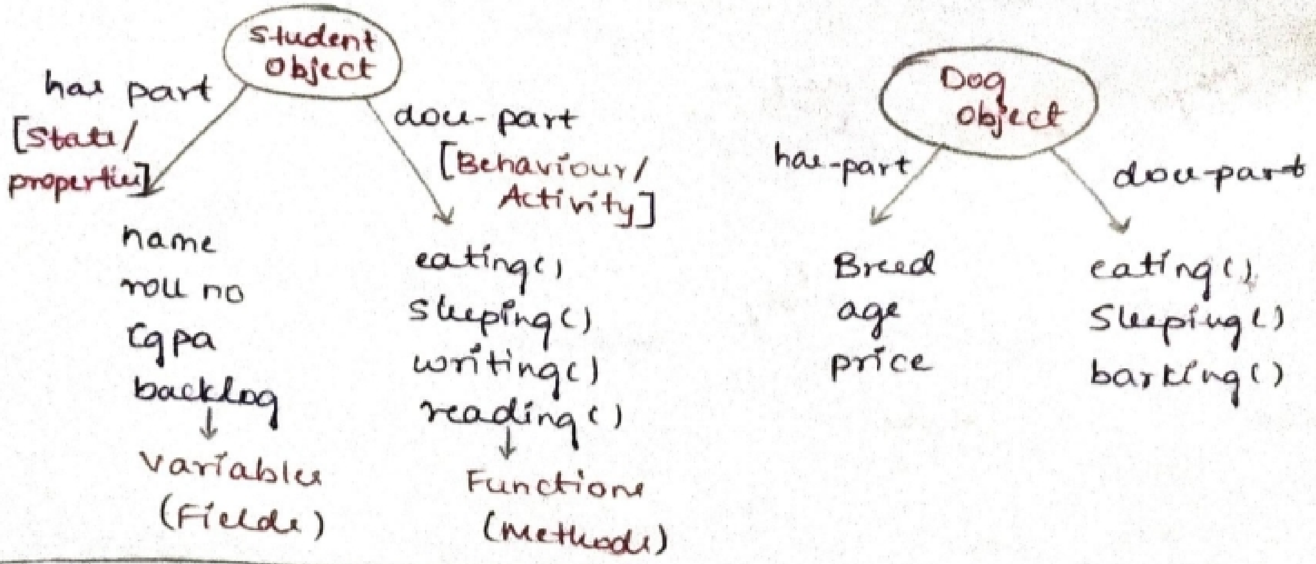


**Fields (variables)**

**Syntax:**

```
   ②            ①
data_type  variable_name ;
```

**Eg:** int rollno;

**Methods (Functions)**

**Syntax:**

```
                                        * Optional
   ④                 ①                   ②
return_type method_name (paramters)
   [output]                        [input]
  {
         ③
     // Body of the Method

  }
```

**Eg:**

```
void eating( )
   {
      //Activity.
   }
```

**Syntax**

①. Name
* ②. Input
③. Activity.
④ Output.

## printing output :

C → printf("Hello, world!");

C++ → cout << "Hello, world!" ;

C# → Console.writeline ("Hello, world!");

JavaScript → Console.log ("Hello, world!");

python → print ("Hello, world!") ;

Java → Sytem.out.printen ("Hello, world "!");
Sytem.out.print ("Hello, world! ");
Sytem.out.printf(" Hello, world! ");

## Reading output:

C → scanf("%x", a);

C++ → cin >> a ;

C# → a = Console.Readline( );

Javascript → a = prompt( );

Python → a = input ( ) ;

Java → Scanner scan = new Scanner (Sytem.in);
a = scan.next();

$$\begin{tikzcd}
 & \text{Student Object} \arrow[dl, "\text{has-part}"'] \arrow[dr, "\text{Does-part}"] & \\
\text{name, roll\_no, cgpa, backlogs} & & \text{eating(), sleeping(), writing(), reading()}
\end{tikzcd}$$

name
roll_no
cgpa
backlogs

eating()
sleeping()
writing()
reading()

State / Properties — Fields [Variables]

Behaviour / Activity. — Methods [Functions]

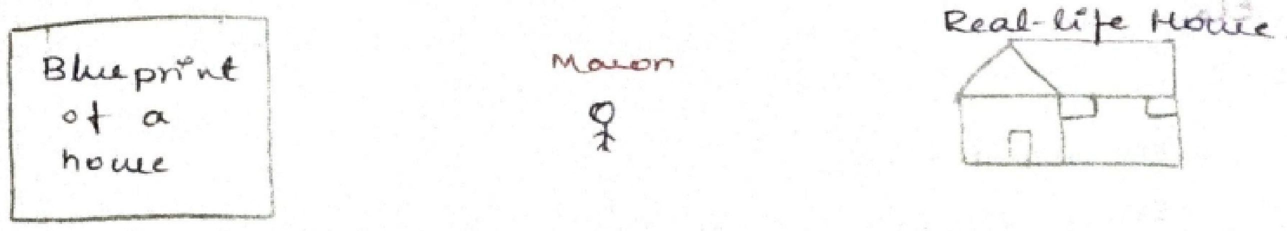
class student

has-part

```
{
   String name;
   int roll_no;
   float cgpa;
   Boolean backlogs;
```

