Chapter 06:

Basic concepts of OOP'S

One mark questions:

1. What is the fundamental idea of object oriented programming?
   Object oriented programming is a programming paradigm that uses objects to design applications and computer programs.

2. What is an object?
   An object is referred as an instance of the class.

3. Define the term class.
   Class is a blueprint from which objects are created.

4. Define the term data abstraction.
   Abstraction refers to the process of representing essential features without including background details or explanations.

5. What is encapsulation?
   Data encapsulation combines data and functions into a single unit called class.

6. What is meant by function overloading?
   Function overloading means two or more functions have same name ,but differ in the number of arguments or data type of arguments.

7. Define polymorphism
   The ability of an operator and function to take multiple forms is known as polymorphism.

8. What is inheritance?
   The process of forming a new class from an existing class is known as Inheritance.

9. What is a base class?
   A base class is a class in Object-Oriented Programming language, from which other classes are derived. ... A base class is also called parent class or super class.

10. What is a derived class?
    A class that is created from an existing class. The derived class inherits all members and member functions of a base class.

11. How are base class and derived class related?
    The derived class acquires the properties of the base class.

12. Define the term data hiding.
    Data hiding is a process of combining data and functions into a single unit.

Two marks questions:

1. What is inheritance?

   It is the process by which objects of one class acquires the properties of another class (OR) The process of forming a new class from n existing class is known as inheritance.

2. Explain or define class.

   A class is a way of grouping objects having similar characteristics.

3. Explain overloading.

   Overloading are those which has multiple functions, same name and different types or amount of parameters.

4. Write any two advantages of OOP's.
   - Inheritance supports code reusability.
   - Data hiding helps to secure programs.

5. Write any two drawbacks of OOP's.
   - Special kills are required.
   - No set standards.

6. Write any two applications of OOP's.
   - CAD/CAM software.
   - Simulation and modelling.

7. Mention the types of inheritance.
   - Single level inheritance.
   - Multi level inheritance.
   - Multiple level inheritance.
   - Hierarchical inheritance.
   - Hybrid inheritance.

Five marks questions:

1. Write the difference between procedural oriented programming and object oriented programming.

Sl. no	Procedure oriented programming	Object oriented programming
1.	Variables	Objects
2.	User-defined data types	Classes
3.	Function call	Message passing
4.	Functions	Methods
5.	Data is not secured.	Data is secured using data abstraction.

2. Explain the advantages of OOP’s.
    i. It is used to model real world entities.
    ii. Inheritance supports code reusability.
    iii. Data hiding helps to secure programs.
    iv. Multiple instances of an object can be created.
    v. Software’s can be easily maintained.
    vi. It can be easily upgraded to larger systems.

3. Explain the disadvantages of OOP’s.
    i. Special skills are required.
    ii. No set standards.
    iii. Proper planning and design is required before using OOP’s.
    iv. It uses a tricky method of programming.

4. Write the real world application of OOP’s.
    i. Computer graphic applications.
    ii. CAD/CAM software’s.
    iii. Object-oriented database.
    iv. User interface design such as Windows.
    v. Real time systems.
    vi. Simulation and Modelling.
    vii. Artificial intelligence and expert systems.

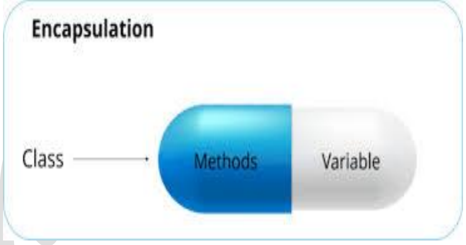
5. Explain the characteristics of OOP’s.
    1. **Class**: The building block of C++ that leads to Object-Oriented programming is a Class. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object. A Class is a user-defined data-type which has data members and member functions. Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions define the properties and behavior of the objects in a Class.

    2. **Object:** An Object is an identifiable entity with some characteristics and behaviour. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. When a program is executed the objects interact by sending messages to one another.

        Example for class and object:

```
class person
{
    char name[20];
    int id;
public:
    void getdetails(){ }
};
int main()
{
   person p1; // p1 is a object
}
```

3. **Encapsulation**: In normal terms, Encapsulation is defined as wrapping up of data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them. Encapsulation also leads to *data abstraction or hiding*. As using encapsulation also hides the data. In the above example, the data of any of the section like sales, finance or accounts are hidden from any other section.



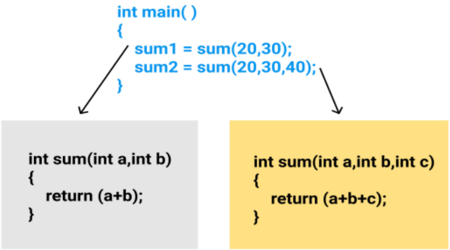
4. **Abstraction**: Data abstraction is one of the most essential and important features of object-oriented programming in C++. Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

5. **Polymorphism:** The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

   An operation may exhibit different behaviours in different instances. The behaviour depends upon the types of data used in the operation.

   C++ supports operator overloading and function overloading.

   - ***Operator Overloading*:** The process of making an operator to exhibit different behaviours in different instances is known as operator overloading.
   - ***Function Overloading*:** Function overloading is using a single function name to perform different types of tasks.

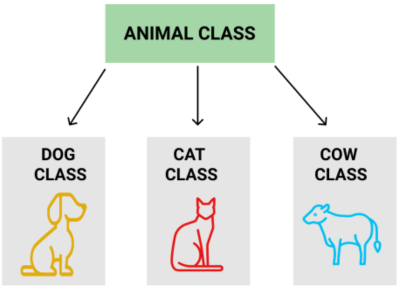


```
int main( )
{
   sum1 = sum(20,30);
   sum2 = sum(20,30,40);
}
```

```
int sum(int a,int b)
{
   return (a+b);
}
```

```
int sum(int a,int b,int c)
{
   return (a+b+c);
}
```

6. **Inheritance**: The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming.
   Example : Dog, Cat, Cow can be
   derived class of
   Animal base class.



7. **Dynamic Binding:** In dynamic binding, the code to be executed in response to function call is decided at runtime.

8. **Message Passing:** Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

Assignment :

1. Explain array and its declaration
2. Explain with example how to access the elements of 1D array.
3. Explain with example how to acces elements of 2D array.
4. Write a program to read and display 1D array.
5. Write a program to arrange elements of 1D array.
6. Write a program to find the location of an element in array.