

Object Oriented Programming through C++ (II-I CSE & IT- R19)

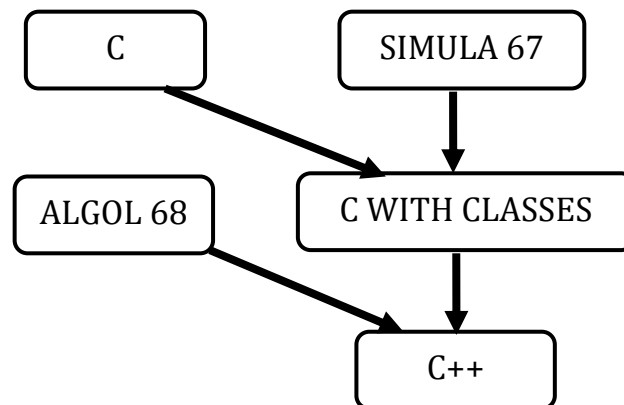
UNIT - 1

Introduction to C++ , Difference between C and C++- Evolution of C++- The Object Oriented Technology- Disadvantage of Conventional Programming- Key Concepts of Object Oriented Programming- Advantage of OOP- Object Oriented Language.

Introduction to C++ / Evolution of C++ :

C++ is an object-oriented programming language. C++ is an extension of C. It was developed by Bjarne Stroustrup at AT&T Bell Laboratories in Murray Hill, New Jersey, USA, in the early 1980's. Stroustrup combines the features of both C and Simula 67 and developed a powerful language which supports OOP with features of C. Various features were derived from **SIMULA67** and **ALGOL68**. Stroustrup called the new language "C with Classes".

In 1983, "C with Classes" was renamed to "C++" . The thought of C++ came from the C increment operator ++. Rick Mascitti coined the term C++ in 1983. C++ is superset of C. All the concepts of C are applicable to C++ also. For developing complicated applications, Object Oriented language such as C++is the most convenient and easy.



Difference between C and C++:

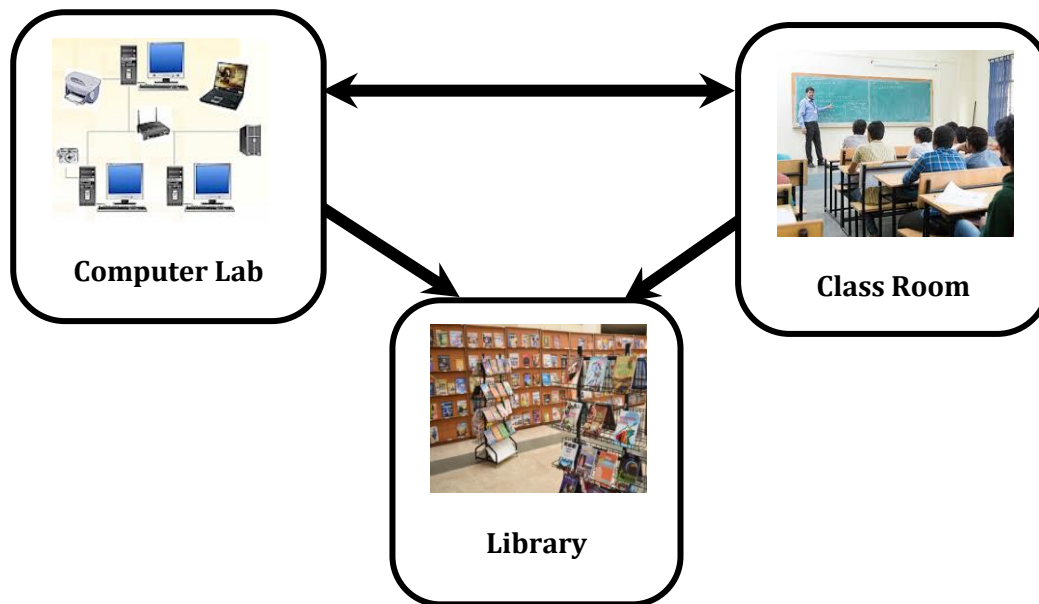
C	C++
It is Procedure Oriented Programming language	It is Object Oriented Programming language
C was developed by Dennis Ritchie between 1969 .	C++ was developed by Bjarne Stroustrup in 1979 .
C is a subset of C++.	C++ is a superset of C. C++ can run most of C code while C cannot run C++ code.

