OOPs - Class-1

Date- 23/10/2023

```
int fun()
{
  Imp kaam
}
int main(){        — fun
 int a=fun();      — function
 cout<<a<<endl;
}
```

* functional Programming

$f_1 \rightarrow f_2 \rightarrow f_3$

f_1

main

return ho jayega.

=> Problem Statement → 2 Students print their
- id
- name
- age
- no of sub.

how you can solve this.

```
int main(){
 int id1, id2;
 string name1, name2;
  int nos1, nos2;
  int mark1[5], mark[5];
 }
```

50 students hai toh kaun bnayega 50 variable bhut muskil hai
- Easy to implement nhi hai
- Easy to extend nhi hai
- Maintainenance bhi nhi hai

OOps father
- (1) All these things perform in functional Programming
- (2) Student → Real life role
  - id
  - name
  - marks
  - nos
  - age

Behaviour of student bhi hota hai
- study()
- Cheating()
- Bunk()
- Gaming()

Problem
↳ Student ke sari properties or behaviour ko programming me kaise likhe

```
class Student {
    int id;
    int age;
    String name;
    int mos;
    void study()
    void Bunk()
    void gaming()
}
```

Properties / State / attribute

Behaviour / method fun

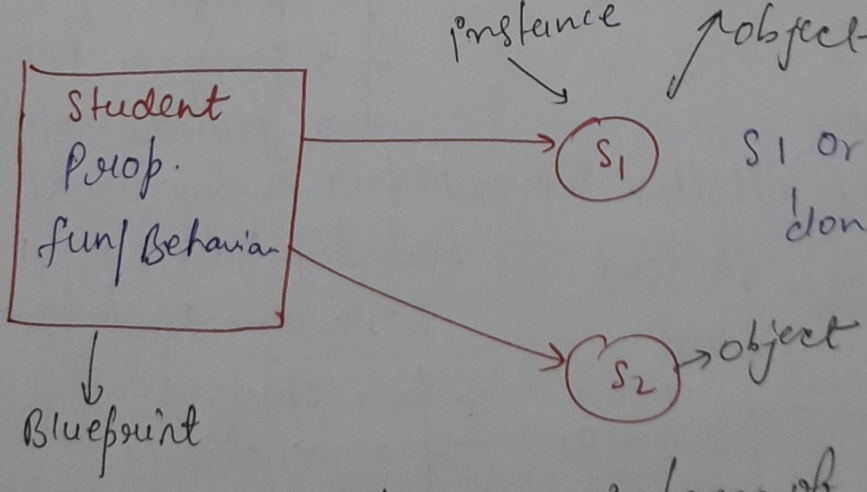
Agar mujhe koi bhi kaam karana hota tha toh uske liye function likh deta the jo jaise palindrome print karna ho or ye sab kum functional programming me karte hai

It means function kaam karte hai

Class → Bundle of properties and behaviour.

* First capi letter capital me rakhte hai class bnate samay. jo bhi class ka name hota hai usko.

Real life me pehle blue print bnate hai phir use implement karte hai same things happen in programming.



Object → instance of class

thought come to reality

Empty class ka size → 1 byte ata hai

→ Object Oriented prog →.

Student ko bundle me pack kar diya aur bola jo jo is bundle ka instance banega vo student khelga.

class → DT

① Empty class size = ?.

② class non-empty hai toh size = ?.

1 byte → smallest addressable memory.

Object

(S1) Props behaviour

(S2) props behaviour.

We can call custom datatype to class.

```
class Student{
    int age;
    void study(){ }
};
int main(){
    cout<<sizeof(Student)<<endl;
```

function ka size nhi hoga

```
void study(){
    int a;
}
```

func^n jab call hoga toh memory me aayega.

Empty class me 1 byte hoga. memory allocate.

→ 4 byte.

```
class Student{
    int id;
    int age;
    bool present;
```

→ 8 byte (ab) adding bool → 12 byte

Why

int
int
bool

→ 8 byte

9 byte ana chahiye

→ lekin 12 byte print ho rha hai

compiler bol rha hai ki mai jab 4 byte utha ta hu toh fast processing hoti hai

CPU Register

4 byte → 32 bit

memory. Pipe

fast fast

| 4 | 4 | 1 | 3 |

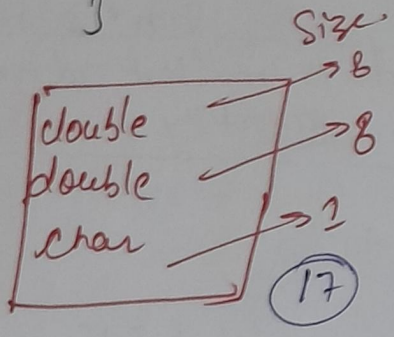
extra byte adding ⇒ padding

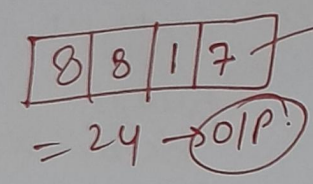
```
class Student {
    double a
    double b;
    char c;
}
```

print => 17

Modern compiler like hi bnae huye hai

compiler dekhe 4 byte hai nhi then compiler apne aap ko ready karta hai to pic sabse big data type.





