Day 7

12 April 2024 01:03 PM

function of Structure –

```
#include<iostream>
#include<string>
using namespace std;
struct student
{
    int roll;
    char name[20];
    float per;
};
student input()
{
    student temp;
    cin>>temp.roll;
    cin.ignore();
    cin.getline(temp.name,20);
    cin>>temp.per;
    return temp;
}
void output(student s)
{
    cout<<s.roll<<"\t"<<s.name<<'\t'<<s.per<<endl;
}
int main()
{
    student s1,s2;
    cout<<"Enter roll name and per of a student:";
    s1=input();
    cout<<"Enter roll name and per of a student:";
    s2=input();
    output(s1);
    output(s2);
    return 0;
}
```

pointer of a structure :→

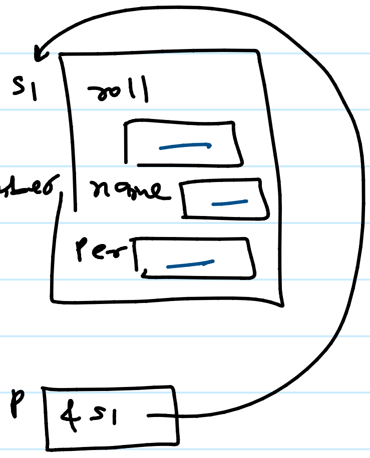
student s1; →

. high

s1.roll = 101; → object to member

* low

student *p = &s1;



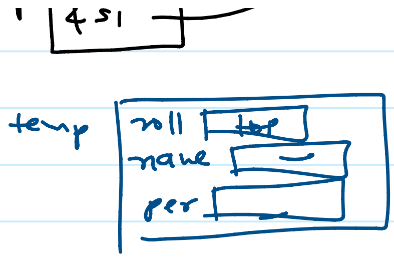
student temp;

shallow copy $\rightarrow$

student temp;

temp = $\underset{s1}{*p}$;

temp.roll=101;



(*p).roll =101;

$\boxed{\text{or}}$

p->roll=101; $\leftarrow$ pointer to member

.

```
#include<iostream>
#include<string>
using namespace std;
struct student
{
    int roll;
    char name[20];
    float per;
};
void input(student *p)
{
    cin>>p->roll;
    cin.ignore();
    cin.getline(p->name,20);
    cin>>p->per;
}
void output(student s)
{
    cout<<s.roll<<"\t"<<s.name<<'\t'<<s.per<<endl;
}
int main()
{
    student s1,s2;
    cout<<"Enter roll name and per of a student:";
    input(&s1);         //call by address
    cout<<"Enter roll name and per of a student:";
    input(&s2);
    output(s1);
    output(s2);
    return 0;
}
```

.

```
#include<iostream>
#include<string>
using namespace std;
struct student
{
    int roll;
    char name[20];
    float per;
};
void input(student &p)
{
    cin>>p.roll;
    cin.ignore();
    cin.getline(p.name,20);
    cin>>p.per;
}
void output(student s)
{
    cout<<s.roll<<"\t"<<s.name<<'\t'<<s.per<<endl;
}
int main()
{
    student s1,s2;
    cout<<"Enter roll name and per of a student:";
    input(s1);          //call by reference
    cout<<"Enter roll name and per of a student:";
    input(s2);
    output(s1);
    output(s2);
    return 0;
}
```

<u>static</u> (storage class)

```
void  inc ( )
{
        int a=5;
        a++;
        cout<<a;
}
int main()
{
        inc ();
        inc ();
```

a $\boxed{\cancel{5}\,6}$

output

$\boxed{6\ \ 6\ \ 6}$

```
{          inc();
           inc();
           inc();
}
```

```
void inc()
{
        static int a=5;
        a++;
        cout<< a;
}
int main()
{          inc();
           inc();
           inc();
}
```

a $\boxed{\cancel{5}\,\cancel{6}\,\cancel{7}\,8}$ ← Life → global
scope – local

$\boxed{6 \quad 7 \quad 8}$

typedef int chintu;

