OOP.

- C is function-oriented programming
- Global variable security

Object-Oriented Principle / Paradigm / Properties

1) Class
2) Object
3) Encapsulation — Data Hiding + Data Abstraction
4) Polymorphism — Compile time / Run-time
5) Inheritence — Single / Multiple / Multi-level / Hierarchical
6) Dynamic Binding
7) Message Passing

In Java, variables/data are called fields.
Functions are called methods.

- Class (group of entity that can be distinguished from other classes)
- Class + its properties = object (instance of a class)

- Polymorphism — one thing, many forms
- Inheritence — reusability of code

Java (OOP)

~~import~~

```
#include <stdio.h>
#include <stdlib.h>
  int main ()
  {
   int x, y, sum = 0;
   printf ("Enter values of
              x and y : ");

   scanf (" %d %d ", &x, &y);
   sum = x + y;
   printf (" Sum : %d ", sum);
   ~~return 0;~~
  }
```

concatenation operator.

```
import java.io.* ;
Class Add
{
   int x, y, sum = 0;
 Public
     void getdata ()
      {
        x = 10 ;
        y = 20 ;
      }
     void calculate ()
      { sum = x + y ;
      System.out.println
          (" The sum is " + sum);
      }
 }
 class ~~main~~ Ori
 { public static void main
            (String args [])
   { Add a1 = new Add();
     a1.getdata ();
     a1.calculate ();
 } }
```

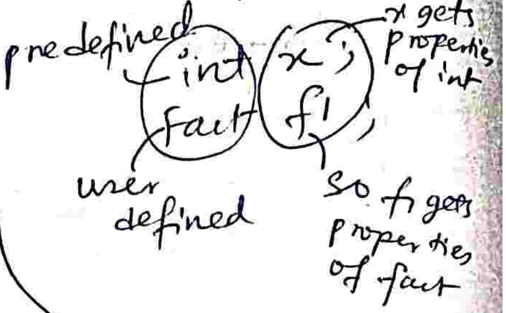
class object

Factorial

```
import java.io.sys
Class factorial
{
        int i, f=1
        Public
            void ~~getdata~~ ~~input~~ fact ()
            {
                for (i=1; i<=5; i++)
                {
                    f=f*i;
                }
            System.out.println ("Factorial is"+f);
            } 
    class main
    { public static void main (String args[])
        {
            Factorial f1 = new Factorial();
            f1.~~calcu~~ fact();
        }
    }
```

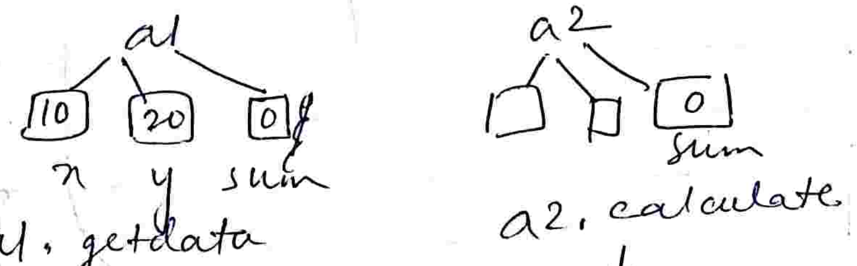
~~print~~ System — Class, out — object

★★ class
(user defined data-type
or
Abstract data type



AP

Every object gets personal copies of all global variables



#include <stdio.h>
pre-processor
(সবাইকে প্রি-প্রসেস করে)

$$\begin{tikzcd}
X.c \arrow[d, "compile"] & x.cpp \arrow[d, "compile"] \\
x.exe\ (bit\ code) & x.exe\ (bit\ code)
\end{tikzcd}$$

C, C++ → Platform dependent language

$$\begin{tikzcd}
X.java \arrow[d, "compile"] \\
x.class\ (byte\ code\ or\ intermediate\ code) \arrow[d, "interpreter"] \\
x.obj\ (executable\ code)
\end{tikzcd}$$

So, java is Platform independent language
or
Platform Neutral Lang.
or
compiler/Interpreter based language

javac (java compiler)
java (java interpreter)

present in JVM
Java Virtual Machine

```
cmd
>>> javac X.java ↵
>>> java x ↵
```

save name by main class file

C → top down programming

Java, C++ → bottom up programming

OOP 15/2/24.