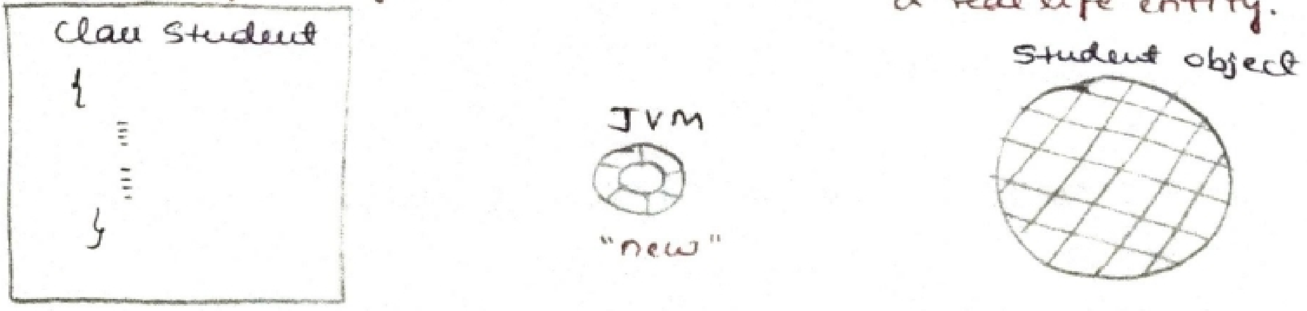
Does-part

```
   void eating( )
    {
     Sytem.out.println("student is eating...");
    }
   void sleeping()
    {
     Sytem.out.println("student is sleeping...");
    }
   void writing()
    {
     Sytem.out.println("Student is writing....");
    }
   void reading()
    {
     Sytem.out.println("Student is reading...");
    }
}
```



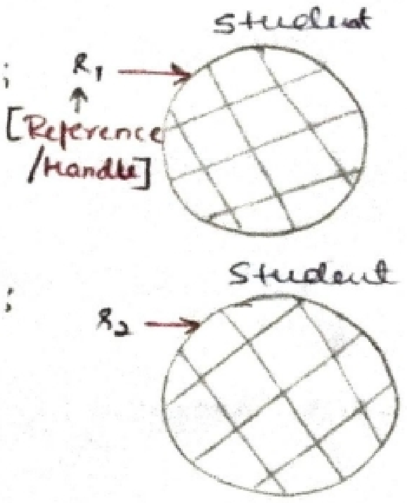
"A class is a blueprint of an object"

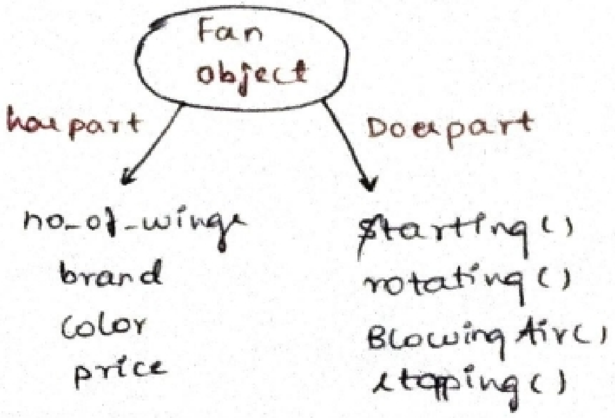
"An object is a real life entity".



```
Student  r1 = new  Student();
    r1. writing();
    r1. writing();

Student  r2 = new  Student();
    r2. eating();
```





| $f_1$ | $f_2$ | $f_3$ |
|---|---|---|
| starting()<br>stopping() | starting()<br>rotating() | starting()<br>rotating()<br>BlowingAir()<br>stopping() |

```
class Fan
{
          int  no_of_wings;
has-  ←   String  brand;
part      String  color;
          float  price;

           void  starting()
            {
              System.out.println("Fan is starting...");
            }

           void  rotating()
            {
              System.out.println("Fan is rotating...");
            }
Does- ←
part       void  blowingAir()
            {
              System.out.println("Fan is blowingAir...");
            }
           void  stopping()
            {
              System.out.println("Fan is stopping...");
            }
    }

    Fan  f1 = new Fan();
         f1.starting();
         f1.stopping();

    Fan  f2 = new Fan();
         f2.starting();
         f2.rotating();

    Fan  f3 = new Fan();
         f3.starting();
         f3.rotating();
         f3.blowingAir();
         f3.stopping();
```