DEPARTMENT OF COMMERCE (CA)

OBJECT ORIENTED PROGRAMMING WITH C++ (SEMESTER-IV)

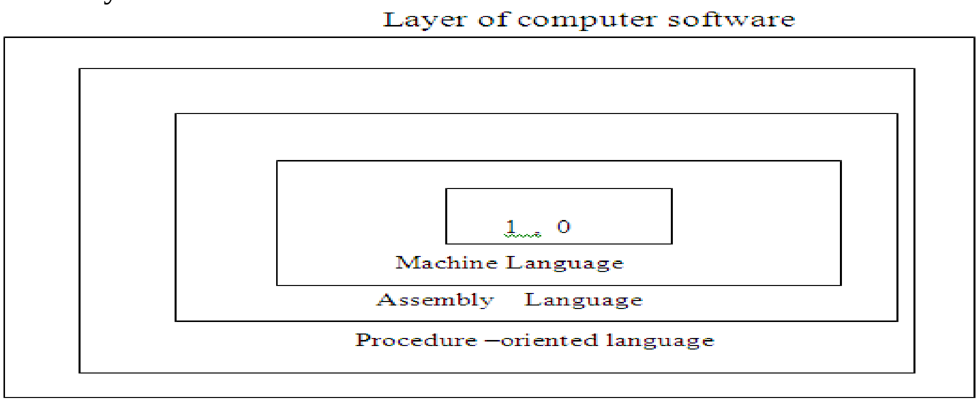
II-B.COM (CA) **SUBJECT CODE - 18BCA42C**

UNIT – I

Principles of object-oriented programming – a look at procedure and Object-oriented paradigm – Basic concepts of object Oriented programming –Benefits of OOP – Object oriented languages – application of OOP – beginning With C++ - What is C++? – Applications of C++ - C++ statements – Structure of C++ program.

Software evolution:

Software evolution has various layers. Each layer represents improvement over the previous layer.

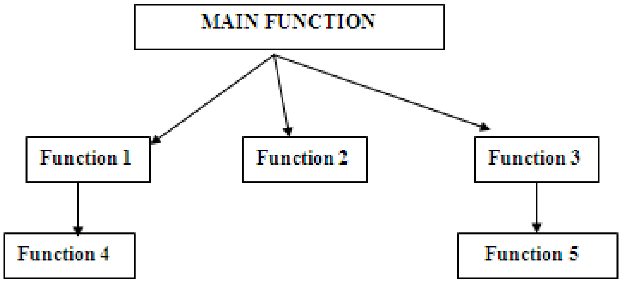


- Many programming approaches have been into play from the time of inventions of the computer.
- These techniques include modular programming, top-down programming, bottom-up programming and structured programming.
- The primary motivation in each case has been the concern to handle the increasing complexity of programs that are reliable and maintainable.
- To build today’s complex software, it is just enough to put together a sequence of programming statements and set of procedures and modules but to incorporate sound construction techniques and program structures that are easy to comprehend, implement and modify. C, structured programming was a powerful tool that enabled programmers to write moderately complex programs fairly easily.
- However, as the programs grew, even the structured approach failed to show the desired results in terms of bug-free, easy-to-maintain and reusable programs.

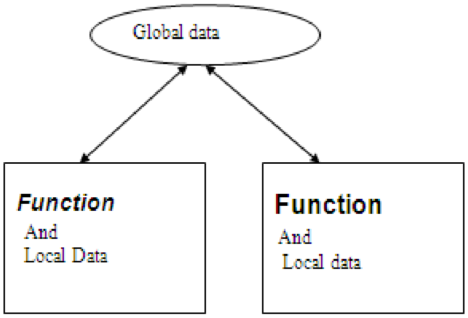
- OOP is an approach to program organization and development that attempts to eliminate some of the pitfalls of conventional programming methods by incorporating the best of structured features with several powerful new concepts.

Procedure oriented programming:

- Problem is decomposed into subproblems, each problem can b e solved by a separate function.
- Pictorially that can be represented as follows:



- Less importance is given to data , because a data can be globally accessed by any function
- The global data can be corrupted easily.
- We can not model real world problems very easily.



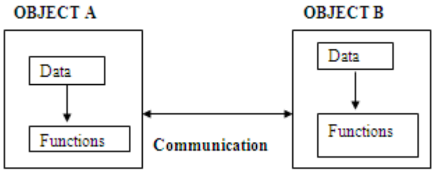
Characteristics of procedure oriented programming:

- Emphasis on algorithms.
- Large programs are divided into smaller programs known as functions.
- Most of the functions share global data.
- Data move openly among functions.
- Functions transform data from one form to another
- Top-down approach is followed.