Data is not protected in C	Data is secured in C++ using data hiding concept
C uses stdio.h file for input and output functions	C++ uses iostream.h for these functions
Constructor and Destructor are absent in C	Constructor and Destructor are present in C++
Inline functions can be used as macros in C	Inline functions are supported by C++
In C , function overloading is available	In C++, function overloading and operator overloading also possible.
1. scanf() Function used for Input. 2. printf() Function used for output.	Cin>> Function used for Input. Cout<< Function used for output.
Top down approach is used in Program Design.	Bottom up approach adopted in Program Design

The object oriented Technology:

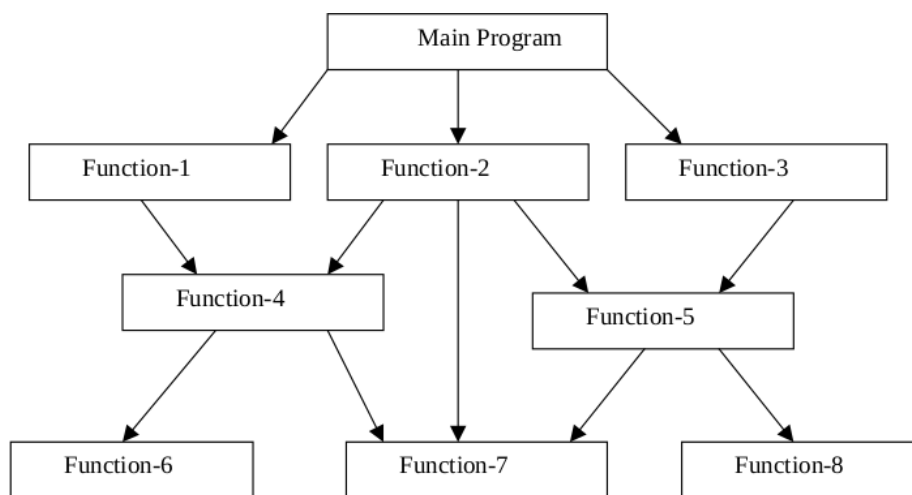
Nature is composed of different objects. Living beings can be categorized into different objects. For eg, An institute is divided into different departments such as class room, laboratory , library etc. Each one perform its own activities in association with other departments. Each department may be considered as a module and it contains classes and objects in C++ language. These classes and objects can be extended in life . Generally, objects are in terms of entities. In OOP, objects of a program interact by sending messages to each other.

Each Department has its own work. It is comfortable to accomplish goals and objectives. The activities are carried out smoothly. The burden of one particular department has to be shared among different departments with personnel. The staff in the department is controlled properly and act according to the instructions laid down by the management. The faculty performs activities related to the teaching. If the higher authority needs to know the details regarding theory, laboratory, seminar and project loads of individual faculty, then a person from the department accesses the data and provides the higher authority with the requisite information. It is also good to think that no unconnected person from another department reads the data or attempts to make any alteration that might corrupt the data.



Procedure Oriented Programming:

The Conventional programming languages such as COBOL,FORTRAN, C etc. are commonly known as procedure oriented languages.The problem in these languages is viewed as the sequence of things to be done such as reading, calculating and printing. The primary focus is on functions. A typical structure for procedural programming is shown in fig. The technique of hierarchical decomposition has been used to specify the tasks to be completed for solving a problem.