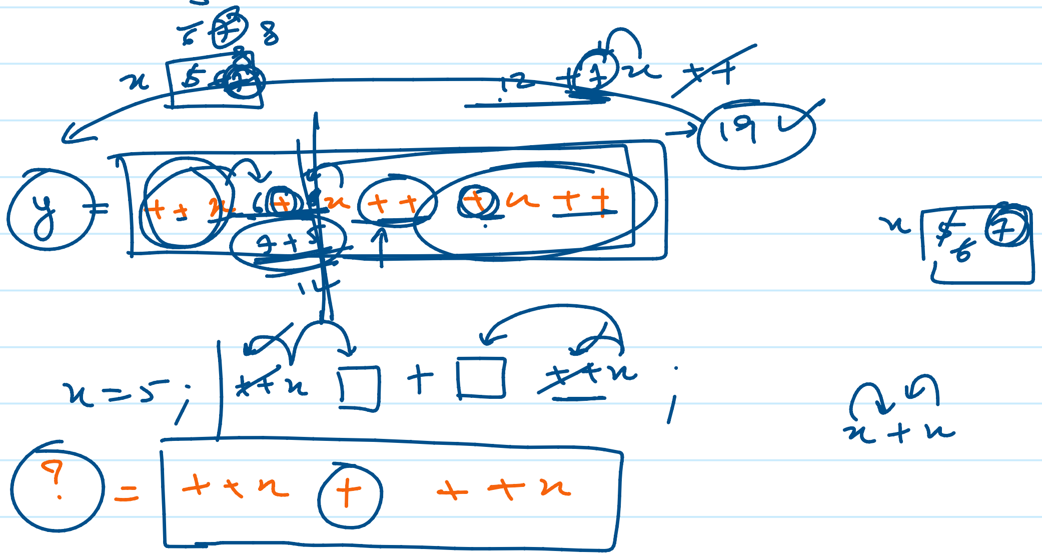
<u>class</u> & <u>object</u> :→ user defined data type

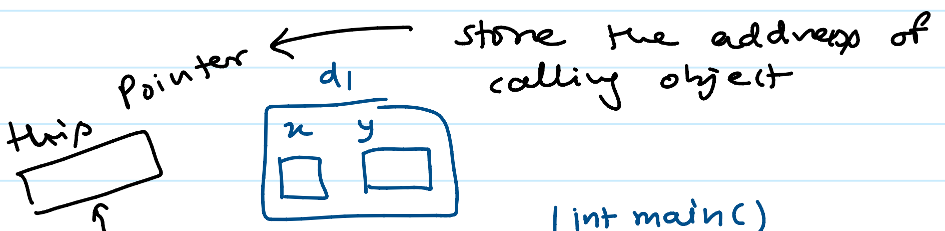
```
class student
{
    private:   ← default
            int roll;
            char name[10];      } member
            float per;            variables

    public:
            void input()        } member
            {                     function
                =
            }
```

```
      {
         =
      }
      void output()
      {
         =
      }
};
```

funct





```
class data
{
        int x;
        float y;
        public:
            void setdata(int x, float y)
            {
              this -> x = x;
              this -> y = y;
            }
            void output()
            {
                  cout<<x<<" "<<y<<"\n";
            }
};
```

```
int main()
{
    data d1;
    d1.setdata(10, 8.7);
    d1.output();
}
```

&d1

calling object

member function outside the class :-

```
class data
{       int x;
```

```
class data
{       int x;
        float y;
        public:
          void ~~setdata (int x, float y)~~   |  void setdata (int, float);
          {
            ~~=~~
            ~~=~~
            ~~=~~
          }
};
inline void data::setdata (int x, float y)
  {
      this -> x=x;  ✓
  }
```

```
#define  sum (a,b)   a+b
            ↑         ↑
           call      Return          ← Compile time

sum (10,20)  10+20

                  100+200
sum (100,200)
```

inline → Request to the compiler

```
~~inline~~  int  sum (int a,  int b)
  {
      return a+b;
  }

sum (10,20);
```

Static & Constant :→

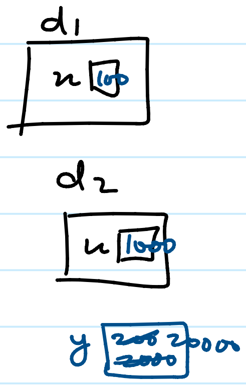
Static & Constant :→

```
class data
{     public:
          int x;            ← Instance member Variable (member of object)
          static int y;     ← static member variable (member of class)

};
int data::y;  ← memory ✓
```

```
int main()
{
        data d1;
        data d2;
        d1.x=100;
        d1.y=200;
        d2.x=1000;
        d2.y=2000;
        data::y=20000;
```

Error → ~~data::x=100;~~



```
#include<iostream>
using namespace std;
class data
{
    int x;
    int y;
    public:
        data(int x=0, int y=0)
        {
            this->x=x;
            this->y=y;
        }
        void fun1()     //instance member function
        {
            x=10;
            y=20;
        }
        static void fun2()      //static function
        {
            //x=100;
            //y=200;
            data temp;
            temp.x=100;
        }
        void output() const  //constant function
        {
            //a=100;     error
            cout<<x<<' '<<y<<endl;
        }
};
//int data::y;
int main()
{
    data d1;
    d1.fun1();
    d1.output();
    const data d2(100,200);
    //d2.fun1();
    d2.output();
}
```