Object – oriented programming paradigm :

- Importance is given to data , i.e., it can not move freely around the system. By this way we can protect data from accidental modifications.
- Programs are divided into objects and member functions.
- Object characters are defined by data structures.
- Functions that operate on the data are grouped together in the data structure.
- The data of the object can be accessed only by the functions associated with that object.

Organization of data and functions in OOP:



- Data is hidden (encapsulated) and can not be accessed by external functions.
- Objects may communicate with each other through functions.
- Bottom – up approach is followed.

Definition of object-oriented programming:

"Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand".

Basic Concepts of object-oriented programming:

Object:

- Object is a run-time entity
- Example: an item, a place, a bank account, person, student, etc.,
- Object is treated as real world objects.
- It consumes memory space like structure in C.
- It interacts by sending messages to one another.
- Each object contains data and code to manipulate that data.
- Object can interact without having to know details of each others data or code.

Example:

Object : STUDENT
Data : Roll number, Name, Marks etc.
Functions: Total, Average, Result etc,

Class:

- Objects contain data and code to operate on data.
- Objects are variables of type Class.
- Class is a user-defined data type , each object is associated with the data of type class.
- Example; birds peacock;

  This means the object ‘ peacock’ belongs to the class ‘birds’.

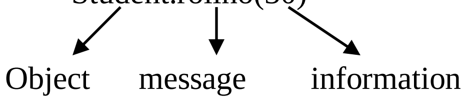
Dynamic binding(late binding):

- It is a process of linking the procedure calls(codes) to the calling point (Programs) during runtime.
- At run –time, the code matching the object under current references will be called.
-

Message passing:

- After creating objects from the class definition, we can make communications between objects with the
  help of messages or methods
- message passing means specifying the name of the object, the name of the function(message) and the information to be sent (parameters).

- Example: Student.rollno(30)



Delegation :

Two classes can be joined either by inheritance or delegation which provides Reusability of class. When objects of one class is used as data member in other class. Such composition of objects is known as Delegation. Delegation type of Relationship is called Relation.

Principles of Object- Oriented Programming Language

Data abstraction and encapsulation:

- Data and functions are kept into a single unit (class).This concept is called as encapsulation.
- Data can be accessed by the function of that class.
- The function provides the interface between the object's data and the program.
- The insulation of the data from direct access by the program is called data hiding.
- Classes are known as abstract data type (ADT) because they define list of abstract attributes such as type, size and functions to operate on these objects.
- Abstraction means defining attributes of an object without knowing background details or explanation.

Inheritance:

Object of one class posses the properities of another class.

- Example : Pet animal(Dog) has the general property of Animals and it posses its own characteristics.
- Each derived class shares common property with the class from which it is derived(base class).
- Inheritance supports the concept of reusability.
- The new derived class will have the combined features of both the classes.

Polymorphism:

- It is an ability to take more than one form.
- A function can be used to handle different operations with different type of arguments.
- Example: Consider the operation ADD, it adds two numbers. If the two operands are string , then the operation produces a third string .

Benefits of OPP:

This new technology promises greater productivity, better quality software and lesser maintenance cost. Other advantages are :

- Existing classes can be extended using inheritance that avoids redundant code.
- Data hiding helps the programmer to build secure programs and so it cannot be invaded by code in other parts of the program.
- Multiple instances of an object can co-exist without any interference.

- It is easy to partition work in a project based on objects
- Object oriented systems can be easily upgraded from small to large systems
- Communication between objects is simpler.
- Software complexities are easily manageable.

Object-oriented languages:

Object oriented programming languages are classified into two types based on the concepts they support. The two types are :

- Object-based languages
- Object-oriented languages

Object-based languages is that style of programming that supports encapsulation and object identity. The other features available in this type are Data hiding, Automatic initialization and clear up of objects, operator overloading. It also supports programming with objects. The two features that are not supported by object-based languages are dynamic binding and inheritance. E.g. Ada

Object-oriented languages incorporates in it features like data hiding, encapsulation, operator overloading, automatic initialization and clear-up of objects, along with inheritance and dynamic binding. E.g. C++, Object Pascal, Small Talk. In general, Object-oriented language=Object-based + inheritance + Dynamic binding

Applications Of Object-Oriented Programming:

The most popular applications of object-oriented programming has been in the area of user interface design such as windows. Promising areas for application of oops includes:

- Real-time systems
- Simulating and modeling
- Object-oriented databases
- Hypertext, Hypermedia and expert text
- Artificial Intelligence and expert systems
- Neural networks and parallel programming
- Decision support and office automation systems
- Computer Integrated Manufacture/Computer Aided Design and Manufacture etc

Object-oriented paradigm came from a language, matured into design and has now moved into analysis. This technology is going to change the way software engineers think, analyze, design and implement systems in future.