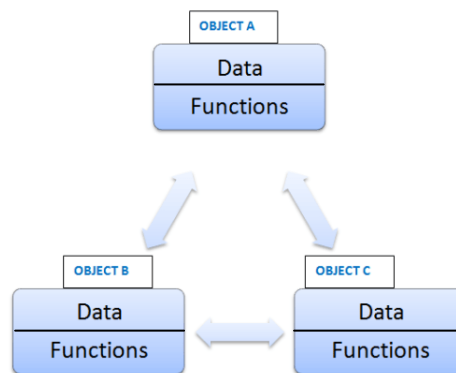


Features of Procedure-Oriented Programming :

1. Importance is given to the functions
2. Large programs are divided into smaller programs known as functions.
3. Most of the functions share global data.
4. Data move openly around the system from function to function.
5. Functions transform data from one form to another.
6. Employs top-down approach in program design.

Object Oriented Programming:

The major motivating factor in the invention of object-oriented approach is to remove some of the flaws encountered in the procedural approach. OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the function that operate on it, and protects it from accidental modification from outside function. OOP allows decomposition of a problem into a number of entities called objects and then builds data and function around these objects. The organization of data and function in object-oriented programs is shown in fig. The data of an object can be accessed only by the function associated with that object. However, function of one object can access the function of other objects.

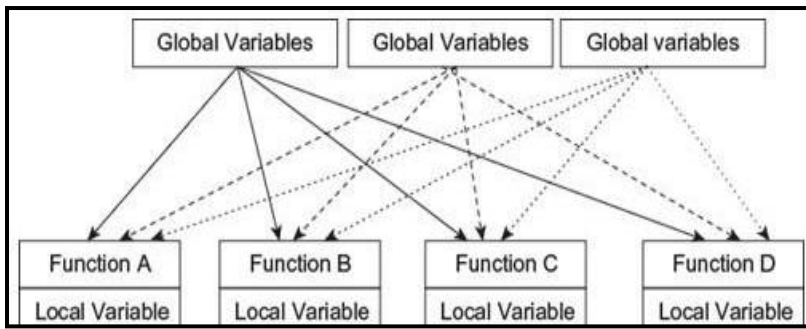


Features of Object Oriented Programming :

1. Emphasis is on data rather than procedure.
2. Programs are divided into what are known as objects.
3. Data is hidden and cannot be accessed by external function.
4. Data structures are designed such that they characterize the objects.
5. Functions that operate on the data of an object are tied together in the data structure.
6. Objects may communicate with each other through function.
7. New data and functions can be easily added whenever necessary.
8. Follows bottom up approach in program design.

Disadvantages of Conventional Programming:

The Conventional programming languages such as COBOL, FORTRAN, C etc. are commonly known as procedure oriented languages. The problem in these languages is viewed as the sequence of things to be done such as reading, calculating and printing. The primary focus is on functions. A typical structure for procedural programming is shown in fig. The technique of hierarchical decomposition has been used to specify the tasks to be completed for solving a problem.



Disadvantages:

1. Large programs are divided into smaller programs known as functions. These functions can call one another. Hence security is not provided.
2. No importance is given to security of data and importance is given to doing things.
3. Data move openly around the system from function to function.
4. Most functions accesses global data, what happens to the data is not known.

Basic Concepts (Key Concepts) of Object Oriented Programming

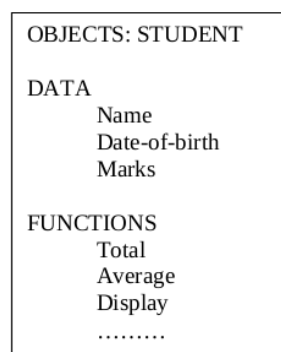
It is necessary to understand some of the concepts used extensively in object-oriented programming. These include:

1. Objects	5. Inheritance
2. Classes	6. Polymorphism
3. Abstraction	7. Dynamic binding
4. Encapsulation	8. Message passing

1) Objects

Objects are the basic run time entities of real world. They are instances of class, which have data members and uses various member functions to perform tasks. Objects may represent a person, a place, a bank account.

When a program is executed, the objects interact by sending messages to one another. For example, if “customer” and “account” are to objects in a program, then the customer object may send a message to the account object requesting for the bank balance. Each object contain data, and code to manipulate data. Objects can interact without having to know details of each other’s data or code.



