Chapter 2

**Object Oriented Programming**

**Class Fundamentals**

## Class

Collection of objects is called class. It is a logical entity.

A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

**Syntax**:

Class class\_name

{

Member of class;

Methods of class;

}

Creating class objects:

**Syntax:**

Classname obj\_name = new classname();

**Example :**

Student s1=new Student();

Accessing members of class

Syntax: objname .memebr variable;

Example: s1.rno;

Accessing methods of class

Syntax: objname .methodname();

Example:

s1.showdata();

s1.insertdata();

## Method:

## Create a Method

A method must be declared within a class. It is defined with the name of the method, followed by parentheses **()**. Java provides some pre-defined methods, such as System.out.println(), but you can also create your own methods to perform certain actions:

### Example

Create a method inside Main:

public class stud {

static void myMethod() {

// code to be executed

}

}

* Creating Method. Method definition consists of a method header and a method body. nameOfMethod − This is the method name.
* Method Calling. For using a method, it should be called. There are two ways in which a method is called i.e., method returns a value or returning nothing (no return ...
* The void Keyword. The void keyword allows us to create methods which do not return a value. Here, in the following example we're considering a void method methodRankPoints.
* Passing Parameters by Value. While working under calling process, arguments is to be passed. These should be in the same order as their respective parameters in the method specification.
* Method Overloading. When a class has two or more methods by the same name but different parameters, it is known as method overloading.

**w r p to addition of two numbers using class**

**using return val**

class classaddition {

public int add(int a, int b)

{

int sum = a + b;

return sum;

}

public static void main(String[] args) {

int num1 = 25;

int num2 = 15;

classaddition obj = new classaddition();

int result = obj.add(num1, num2);

System.out.println("Sum is: " + result);

}

}

**Method 2: using void**

// Without return value

class classaddition1 {

public void add(int a, int b)

{

int sum = a + b;

System.out.println("Sum is: " + sum);

}

public static void main(String[] args) {

classaddition1 obj = new classaddition1();

obj.add(10,20);

}

}

**program 1: w r t program to insert students data in a class and display it.**

import java.util.\*;

public class class1

{

int a,b;

void indata()

{

Scanner sc=new Scanner(System.in);

a= sc.nextInt();

b= sc.nextInt();

}

void showdata()

{

System.out.println("a is" +a);

System.out.println("b is" +b);

}

public static void main (String args[])

{

class1 c1 = new class1();

c1.indata();

c1.showdata();

}

}

Program 2: write a program to insert student data having student rno,student nm, s\_mob and student 3 subject marks and calculate total and percentage and remark if student per is greater than 50 then student is pass other wise student is fails and show all data

Member variables: rno,nm,mob,s1,s2,s3,tot,per

Member methods : insertdata(), cal\_tot(), cal\_per, cal\_remark() showdata()

import java.util.\*;

public class studclass

{

int rno, stud\_mob, s1,s2,s3,tot,per;

String stud\_nm;

void insertdata()

{

Scanner sc=new Scanner(System.in);

rno= sc.nextInt();

stud\_mob= sc.nextInt();

nm=sc.next();

s1= sc.nextInt();

s2= sc.nextInt();

s3= sc.nextInt();

}

void cal\_tot()

{

tot=s1+s2+s3;

}

void cal\_per()

{

per=tot/3;

}

Void cal\_remark()

{

If (per>50)

{

System.out.println(“student is pass”);

}

else

{

System.out.println(“student is pass”);

}

void showdata()

{

System.out.println("rno is" +rno);

System.out.println("mob no is" +stud\_mob);

System.out.println("name is" +nm);

System.out.println("sub 1 is" +s1);

System.out.println("sub2 is" +s2);

System.out.println("sub3 is" +s3);

System.out.println("total is" +tot);

System.out.println("percentage is" +per);

}

public static void main (String args[])

{

studclass c1 = new studclass();

c1.insertdata();

c1.cal\_tot();

c1.cal\_per();

c1.cal\_remark();

c1.showdata();

}

}

Program 3: write a program to insert student data and calculate tot and percentage and remark as per percentage and show all data

Per >70 and per<60 -.is dis

Per>60 and per<50 is first

Per >50 and per<40 -> second

Per>40 and <50 pass

Other wise fail

import java.util.\*;

public class studinfo

{

int rno,s1,s2,s3,tot,per;

