What is OOP?

Object Oriented Programming is a methodology or paradigm to design a program using classes and objects.

1. Data Abstraction
2. Data Encapsulation
3. Inheritance
4. Polymorphism

What are classes?

Class is a user-defined datatype.

Blueprint

```
class Player {
    int health;
    int score;
    String name;
}
```

What are Objects?

Object is an instance of the class.

```
class Player {
    int health;
    int score;
    String name;
}

Player harsh;
```

Why Classes?

Classes enables us to apply OOP concepts.

1. **Data Abstraction**
2. **Data Encapsulation**
3. **Inheritance**
4. **Polymorphism**

Access modifiers:

private
protected
public

← Inheritance

Array approach Vs Class

Class variables
Declaration, Initialization and Accessing

→ Data members   procte

→ member functions

          ⌐ class   ke andar

              vaale function.

Access specifiers

Private, Public and Protected

Getters and setters

Ques : Create a class 'book' with name, price and number of pages as its attributes. The class should contain following member functions :

- **countBooks(int price)** : This function will return count of all the books that have a price less than the given price.

- **isBookPresent(string title)** : This will return a boolean value indicating whether any book with the given title exists or not.

Ques : Find the error

```
class emp{
    int ecode;
     class emp e;
}
```

Passing class objects to functions

Array of Class objects'

Why?

Array of Class objects'

Declaration and Access

Ques : **Create a class "cricketer" that contains name of cricketer, his age, number of test matches that he has played and the average runs that he has scored in each test match. Create an array of data type "cricketer" to hold records of 20 such cricketers and then write a program to read these records**

Constructors

Types of Constructors

- **Default constructor**
- **Parameterized constructor**
- **Copy constructor**

Default constructor

Parameterized constructor

Destructor