2) Class:

It is similar to structures in C language. The collection of similar objects is called a class. Class binds the data and its related functions together. Class is a user-defined data type but behaves as a built-in type. For eg, fruit is a class. Mango and apple are objects.

3) Data Abstraction

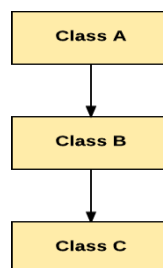
Abstraction refers to the act of representing essential features without including background details or explanations. i.e Abstraction refers to showing only the essential features of the application and hiding the details. In C++, classes provide methods to the outside world to access & use the data variables, but the variables are hidden from direct access. Abstraction is the procedure of identifying properties and methods related to a specific entity. Classes use the theory of abstraction, they are known as Abstract Data Types (ADT).

4) Encapsulation

The wrapping up of data and functions into a single unit is known as Encapsulation. i.e a class is provides Encapsulation, because it groups data and its related functions together into single unit. The data is not accessible by outside functions. The functions that are defined in class can access the data. These functions prepare interface between objects data and the program. Protecting the data from direct access by program is called data hiding.

5) Inheritance:

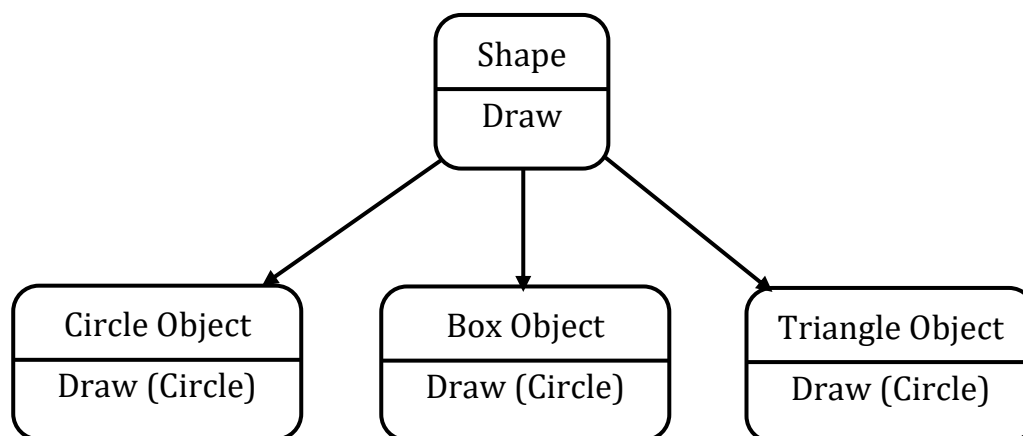
It is the process by which objects of one class acquires the properties of objects of another class. Inheritance is a way to reuse once written code again and again. The class which is inherited is called base class & the class which inherits is called derived class. So when, a derived class inherits a base class, the derived class can use all the functions which are defined in base class, hence making code reusable. The programmer can add the new properties to the existing class without changing it. This can be done by deriving a new class from the existing class. The newclass will possess the features of both the classes.



6) Polymorphism

It is the ability to take more than one form. Polymorphism makes the code more readable. It is a feature, which will create functions with same name but different arguments, which will perform differently. That is function with same name,

functioning in different. Function overloading and operator overloading are the examples of Polymorphism. Using a single function name to perform different type of task is known as function overloading. The process of making an operator to exhibit different behaviors in different instances is known as operator overloading.



7) Dynamic binding:

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run time. It is associated with polymorphism and inheritance. A function call associated with a polymorphic reference depends on the dynamic type of that reference.

Consider the procedure “draw” in the above fig. by inheritance, every object will have this procedure. Its algorithm is, however, unique to each object and so the draw procedure will be redefined in each class that defines the object. At run-time, the code matching the object under current reference will be called.

