CEAT202:

OOPS

CH:1 :-

#OOPS in C++ :-

It is a type of programming which uses object & classes its functioning. It is based on real world entities like inheritance, polymorphism, data hiding etc.

Basic Concepts of OOPs :- Encapsulation, Inheritance, Abstraction, Class, Objects, Polymorphism.

- **Class :-** It is basic str. block on ~~obj. orie~~ OOP. It is a d-type that has its own members i.e :- data members & member func^n. Members of class can accessed by making instance of class.

  Properties :-
  - It is a user defined d-type
  - Data members are variables of the class
  - Member func^n are the methods that are used to manipulate data members.

  Data members define the properties of the class where as member func^n defines the behaviour on class.

  Syntax :-

```
class class_name {
    data_type data_name;
    return_type method_name (parameters);
}
```

- **Object :-** It is instance of class. It is entity that is created to allocate memory. It is entity that access data member & member func^n of class in OOP

  Syntax :-

```
Class_name object_name;
```

Example :-

```
#include <iostream>
using namespace std;
class demo                    // class name
{
  public:                     // ~~variable~~ members
  int f_num, s_num;
  sum(int a, int b)           // member func^n
  {
    cout<<a+b;
  }
};                            // — class created
main();
{
  demo d1;                    // object
  d1.sum(d1.f_num=10, d1.s_num = 39);
  return 0;
}
```

~~Output~~ :- 49

- **Encapsulation**:- is binding together the data & related func^n that can manipulate the data. It is the concept of wrapping together of data & info. in a single unit.
- **Polymorphism**:- It is the ability of OOP to do some work using multiple forms
- **Inheritance**:- The capability of a class to derive properties & characteristics from another class is c inheritance
  The class that derives properties from other class is known as child class or subclass & the class from which the properties are inherited is base class or parent class or Super Class.
  Child class also known as derived class.

Types of inheritance :-
- Single inheritance
- Multiple inheritance
- Hierarchical inheritance
- Multi Level inheritance
- Hybrid inheritance

- **Abstraction**:- It is the concept of hiding data & showing only relevant data to the final user.
  There are 2 ways which can accomplish data abstraction:-
  - Using Classes
  - Using header files.

OOPs Languages :- Java (from sun microsystem), JS (Open source), Python, C++, Visual Basic.Net, Ruby, Scala, PHP

⇒ We can give members of the class as public, private or protected

Procedural Programming:-

It is a programming paradigm built around the idea that programs are sequences of instr. to be executed.
It focuses heavily on splitting up programs into named sets of instr. called procedures, analogous to func^n.
The procedural code is the one that directly instructs a device on how to finish a task in Logical steps
This paradigm uses a linear top-down approach & treats & data & procedures as 2 diff. entities.

Features of Procedural Programming:-
- Pre defined func^n
- Local variable
- Global variable
- Modularity
- Passing Parameters

Advantages of Procedural Programming :-

- It is excellent for general-purpose programming.
- The coded simplicity along with ease of implementation of compilers & interpreters
- A large variety of books & org course material available on tested algo., making it easier to learn along with the way

e.g:- BASIC (Beginners All Purpose Symbolic Instr. Code)
FORTRAN (formula Translation)
COBOL (Common Busines Oriented Language)
PASCAL, C

OOPs :-

It is a fascinating programming model that uses the concept of obj. It organizes S/W design around obj. or data rather than logic & func^n. As a result, developers who use OOP aim to manipulate the obj. rather than the program logic.

Advantages:

- Re-usability
- Data Redundancy
- Code Maintenance
- Security (Abstraction)
- Easy troubleshooting

e.g:- JAVA, PYTHON, C++, C#, JS, PHP, RUBY

Difference B/W Procedural & OOPs :-

Parameter	Procedural Programming	OOPs
Security	Does not offer any method of hiding data. Thus, less secure when compared to OOPs	Hiding data is possible due to abstraction. Thus, more secure than the Procedural Programming
Division of Program	Divides the program into small programs & refer to them as func	Divides the program into small parts & refer to them as obj.
Approach	Follows a Top-down approach	Follows a Bottom Up approach
Orientation	Str./Procedure oriented	Object oriented.

⇒ Benefits of OOP :-

is easier with OOP

- Troubleshooting
- Productivity
- Code flexibility
- Solving problems
- Code maintainance
- Classes & Obj.
- Inheritance
- Polymorphism