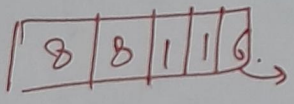
= 24 → O/P

① Sabse big datatype

② Align sum of datatype in the boundary of sabse bada datatype.

Sabse bade data type ke nearest multiple pe le aao memory me.

double
double
char
bool

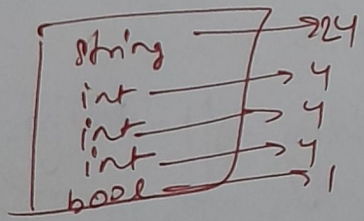


* Internally working of compiler

size print hoga (24) =

Sabse bada data type string hai agar class ke andar string aur kuch declare karegye toh uski size 24 byte ayegi

Abstraction → upar upar se use karo don't go into implementation



24 → multiple → 48

24+4+4+4+1+(1)

↓ Padding

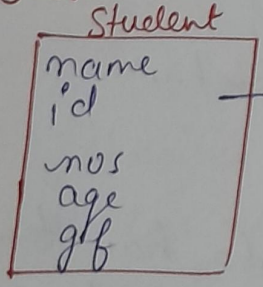
37 byte.

print O/p → 48.

Life → har insaan ki kuch ① private info
② public info
③ protected info

* class by default ~~bnate ho~~ toh private hoti hai
by default humari information bhi private hoti hai
Must be a way jo chize public kar paaye-
↳ Access modifiers → They define scope of access.
class me kuch info-
public hogi or kuch
private hogi
and private wali data access nhi kar payegye.

Better tarika




```
student S1;
S1.name = "goli";
S1.age = 12;
S1.id = S12;
S1.gf =
```

→ baar likha pead baad rha hai

To resolve this we use constructor.

```
constructor()
{

}
```

It is the only function in the class or in C++ that has no return type.

Short for → ctor ⇒ It construct an instance of class.

Syntax → class_name() {
}.

class ke properties ki value ko assign karne ke liye constructor banate hai
by default agar hum constructor nhi likhegye toh compiler ne khud se add kar diya tha.

constructor mandatory hota hai class me.
aur jab hum khud se likh degye constructor toh compiler add nhi karega fir se..

⇒ constructor call hota hai in data members ke value assign karne ke liye.

⇒ toh default constructor inki value garbage assign kar dega.

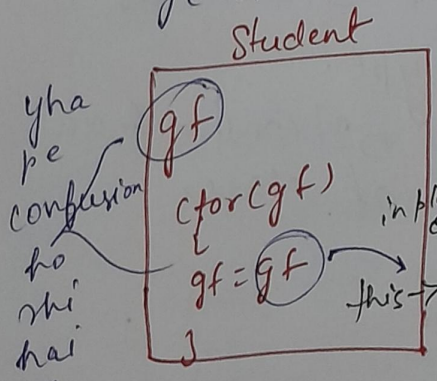
constructor public hota hai

```
class Student{
public
int id;
int age;
bool present;
Student(){
cout<<"Student ctor called";
}
main(){
Student s1;
s1.age;
}
```

#output → Student ctor called
0.

* By default
* Parameterised constructor
* multiple constructor create kar sakte hai

Polymorphism → One names many forms.
yha pe Student alag form dikha rha hai toh ye student constructor bhi polymorphism hai




```
int main(){
Student s1(----, "chutki")
}
```

By default class ke andar ek aur variable hota hai aur vo pointer type ka hota hai

this.

this is a pointer that points at current object.

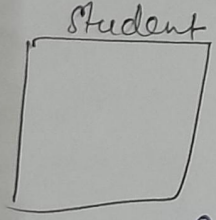
~~cout~~ * this pointer added by compiler privately.

~~Starting me kya~~

Student bhi datatype ki tarah act kar rha tha.

Toh student ko heap me allocate karna hai toh kaise kare.

```
Student *s = new Student
                ( constructor
                  called kar
                  do
                );
```



Student s1( )

↓

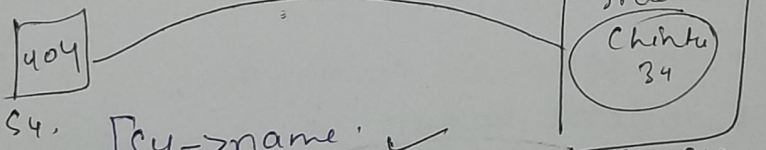
stack

Obj is stack

jab object stack me allocate hota hai toh `s1.name` se access karte hai.

student *s4 = new student(———);

heap



heap pe object bna hua hai toh toh data members ko access karne ka tarika

```
s4->name;
(*s4).name;
```

→ access allocate karega

Student s1 → int a
" s2 → " b.

int a allocate karte hai toh ye stack pe allocate hoga

Heap pe agar int allocate karna ho toh

```
int *a = new int(5);
```

(*s4).name

Agar object heap me allocate kar rhe hai toh ~~free~~ memory leak hogi isliye memory free karna padega

```
delete s4;
```

14.10.

1. const
2. static
3. initialisation distance
4. Macro