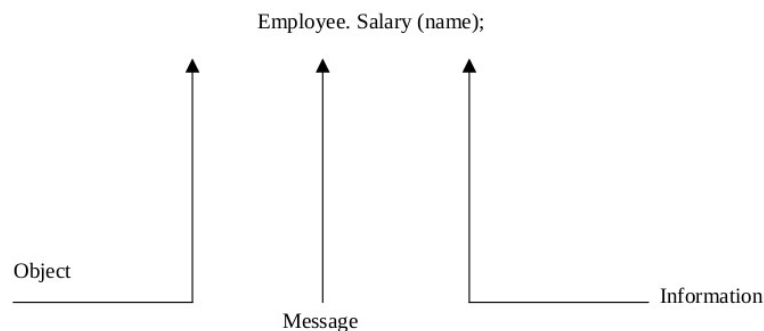
8) Message Passing:

An object-oriented program consists of a set of objects that communicate with each other. The process of programming in an object-oriented language, involves the following basic steps:

1. Creating classes that define object and their behavior,
2. Creating objects from class definitions, and
3. Establishing communication among objects.

Objects communicate with one another by sending and receiving information much the same way as people pass messages to one another. The concept of message passing makes it easier to talk about building systems that directly model or simulate their real-world counterparts. A Message for an object is a request for execution of a procedure, and therefore will invoke a function (procedure) in the receiving object that generates

the desired results. Message passing involves specifying the name of object, the name of the function (message) and the information to be sent. Example:



Applications of OOP:

1. Real-time system
2. Simulation and modeling
3. Object-oriented data bases
4. Hypertext, Hypermedia, and expertext
5. AI and expert systems
6. Neural networks and parallel programming
7. Decision support and office automation systems
8. CIM/CAM/CAD systems

Advantages of OOP:

1. It models real world well.
2. With OOP, programs are easy to understand.
3. OOP offers classes reusability. Already created classes can be reused without write it again.
4. Data hiding facilitates the programmer to design and develop safe programs.
5. With OOP, programs are easier to test, manage and maintain.
6. Objects can communicate with each other by message passing mechanism.

Disadvantage of OOP:

1. With OOP, classes tend be overly generalized.
2. The relation among classes become artificial at times.
3. The OOP programs design is tricky.
4. Also one needs to do proper planning and proper design for OOP Programming.
5. To Program with OOP, Programmer need proper skills such as design skills , programming skills, thinking in term of objects etc.

Object based Vs Object Oriented Programming Languages:

The languages should support several of the OOP concepts to claim that they are object-oriented. Depending upon the features they support, they can be classified into the following two categories:

1. Object-based programming languages, and
2. Object-oriented programming languages.

Object-based programming is the style of programming that primarily supports encapsulation and object identity. Major features that are required for object based programming are:

- Data encapsulation
- Data hiding and access mechanisms
- Automatic initialization and clear-up of objects
- Operator overloading

Languages that support programming with objects are said to be object-based programming languages. They do not support inheritance and dynamic binding. Ada is a typical object-based programming language. Object-oriented programming language incorporates all of object-based programming features along with two additional features, namely, inheritance and dynamic binding. Object-oriented programming can therefore be characterized by the following statements: Object-based features + inheritance + dynamic binding.

1.11. Structure of C++ Program

The structure of C++ program is divided into four different sections:

- (1) Header File Section
- (2) Class Declaration section
- (3) Member Function definition section
- (4) Main function section

1) Header File Section

This section contains various header files. You can include various header files in to your program using this section. Header File contains declaration and definition of various built in functions as well as object. In order to use this built in Functions or object we need to include particular header file in our program.

For example:

```
# include <iostream.h >
```

2) Class Declaration section

This section contains declaration section of Class. You can declare a class and then declare data members and member functions inside that class. You can also inherit one class from another existing class in this section.

For example:

```
class Demo
{
    int a, b;
    public:
    void input();
    void output();
};
```

3) Member Function definition section

1. This section is optional in the structure of C++ program.

2. Because you can define member functions inside the class or outside the class. If all the member functions are defined inside the class then there is no need of this section.