String nm;

void indata()

{

Scanner sc=new Scanner(System.in);

rno= sc.nextInt();

nm=sc.next();

s1= sc.nextInt();

s2= sc.nextInt();

s3= sc.nextInt();

}

void cal\_tot()

{

tot=s1+s2+s3;

}

void cal\_per()

{

per=tot/3;

}

void cal\_remark()

{

if(per>70)

{

System.out.println(" student is dis");

}

else if (per>60 && per<70)

{

System.out.println(" student is first class");

}

else if (per>50 && per<60)

{

System.out.println(" student is second class");

}

else

{

System.out.println(" fail");

}

}

void showdata()

{

System.out.println("rno is " +rno);

System.out.println("name is " +nm);

System.out.println("sub 1 is " +s1);

System.out.println("sub2 is " +s2);

System.out.println("sub3 is " +s3);

System.out.println("total is " +tot);

System.out.println("percentage is " +per);

}

public static void main (String args[])

{

studinfo c1 = new studinfo();

c1.indata();

c1.cal\_tot();

c1.cal\_per();

c1.showdata();

c1.cal\_remark();

}

}

W a p to create class of dmart having insert records p\_id , product name, quant and p\_rate then calculate total amt of product

If amt is >5000 , customer gets 20% dis

If amt 5000-3000 - 15%

If amt 1000 -3000 -> 7 %

Other wise non of discount will get to customer.

Then calculate payable amt of customer and show data

4500 - 15% = (4500\*15)/100 x

Payable amt = amt –dis

Dmart

Pid

Pname

Pqty

Prate

Amt

Dis

Payable amt

Thank you for visit D mart

Member variables : pid,pnm,qty,rate,tot,dis,payamt

Methods: insert,cal\_tot,cal\_dis, cal\_payamt, showdata

* W a p to crate class emp having members eid,ename, empsal,empdisg,empdept and insert all these data then calculate incentive of emp

If empsal >30000 , then inc =2000

If sal 20000 – 30000 then inc = 3000

If sal 10000 -20000 then inc = 4000

Then calculate emp payable salary in their account and show all data

Memebrs : id,nm,sal,dept,disg, tt sal,paysal,inc

Methods : insert,cal\_sal, showdata

# Method Overloading in Java

If a [class](https://www.javatpoint.com/object-and-class-in-java) has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

If we have to perform only one operation, having same name of the methods increases the readability of the [program](https://www.javatpoint.com/java-programs)

**Advantage of method overloading**

Method overloading *increases the readability of the program*.

Different ways to overload the method

There are two ways to overload the method in java

1. By changing number of arguments
2. By changing the data type

// Program of addition of numbers using Method overloading

class methodoverloading {

public void add(int a, int b)

{

int sum = a + b;

System.out.println("Sum is: " + sum);

}

public void add(int x, int y, int z)

{

int sum1 = x+y+z;

System.out.println("Sum is: " + sum1);

}

public void add(double m, double n)

{

double sum2 = m+n;

System.out.println("Sum is: " + sum2);

}

public static void main(String[] args) {

methodoverloading obj = new methodoverloading();

obj.add(10,20);

obj.add(10,20,30);

obj.add(10.3,20.2);

}

}

**Recursion in Java**

Recursion in java is a process in which a method calls itself continuously. A method in java that calls itself is called recursive method.

It makes the code compact but complex to understand.

**Syntax:**

returntype methodname()

{

//code to be executed

methodname();//calling same method

}

Java Recursion Example 1: Infinite times

**public** **class** RecursionExample1

 {

**static** **void** p(){

System.out.println("hello");

p();

}

**public** **static** **void** main(String[] args) {

p();

}

}

Example : calculate factorial of numberusing recursion

public class recursionfact

{

static int factorial(int n)

{

if (n == 1)

return 1;

else

return(n \* factorial(n-1));

}

public static void main(String[] args)

{

System.out.println("Factorial of 5 is: "+factorial(5));

}

}

**Example : Addition of number using recursion**

public class recursion

{

public static int sum(int k)

{

if (k > 0)

{

return k + sum(k - 1);

}

else

{

return 0;

}

}

public static void main(String[] args)

{

int result = sum(10);

System.out.println(result);

}

}