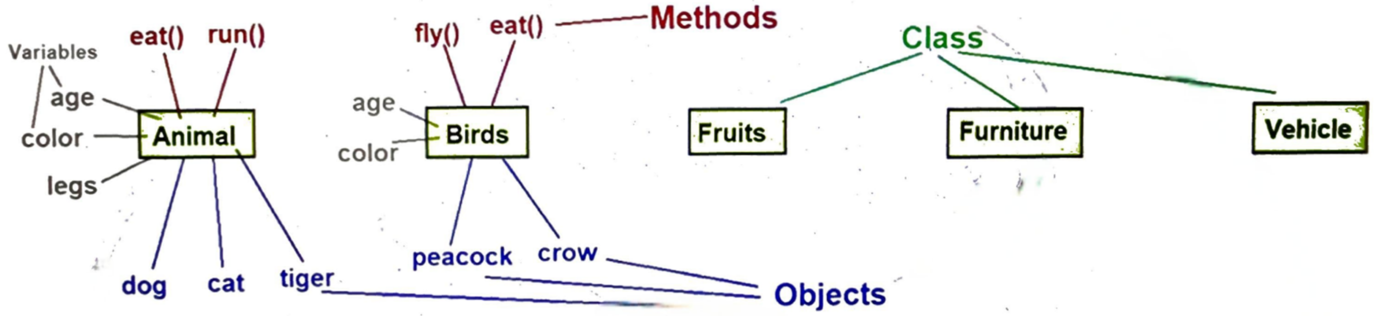
=> OOPs(Object-Oriented Programming System) :Programming Paradigm
->In OOP program is divided into parts i.e. Objects

=> Programming Paradigm :-

-> Programming paradigm is a way or an approch to solve any problem or to achieve any task using any programming languages

-> OOP is the programming paradigm based on the concept of Objects which contains the data(fields or variables) and methods
-> It is the most popular programming paradigm used by the programmers
-> For examples : Java, Python, C++ etc
-> Features of OOP :-
1. Class, Objects & Methods
2. Inheritance
3. Polymorphism
4. Encapsulation
5. Abstraction

⇨ Real World Example of Class, Methods & Objects



=> Class :-

-> A class is a user defined blueprint or prototype which is used to create an object

-> Class is a logical entity or say its not a real world entity or class is not physical

-> Real world example :- Animal, Birds, Vehicle, Fruits etc

-> Class represents the set of properties or methods that are common to all the objects of one type

-> Syntax :

```
access-modifiers class ClassName extends ParentClassName implements InterfaceName
{
        //variables
        //constructors
        //methods
        //nested class, interfaces
}
```

-> Simple syntax :

```
access-modifiers class ClassName
{
      //variables
      //methods
}
```

-> Simple class

```
class Animal
{
    int age=10;
    String color=black;
}
```

=> Methods :-

-> A set of codes which perform a particular task

-> Syntax :

```
access-modifiers return-type methodName(list of parameters) throws ExceptionClassName, -,-
{
         //statements
}
```

-> Simple Syntax :-

```
return-type methodName(list of parameters)
{
     //statements
}
```

-> Example :-

```
        void eat()                                              //method declaration
        {
                                                              //method defination (body)
           System.out.println("im eating");
        }
```

=> Objects :-

-> Object is an instance of class
-> Object is physical entity or object is real world entity

-> Syntax :
1. Creation of an object
ClassName object_name(ref_variable_name) = new ClassName();
-> Animal regun = new Animal();

2. Calling variables or methods from object
object_name.variable_name; -> regun.age;
object_name.methodName(); -> regun.eat();

=> Points to remember :-
-> We can only use public or default accessmodifiers but not private or protected with outer
class.
-> For inner class we can use all accessmodifiers i.e. public, proctedted, default and private