3. This section is used only when you want to define member function outside the class.

4. This section contains definition of the member functions that are declared inside the class.

For example:

```
void Demo::input()
{
    cout << "\n Enter value of a : ";
    cin >> a;
    cout << "\n Enter value of b : ";
    cin >> b;
}
```

4) Main function section

In this section you can create an object of the class and then using this object you can call various functions defined inside the class as per your requirement.

We can also compare the structure of C++ program with client server application. In client server application client send request to the server and server sends response to the client.

For example:

```
int main ()
{
    Demo d1;
    d1.input();
    d1.output();
    return 0;
}
```

In above C++ structure the class declaration section and member function definition section both together works as a server and main () function section works as a client.

Because in main () function section we create an object of the class and then using that object we make a call to the function declared in the class.

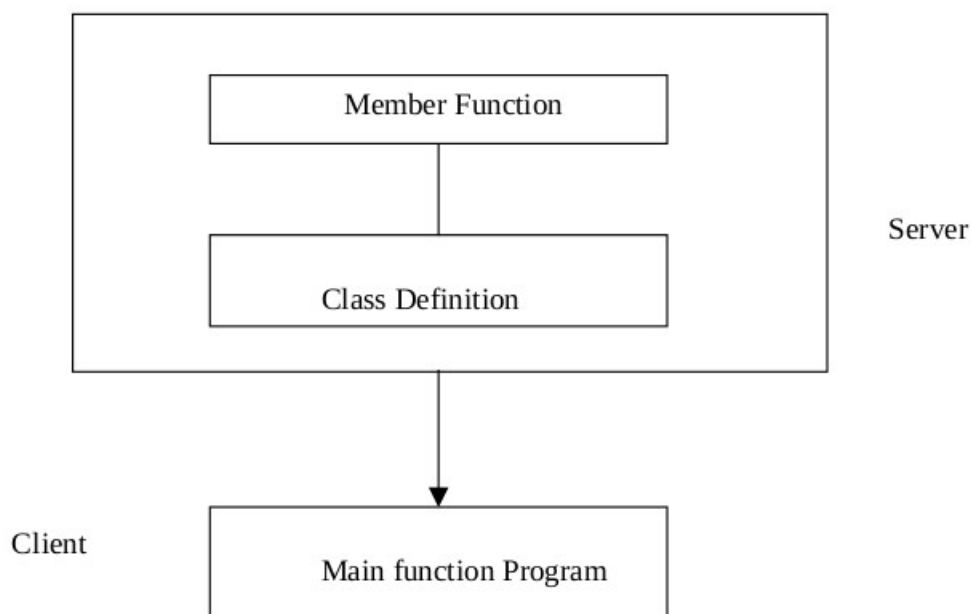
OBJECT ORIENTED LANGUAGES

There are many languages which supports object oriented programming. The following table describes OOP languages and features supported by them.

Object Oriented Languages -A Comparison

Feature	C++	Objective C	Ada	Java
Encapsulation	Yes	Yes	Yes	Yes
Inheritance	Yes	Yes	No	Yes
Multiple Inherit.	Yes	Yes	No	No
Polymorphism	Yes	Yes	Yes	Yes
Binding (Early or Late)	Both	Both	Early	Late
Concurrency	Poor	Poor	Difficult	Yes
Garbage Collection	No	Yes	No	Yes
Genericity	Yes	No	Yes	Limited
Class Libraries	Yes	Yes	Limited	Yes

Fig: Client - Server Model



Important Questions

1. List the drawbacks of conventional programming. Explain how object oriented programming overcome them. (OR)

Describe the demerits of conventional programming with suitable examples.

**Ans : Refer Topic Disadvantages of Conventional Programming (PageNo: 6)
Object Oriented Programming: (Page No : 5)**

2. Explain about polymorphism and encapsulation.

Ans : Refer Topic Polymorphism & Encapsulation (Page No: 7, 8)

3. List the similarities and differences between C and C++. (OR)
Compare the features of C programming language to that of C++programming language.

