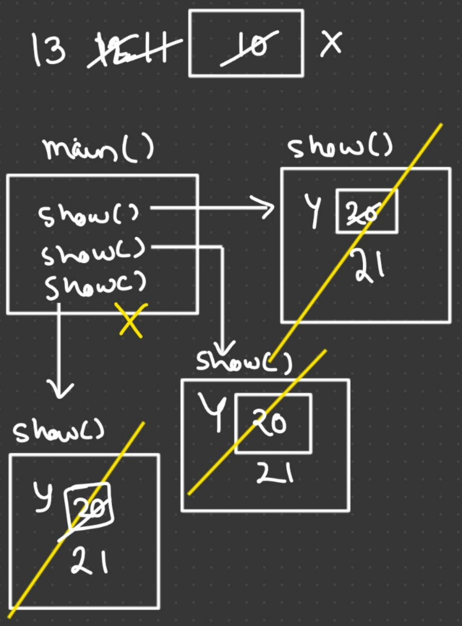
Static Variables & function

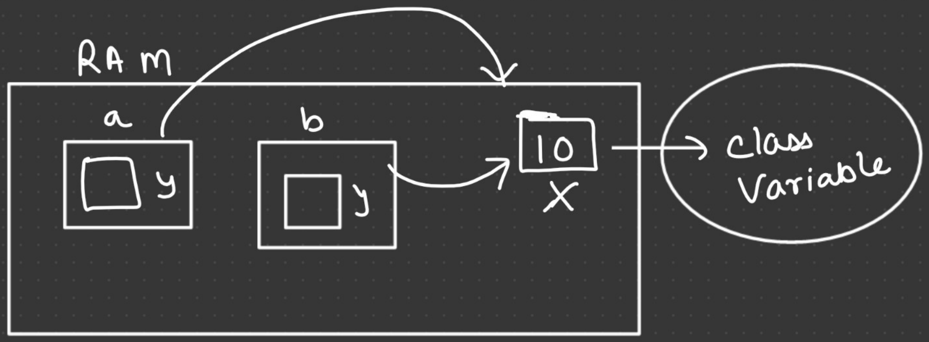
```
int  show()
{
    Static   int  x = 10;
             int  y = 20;

    cout<< x<<y;
       x++;
       y++;
}
```

```
int main()
{
    show();
    show();
    show();

    return 0;
}
```



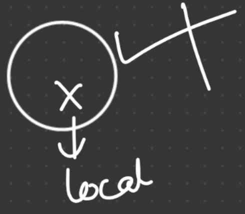
```
10 , 20
11, 20
12, 20
```

```
Class  A
{ public:
    static  int  x;
    int  y;
};
int  A::x  = 10;

int main()
{
    A a, b;

    return 0;
}
```

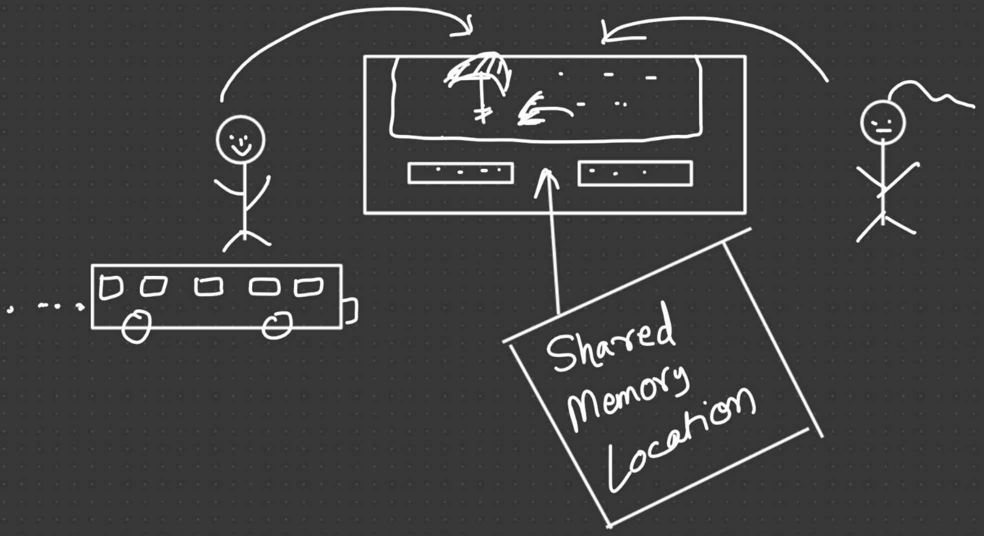


1) A::x

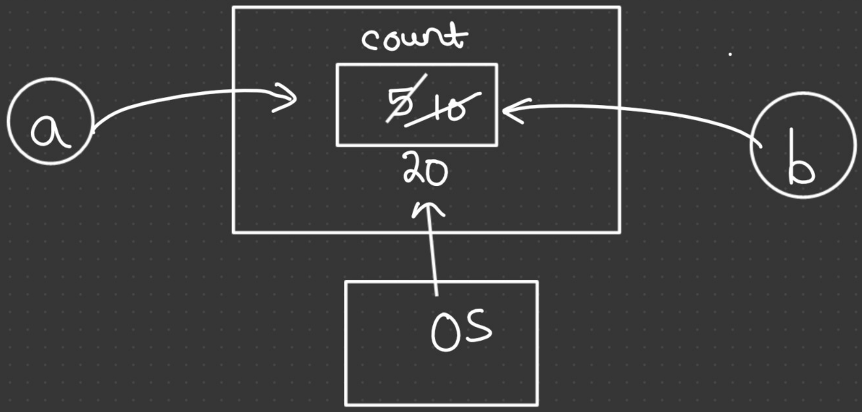
2) a.x , b.x



Always remember to intialize the static variable Outside the class using :: operator



class variable



Static function

```
Class A
{
   public :

        Static int x;
        int y;

        static void show()
        {
            cout<<x; ——> This is allowed
            cout<<y; ——> This is not allowed.
        }

};
```

static fn can only access static variables. because non-static member may not exist before creation of any object

And we can call a static fn even if no object of our class is created.

```
A::show()    ✓

A a;
  a.show()   ✓
```

But non-static function (like main f^n) can access both static & non-static variables.