DEFINING A CLASS :

A class is the basic element of object oriented programming.
A class defines the shape and behavior of an object.
A class is a template for multiple object with similar feature.

Any concept represented in a class is encapsulated in a class. When an application is written, classes of objects are defined.

To create a class a keyword ‘class’ followed by curly brackets is used. As shown below

```
class {
}
```

DEFINING CLASS ATTRIBUTES :

A class called Professor includes all its features serving as template for the concepts. Each property treated as an attribute of that class.

In Java attributes are declared inside the class.

For example, the professor’s name, professors mobile are its attributes. The definition of the class ‘Professor’ would be:

```
class Professor {
String professorName;
String professorAddres;
int mobile;
```

}

DEFINING CLASS METHODS:

Apart from defining an object's structure, a class also defines its functional interface, known as methods.

To create a method, syntax is as shown below

```
class {
return_type method_name(){
}
}
```

In Java each method requires a class.

For example, the class 'Professor' can have a method (showProfessorName) that display the name of the professor,

```
class Professor {
String professorName;
String professorAddress;
int mobile;

void showProfessorName(){
System.out.println("Name of the professor is" +name);
}
}
```

DEFINING CLASS OBJECTS:

Once a class ‘Professor’ is defined, instance (object) of that class can be created and each instance can have different attributes. When the program runs, instances of the class are created and discarded as when required.

An instance can be created in a method of class only.

The terms instance and object are used interchangeably.

For example,

```
class Professor {
String professorName;
String professorAddress;
int mobile;

void showProfessorName(){
System.out.println(“Name of the professor is” +professorName);
}

void showProfessorAddress(){
System.out.println(“Name of the professor is” +professorAddress);
Professor p = new Professor(); // Object p of class ‘Professor’
p.showProfessorName(); // p called method
System.out.println(“mobile is” +p.mobile); // p called attribute
}
}
```

Related Posts:

1. Can Java have same name variable
2. Types of variables in Java programming
3. JAVA and its Support Systems
4. JAVA environment
5. JAVA program structure
6. Tokens
7. Java statements
8. Java virtual machine
9. C++ Versus JAVA
10. Constants and Variables in Java
11. Data types JAVA
12. Constructor in JAVA
13. Array in Java
14. Applet
15. Applets Vs Applications
16. Writing applets
17. Applets life cycle
18. Creating an Executable Applet
19. Graphics in Applet
20. Applet image display
21. Applet digital clock
22. Applet mouse event handling
23. JDBC
24. Execute an SQL Statement
25. Process the result
26. CLOSE THE DATABASE CONNECTION

27. File handling
28. Define a class to declare an integer array of size n and accept the elements into the array.
29. Define a class to declare an array of size 20 of the double datatype, accept the elements into the array and perform the following: Calculate and print the sum of all the elements.
30. Java program for String, to uppercase, to equal, length of string
31. Write a Java program for Buble sort.
32. Write a Java program String to uppercase and count words startig with ‘A’
33. How to set path in Java
34. Understanding public static void main (String args[]){ } in Java
35. Difference between static and non static methods in Java