Class The Basis
Of All Computations

Two Programming Styles

Programs can be written in two programming styles:
Procedure Oriented Programming and Object Oriented Programming.

Procedure Oriented Programming

Procedure Oriented Programming focuses on writing instructions that manipulate data. In POP programs are written using smaller, reusable procedures. Procedures are also known as functions. Examples of procedural programming languages include C and Pascal.

A **Procedure Oriented Program** in **C** to compute area of two rectangles

```
#include <stdio.h>
void findarea()
{
   float length, breadth, area;
   printf("Enter length of the rectangle: ");
   scanf("%f", &length);
   printf("Enter breadth of the rectangle: ");
   scanf("%f", & breadth);
   area = length * breadth;
   printf("The area of the rectangle is: %.2f", area);
}
void main()
{
   findarea();
   findarea();
}
```

Procedure / Function

Procedure calls / Function calls

Procedure

Description

1. Two set of instructions are named as **main**() and **findarea()**. The **main()** function executes at first.
2. The method calls executes the method definition two times.

Object Oriented Programming

It organizes instructions into classes, variables and methods. Programs are done using objects. Common object-oriented languages includes Java, C++, and Python.

An **Object Oriented Program** in **Java** to ompute area of two rectangle objects

```
import java.util.*;
public class Rectangle
{
    double l,b;
    public void findArea()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter length and breadth");
        l = sc.nextDouble();
        b = sc.nextDouble();
        double a = l*b;
        System.out.println("Area = "+a);
    }
    public static void main()
    {
        Rectangle rect1 = new Rectangle();
        Rectangle rect2 = new Rectangle();
        rect1.findArea();
        rect2.findArea();
    }
}
```

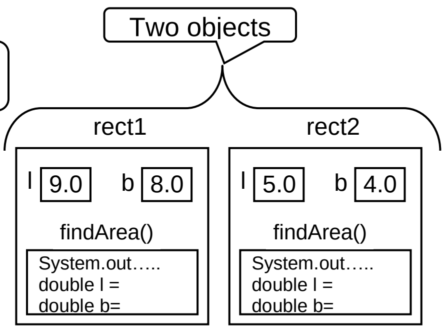
Characteristics / instance variables. These are declared in class level.

The method is **non-static**.

Behaviour / member method.

Object creation statements
The **rect1** and **rect2** are objects

message /
member method calls



Description

1. When objects are created two groups of memory cells are formed named **rect1** and **rect2**. Each group contains individual copies of instance variables and member method.
2. Each **member method** is called indiviually using objects.

Definitions, Examples And Program Terms

Class

A class **represents** a set of **objects** that share **common characteristics** and **behaviour**.

E.g.: **Rectangle** is a **class** and its **objects** are different rectangle shapes. Their **common characteristics** are **length** and **breadth**. One of the **behaviour** is **finding area**.

Software term of a class is **class** itself. A class is defined with **instance variables** and **member methods**.

Object

Object is an **identifiable entity** with some **characteristics** and **behaviour**.

E.g.: A particular rectangle object is with its **characteristics** length and breadth and its **behaviour** is finding area.

In Software an object is termed as **object** itself.

Characteristics

Characteristics can be defined as the **unique things** that an **object have**.

E.g.: The **length** and **breadth** are the **characteristics** that a rectangle object have.

Software terms of characteristics are **instance variables, member variables** and **data members**.

State

State of an object can be defined as the **values** of its characteristics at a **given point of time**.

E.g.: If a rectangle object's length is **10** and breadth is **8** then the states of the characteristics are **10** and **8**.

Software terms of state are **value**, **data**, **literal** and **constant**.

Behaviour

Behaviour is an **action** performed **by an object with its characteristics**.

E.g.: **Finding area** is one of the behaviour of an object of Rectangle class. By **multiplying** characteristics **length** and **breadth** the **area** can be found.

Software terms for behaviour are **method, function** and **procedure**.

Message

Message is a mechanism **to communicate to an object.** Computations are done executing messages.

E.g.: To activate a behaviour of an object we have to call the behaviour using object name.
Software term of message is **method call**. E.g.: **rect1.findArea()**.

Principles of OOP

Principles of OOP are **data abstraction**, **encapsulation**, **inheritance** and **polymorphism**.

Data Abstraction

Data abstraction refers to the act of **representing essential features** without including the background details or explanations in a class.

E.g.: A student object has so many characteristics like roll, name, marks, caste, address etc. To find result all these characteristics are not required. The essential features roll, name and marks only included in this class. All other background details can be avoided.

Use: Memory can be saved. Programming effort can be lessened.

Encapsulation

Encapsulation refers to the **act of wrapping up of data and functions** into a **single unit called class**.

E.g.: The student class is defined with the essential variables like roll, name, mark etc. and methods to calculate average, find result etc. as single unit class definition.

Use: Each problem can be classified and problems can be solved easily without interference of data and methods of other problems. So data are secure.

Inheritance

Inheritance is the process by which **objects of a class** (sub class / derived class) **acquire the properties of another class** (base class or super class).

E.g.: A class named **Area** is defined to find area of rectangle with variables **l** and **b** and method **findArea()** as **base class** and another class **Volume** is defined as **sub class** with a variable **h** and a method **findVolume()**. To find volume in **Volume** class the **l** and **b** can be **acquired** from the **Area** class.

Use: Repetition of variable declarations and method definitions can be avoided. So memory can be saved and programming effort can be lessened.

Polymorphism

Polymorphism is the **ability to behave differently in different contexts** by same named methods, constructors etc.

E.g.: Methods with same name like area() can be defined to find area of circle, square, etc. with different method defenitions for different results. But it requires different parameter (to act as different context).

Use: Programmer need not bother about the method name.

Elementary Concepts of Objects and Class

Class is an object factory / Class is a template / blueprint to create objects

A class is defined with characteristics and behaviours of its objects. Using this class so many objects can be created. So a class is an object factory.

E.g. Using a Rectangle class with l and b as its characteristics and area() as its behaviour so many rectangle objects can be created and area of them can be found.

Class is a modelling for entities and their behaviour / Class is a specification for objects

A class is defined with characteristics and behaviours of its objects (entities) as a model. Using this model (class) so many objects can be created. When objects are created, all objects get the characteristics and behaviour defined in the class.

E.g.: A rectangle object's characteristics are length and breadth, and its behaviour is finding area. Using the behaviour (method) area of it can be found.

Object encapsulates state and behaviour

A class is defined with characteristics and behaviours of its objects. When an object is created, the object gets all characteristics and behaviours defined in the class. These characteristics hold certain values, which are states. So the object encapsulates state and behaviour.

E.g.: A rectangle object's characteristics are length and breadth, and its state can be 10 and 8. One of its behaviour is finding area.

Objects are instances of a class

A class is an explanation about the characteristics and behaviours of an object. Object is an example for this explanation. So objects are instances of a class.

E.g.: The concept of a rectangle is that it is a shape with length and breadth. When we create one with length and breadth as 10 cm and 8 cm as an example then the real rectangle object happens.

Class is an abstraction for sets of objects

An object has many characteristics and behaviour. For a particular purpose all these featues need not to be included. So the class is defined with essential featues. With this class, sets of objects can be created. So a class is an abstraction for sets of objects.

E.g.: A student object has so many attributes like roll, name, marks, caste, gender etc. To find result it needs only roll, name and marks.

Class as user-defined data type

A class is defined with characteristics and behaviours as its members by a user. This class is used data type to create objects. Objects are varaiables of this class.

E.g.: Rectangle rect = new Rectangle(); then the **Rectangle** is a **data type** defined by the user and the object **rect** is its variable.