Accessing Private members

```
import java.io.*;
class Add
{ int x,y, sum=0;
public
    void getdata ()
    { x=10;
      y=20;
      calculate ();
    }
private
    void calculate ()
    {
       sum = x+y;
      System.out.println ("Sum is = " + sum);
    }
}
  class Ori
  {  public static void main(string args[])
     { Add a1 = new Add ();
       a1.getdata ();
     }
  }
```

Access Specifier

1. Public
2. Private
3. Protected
4. Default or friendly

Polymorphism

Compile-time
Eg: Method Overloading

Run-time
Eg: Method ~~Overwriting~~ Overriding

Method Overloading (Compile-time Polymorphism)

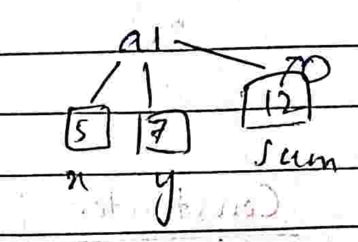
```
import java.io.*;
Class Add
{ int x, y, sum = 0;
 public
   void getdata(int a, int b)       | Add (int a, int b)
   { x = a;
     y = b;
   }
   void getdata(int a)              | Add (int a);
   {
     x = y = a;
   }
   void getdata()
   { x = 10;
     y = 20;
   }
   void calculate()                 | Add ()
   { sum = x + y;
     system.out.println("Sum is " + sum);
   }
} class dri
  {
   public static void main(String args[])
   {
     Add a1 = new Add();
     Add a2 = new Add();
     Add a3 = new Add();
     a1. getdata (5, 7);
     a2. getdata ();
     a3. getdata (12);
     a1. calculate ();
     a2. calculate ();
     a3. calculate ();
}}
```

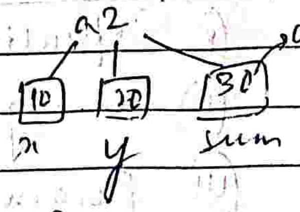
Method signature

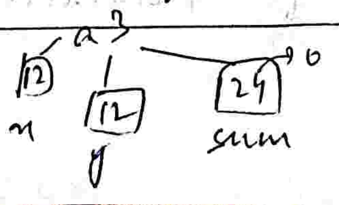
method/ constructor overloading
cause it's a special method

```
Add a1 = new
     Add(10, 20);
Add a2 = new Add(12);
Add a3 = new Add();
a1.calculate();
a2.calculate();
a3.calculate();
```

Constructor overloading







Constructors. – It is a special member method which is used to initialize the objects of a class.

```
import java.io.*;
class Add ~~(( // No-argument constructor~~
{ int x, y, sum=0;
public.
  Add() // no-argument constructer
  {
     x=10;
     y=20;
  }
  void calculate()
  { sum = x+y;
    system.out.println("Sum= "+ sum);
  }
}
 class Ori
{ p.s.v.m. (S. a[])
  { 
    Add a1 = new Add();
    Add a2 = new Add();
    a1.calculate();
    a2.calculate();
  }
}
```

(I) Name must be the same as that of a class.

(II) It doesn't have any return type (not even void)

```
Add (int a, int b) // Parameterized constructer
{ x = a;
  y = b;
}
```

```
{ Add a1 = new Add (10, 20);
  a1.calculate();
```

new operator is used for dynamic memory allocation to the fields (variables) that are in the object (happens when object is called).

this is the constructer.

এমনি তেও Constructer থাকে, compiler নিজেই থেকে বানিয়ে নেয়। যদি na থাকে program এ, so, তখন x, y এর value দিলাম, তখন garbage value এই constructor আসছে, সেটাকে বলে Default constructer.

Constructor (Types:)

(I) Default constructor

(II) No argument constructor

(III) Parameterized constructor

Accepting User Input.

import java.util.Scanner

```
import java.util.Scanner;
 import java.io.*;
class Add
{ int x, y, sum = 0;
  public
     void getdata()
     { Scanner sc = new Scanner(System.in);
       System.out.println("Enter the values of
                                   x and y: ");
         x = sc.nextInt();
         y = sc.nextInt();
     }
    void calculate()
    { sum = x+y;
      System.out.println("Sum = " + sum);
    }
}
class Ori
  { public static void main(String args[]);
    { Add a1 = new Add();
       a1.getdata();
       a1.calculate();
    }
  }
```

Java তে By default সব string এ নেয়

clas obj constructor parameterised

this line connects computer to hardware (keyboard) so, এখানে sc মানে keyboard.

this line means connecting to smonifer

takes input and converts it into int, for float, sc.nextFloat(), etc for char: sc.nextCharAt(), etc

// check Scanner এর line টা getdata() তে না লিখে ~~main~~ Ori-র মধ্যে লিখলে হবে কি?