Introduction to C++

- It is an object oriented programming language.
- It was initially named as ‘ C with Classes’.
- C++ was developed by Bjarne Stroustrup at AT& T Bell Laboratories USA in early 1980’s.
- C++ is a superset of C.
- All C programs are also C++ programs and C program will run under C++ compiler.
- C++ additionally supports Classes, function overloading and operator overloading. These features enable us to create abstract data types, inherit properties from existing data types and support polymorphism, thus making C++ a truly object oriented language.

Applications of C++:

Since C++ allows us to create hierarchy-related objects, special object-oriented libraries can be build that can be used later by many programmers. C++ programs are easily maintainable and expandable.

- C++ can be used to develop databases, compilers, editors and any complex real – life application systems.
- C++ is a versatile language for handling very large programs. It is suitable for virtually any programming task including development of editors, compilers, databases, communication systems and any complex real-life application systems.
- It can also be used to develop hierarchy – related real-world problems.

C++ Statements

C++ is a collection of functions. Execution begins at main(). Every C++ program must have a main(). C++ is a free-form language. Like C, C++ terminates with semicolon.

C++ introduces a new comment symbol //(double slash). Comments start with a double slash symbol and terminate at the end of the line. A comment may start anywhere in the line and whatever follows till the end of the line is ignored. There is no closing symbol.

For e.g.

```
//This is an example of
//C++ program to illustrate
//some of its features.
```

The C comment symbols /*, */ are still valid and more suitable for multilane comments. For e.g.

```
/*This is an example of
C++ program to illustrate
some of its features.*/
```

Output Operator:

C++ introduces two new C++ features, cout and <<. The identifier cout is a predefined object that represents the standard output stream in C++. The standard output stream represents the screen.

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

For e.g.

```
cout<< “C++ is  better C.””;
cout<< string;
```

Input Operator

The identifier cin is a predefined object that represents the standard input stream in C++. The standard input stream represents the keyboard.

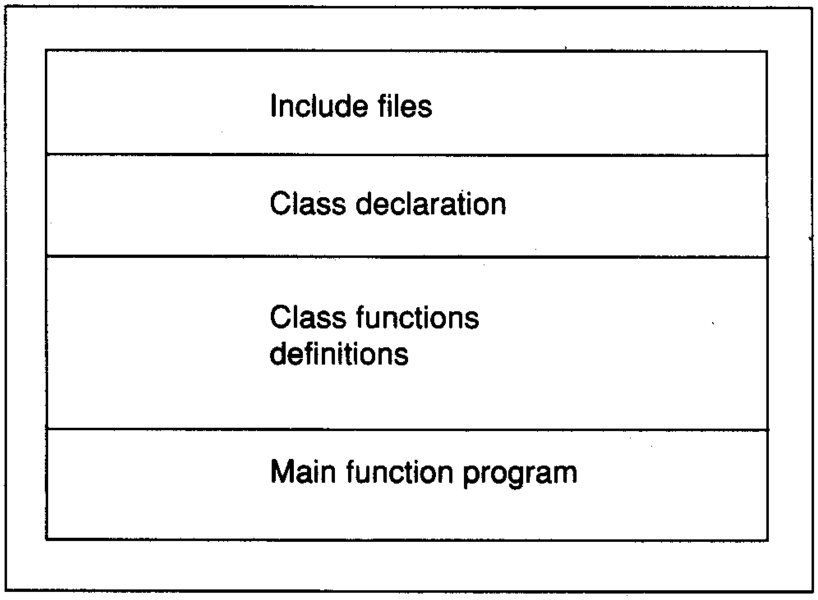
The operator >> is called the extraction or get from operator. It extracts(or takes) the value from the keyboard and assigns it to variable on the its right.

For e.g.

```
cin>> string;
```

Structure of C++ Program

A typical C++ program would contain four sections as shown in the diagram. These sections may be placed in separate code files and then compiled independently or jointly.



- It is a common practice to organize a program into three separate files. The class declarations are placed in a header file and the definitions of member functions go into another file.
- This approach enables the programmer to separate the abstract specification of the interface (class definition) from the implementation details (member functions definition). Finally, the main program that uses the class is placed in a third file which "includes" the previous two files as well as any other files required.
- This approach is based on the concept of client-server model. The class definitions including the member functions constitute the server that provides services to the main program known as client.
- The client uses the server through the public Interface of the class.

REFERENCE:

1. E. Balaguruswamy , " Object oriented programming with C++", TataMcGraw Hill publishing company Limited, 1998.
2. K.R, Venugopal, Rajkumar, T. Ravishankar, "Mastering C++", tata mc graw – Hill publishing company Limited, 1998.
3. D. Ravichandran, Programming with C++", Tata McGraw – Hill published Company Limited.

Prepared By Dr.N.Shanmugavadivu