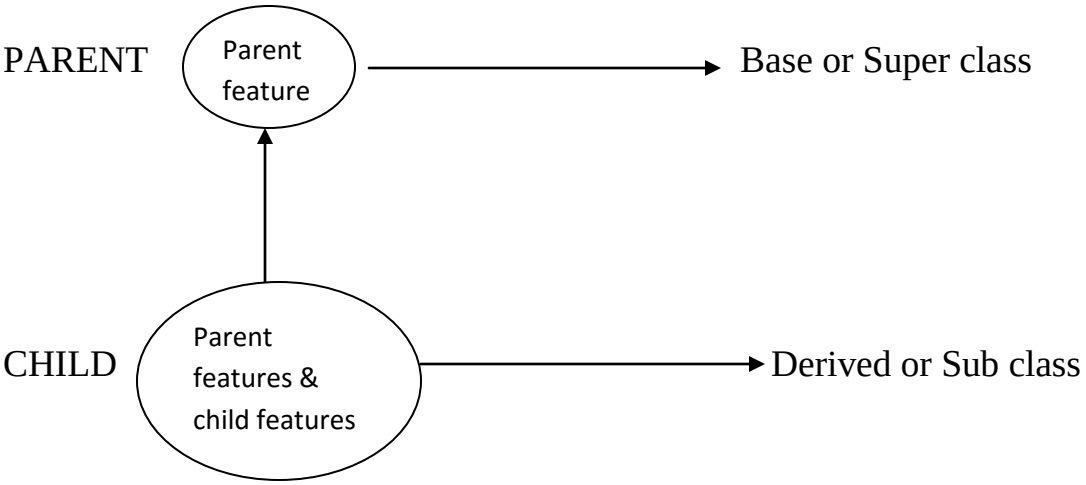
Data abstraction is a powerful technique and its proper usage will result in optimal, more readable and flexible programs.

INHERITANCE

The technique of building new classes form the existing classes is called inheritance.

- Existing class – Base or Super class
- New class – Derived or Sub class

The derived class is acquired the properties from the base class is called inheritance.



POLYMORPHISM

The meaning of polymorphism is many forms. Polymorphism allows a single name to be used for more than one related purpose, which are technically different.

EXMPLE

<< This symbol is called insertion (put to operator) or cout(generally less than symbol)

>> This symbol is called extraction operator cin(generally greater than symbol)

DYNAMIC BINDING

Dynamic binding (also called late binding) means that the code associated with a given procedure call is not known until its call at run time.

MESSAGE COMMUNICATION

A message for an object is interpreted as a request for the execution of a function.

A student is treated as an object sending the message marks to find the marks secured by the student with the specified Roll No.

STUDENT.MARKS(ROLL NO)

In the above line the **STUDENT** is **object** and the **MARKS** is **message** and the **(ROLL NO)** is **information.**

In this case a function call Marks() is treated as a message and a parameter. Roll No is treated as information passed to the object.

BENEFITS OF OOP

OOP offers several benefits to both the program designer and the user.

- The new technology promises greater programmer productivity better quality of software and lesser maintenance cost.
- It leads to saving of development time and higher productivity.
- Through inheritance, it can eliminate redundant code and extend the use of existing classes.
- The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
- It is possible to map objects in the problem domain to those in the program.
- It is easy to partition the work in a project based an objects.
- The data centered design approach enables us to capture more details of a model in implementable form
- Object oriented systems can be easily upgraded from small to large systems.
- Message passing techniques for communication between objects makes the interface descriptions with external systems much simpler.
- Software complexity can be easily managed.

APPLICATIONS OF OOP

OOP has become one of the programming word today

- Software engineers great deal of excitement and interest using the OOPs

- ❖ It is most popular application in the area of user interface design such as windows.
- ❖ Real- business systems are more used the oops.
- ❖ Real time systems.
- ❖ Simulation and modeling.
- ❖ Object oriented database
- ❖ Hypertext, hypermedia and expertext.
- ❖ AI and expert system.
- ❖ Neural networks and parallel programming.
- ❖ Decision support and office automation systems.

C++ PROGRAM

- C++ allows programmers to build large programs with clarity, extensibility and ease of maintenance.
- C++ provides bottom-up object oriented design
- C++ allows us to create hierarchy-related objects.
- It can be build special object oriented libraries which can be used later by many programmers
- C++ is able to map the real world problem property.
- C++ program are easily maintainable and ecxpandable.

SIMPLE C++ PROGRAM

// C++ program → statement comment

#include<iostream> → preprocessor directory.

Void main () → function declaration

{ → Function starting

Cout<< “C++ is better than c \n”; → statement

Return 0;

} → function close

FIRST LINE

// C++ program ⟶ comment statement

// this symbol is called double slash.

Comments start with a double slash symbol and terminate at the end of line.

There is no closing symbol.

Multiple comments can be written as follows:

// this is an example

//C++ program

// loop statement starting

SECOND LINE

THE IOSTREAM FILE

C++ program contains pre processor directive statements at the beginning.

Such statements are pre coded with a # symbol to indicate the presence of a pre processor directive to the complier.

#include<iostream>

The header file io stream should be included at the beginning all programs that use input/ output statements.

MAIN()

C++ program is a collection of functions all functions contain in the main().

As usual execution begins at main().

The C++ statements terminate with semicolons.

OUTPUT OPERATOR

Cout<< "c++ is better than c.";

Cout<< the identifier cout is a predefined object that represented the standard output stream in c++.\

Here, the standard output stream represents the screen. It is also possible to redirect the output to other output devices.

The operator << is called the insertion or put to operator. It inserts or send the contents of the variable on its right to the object on its left.

INPUT OPERATOR:

Cin>> number;

It is an input statement and causes the program to wait for the user to type in a number.

The identifier Cin is a predefined object in C++ that corresponds to the standard input stream.

This stream represents the key board

The operator >> is known as extraction or get from operator. It extracts or takes the value from the key board and assigns it to the variable o its right.

STRUCTURE OF A C++ PROGRAM

INCLUDE FILES
CLASS DECLARATION
MEMBER FUNCTIONS DEFINITIONS
MAIN FUNCTION PROGRAM

EXAMPLE: USE OF CLASS

```
#include<iostream.h>
Class person
{
Private:
        Char name [30];
        Int age;
Public:
        Void get data (void);
        Void display (void);
};
Void person::get data (int x, int y)
{
Cout<< “ENTER THE NAME:”;
Cin>> name;
Cout << “ENTER THE AGE:”;
Cin>>age;
}
Void person::display (void)
{
Cout<< “\n Name:”<<name;
Cout<< “\nAge:”<<age;
}
```

```
Int main()
{
Person p;
p.getdata();
p.dispaly();
return 0;
}
```