Ans : Refer Topic Difference between C and C++ (Page No : 2, 3)

4. Write about inheritance and abstraction.

Ans : Refer Topics Inheritance and Abstraction (Page No : 7)

5. Explain the key concepts of Object Oriented Programming. (OR)
Give the principles of object oriented programming.

Ans : Refer Topic Basic Concepts of OOP (Page No : 6, 7 & 8)

6. Briefly write about the evolution of C++.

Ans : Refer Topic Evolution of C++ (Page No : 2)

7. Present the structure of C++ program. Explain different elements in it.

Ans : Refer Topic Structure of C++ Program (Page No : 10 & 11)

8. How data and functions are organized in Object Oriented Program? Explain with an example.

Ans : Refer Topic Object Oriented Programming (Page No : 5)

9. How does object oriented approach differ from object based approach? Give the applications of OOP.

Ans : Refer Topic Object based Vs Object Oriented (Page No : 10)

Applications of OOP (Page No : 9)

10. What are the drawbacks of monolithic, procedural & structural programming languages?

Procedural Programming Languages: These languages code programs in such a way that the program executes statement by statement, reading and modifying a shared memory. This programming style can be closely associated with the conventional sequential processors linked to a random access memory (RAM). It contains all features of procedural programming as well as some other additional features.

Thus, we can refer to it as a programming language that executes a set of commands in order. These languages are based on the concept defined by the procedure calls. It means that statements are organized into procedures. These procedures are also known as subroutines or functions. The most important advantage of forming procedures is that the procedures are compact and they are also bounded by scope. It is written in a step by step manner, thus it is very easy to follow the smaller programs.

Examples Pascal, Fortran, COBOL

Structured Programming languages: These are based on the top down methodology in which a system is further divided into compositional subsystem. The structured programming concept was formalized in the year 1966 by Corrado Böhm and Giuseppe Jacopini. They showed the theoretical computer program design with the aid of loops, sequences and decisions.

Structured programming is not only limited to the top down approach. It employs methods using:-

1. **Top down analysis for problem solving:** It focuses on dividing the problem into sub parts and hence simplifies the problem solving.
2. **Modularization for program structure and organization:** It organizes large instructions by breaking them into separate and smaller section of modules, sub routines and subprograms.
3. **Structured code for the individual modules:** Control structures are used to determine the exact order in which the set of instructions are to be executed. Therefore, a structured code does not involve GOTO statement as it represents no certain order of execution.

Example ALGOL, Pascal, Pl/I, C, Ada

Object oriented Languages: Object oriented program languages are based on the 'objects' rather than 'actions'. This language gives more preference to data rather than logic. An object can be defined as the pairing of a piece of data with the operations that can be performed on it. It is one of the latest and powerful types. Objects are capable of storing information and they can also interact with other objects.

The three major concepts defined by a object programming language are:-

- Data encapsulation or data abstraction
- Inheritance or derivation
- Dynamic or runtime binding

Examples C++, C#, BETA, Chapel

Advantages of these languages are listed below:

Procedural	Structural	Object Oriented
<ol style="list-style-type: none"> 1. Simple, easy implementation of compilers and interpreters 2. The ability to re-use the same code at different places in the program without copying it. 3. An easier way to keep track of program flow. 4. The ability to be strongly modular or structured. 5. Needs only less memory. 	<ol style="list-style-type: none"> 1. Programs are more easily and more quickly written 2. Programs are reliable as fewer organizational and logical errors occur during the initial stages of program development. 	<ol style="list-style-type: none"> 1. Improved software development productivity due to modularity, extensibility and reusability. 2. Software Maintenance is improved 3. Reusability helps in faster development of programs, as the language comes with rich library of objects 4. Lower cost of Development 5. Higher quality of software can be ensured

Assignment Questions

1. List the drawbacks of conventional programming. Explain how object oriented programming overcome them.
2. Explain the key concepts of Object Oriented Programming.
3. How does object oriented approach differ from object based approach? Give the applications of OOP.
4. Present the structure of C++ program. Explain different elements in it.
5. List the similarities and differences between C and C++.