C++

Chapter one:

- As we know c++ is the object oriented programming, it introduces the concept of object and classes.
- Object oriented programming was much easier than procedure oriented programming.
- It is as well a high level programming language.
- Data and functions that operate on data are combined into single unit called object.
- We cannot access the data easily in object directly.

We can see that data, member function and object have close relation with each other.

The errors obtained through procedure oriented programming approach gave a lesson for development of object oriented programming.

Procedure/Structure for an object oriented programming

- Based on the concept of procedure (function) call.
- Hardware support is provided by the computer processor through the stack register and instructions for calling procedure and returning from them.
- As the task is broken down into functions, it is convenient for making piece of code written by different people or group.
- Smaller programs have no others organizing principle.
- Divisions into functions if program grows larger and more complex.
- Each function has the clearly defined purpose
- A number of functions can be grouped together In a larger entity called module.

Characteristics of the procedure oriented programming

- Large program is broken into small function reducing the size of program.
- Uses function call technique to use the same function at different places in a single program.
- Data moves openly around the system from function to function
- Near to machine level language and hence fast compared to object oriented programming.

- Considered more powerful than object oriented programing.
- Focuses on function rather than data.
- Program design follows top to down methodology.

Limitations of procedure oriented programming

- Difficult in design
- Difficult to create new data types
- No tools available for abstraction and modularization.
- It focuses more on function rather than data.

Characteristics of object oriented programming

- One of the nearest and most powerful paradigms.
- Data hiding
- Models real world problem
- Gives emphasizes on data rather than procedures.
- Programs divided into objects
- Follows bottom to top methodology
- Function and data are tied to single unit called as object
- Data are accessed through visible functions.

Features of object oriented programming

- **Object**

  - Object can interact without having to know the details of each other's data or code.
  - Objects are the runtime entities in an object oriented programming
  - The entities are represented as objects in program
  - Data and functions are tied together in object.

  Example: vehicle

  Data: name of company

  >> type of engine, mileage, price etc. etc.

So the function for its can be written as

Total price ()

Vat ()

Display ()

- Class

  - User defined data type for declaration of objects
  - Any number of objects can be created if a class is defined at once.
  - Class is a template for specialization of data and their operation
  - Class is the implementation of abstract data type.

Class Employee:

Data: Name Id
Function members: Sort name() Total no_of_employee() Annual_income()

Abstraction:

Act of representation of important features without including the internal details of how any object performs its task.

- Provides the interface for using the service provided by the objects.
- Complexity of the program can be managed by abstraction.
- Classes are the concepts of abstraction.

Encapsulation:

- Mechanism of combining of data and function together into single unit
- Encapsulation provides safety to the data from being accessed by other code
- The data in the class can be marked as private or public.
- Only the member class code can access the private function and data.

Inheritance:

- The idea of class leads to the idea of inheritance.
- The concept of inheritance is analogous to using function to simplify traditional procedural language.
- Thus the mechanism of creating new class based on existing class is inheritance.

Reuseability:

- Once the class has been written and debugged, it can be distributed to other programs for use in their own programs.
- In oop concept of inheritance provides an important extension to idea of reuseablilty.
- A programmer can take and existing class and add additional features by deriving a new class from the existing one.
- The idea saves the time and effort of the programmer.

Advantages of OOP

- Provides a clear modular structure.
- Easy to maintain and upgrade and hence supports foster development.
- Message passing provides simpler interface for operation.
- Models the real world program.
- Divisions of program into object makes software development easy.

Disadvantages of OOP

- Requires more time in compilation.
- Difficulty in tracing and debugging error in complex program.
- Requires background knowledge of programming for new comers.
- The size of programmes developed with OOP is larger than the procedural approach.

DIFFERENCES BETWEEN OBJECT ORIENTED PROGRAMMING AND PROCEDURAL ORIENTED PROGRAMMING

Procedural oriented programming	**Object oriented programming**
➢ Emphasis is given to procedure. ➢ Program delivered into function ➢ Data cannot be hidden ➢ Movement of data is form function to function ➢ Difficult in reuseablilty of code	➢ Emphasis is given to data ➢ Program divided into objects ➢ Data can be hidden ➢ Data and function are tied together ➢ Easy in reuseability of code ➢ Models real world problem better than pop

➢ Follows top down methodology ➢ Does not model real world problem effectively ➢ Example: c, fortan	➢ Follows bottom to top methodology. ➢ Example: c++,java

Data abstraction

Abstraction refers to the act of representing essential features without including background details or explanations. Data abstraction is a programming technique that relies on the separation of interface and implementation.