Constructor & Destructor

Q What are const?

A
- Const r a type of function.
- It is called automatically when an obj is created.
- Const do not create obj, rather after creation of an obj, const r called.

Q what does a const do?

A It can perform any task at the time of obj creation.

For eg :- It can be used as an ~~alternative~~ alternative to the set fun. i.e it can assign values to var at time of obj creation.

Q Why r const needed?

A To perform any task ~~@~~ at the time of obj creation.

Q What is the benefit of const?

A
- It can save effort.

  For eg. Set fun need not be called separately for assigning values.
- Const can ~~do~~ assign at time of obj creation.

Q When is the const called?.

A When obj is created. It can't be called it any other time except when obj is created.

Q Who calls the const?/ Do you need to call it explicitly?

A • User doesn't call the const.
It is called auto^{y}!

Q How many times can you call a const for each obj?

A • Const can be called only once for an obj.
It can't be called multiple times.

Q How do you call the const?

A • User need not call the Const. It is called auto^{y}.

Q Write some characteristics of a Const.

Q Write differences b/w a Const & a member fun.

A

Const.	MF
• Must have same names as the Class	• Can have any name.
• Must not have any return type	• Can have void, int, char etc return type.
• Called auto^{y} at the time of obj creation	• Need to be called explicitily by the ~~object~~ obj.
• It is called only once	• Can be called any No. of times
• eg student(int x, int y int z);	eg void set(int x, int y, int z);

Q Steps to use a const.

Q WAP to ~~use~~ show use of a Const & dest.

A <Done later>

Destructors

Q What is a dest?

A
- It is a type of fun.
- It is called auto^{y} when an obj is destroyed.
- ~~Const~~ Dest, do not destroy obj, rather they are called by an obj is destoyed.

Q What does a dest do?

A Any task @ at the time of obj destruction

For eg:- It is mostly used to handle file ~~relatalded~~ related tasks.

- It can be used to save any unsaved file opened by the obj.

Q Why r dest needed?

A To perform tasks that need to be done at time of obj destruction

Q Benefit of using a dest?

A If user forgets to perform any task that needs to be done b4 the obj is destroyed, then dest can do it auto^{y}. For eg:

- B4 destruction of obj, dest. will Check whether user has saved the file or not. If not, then dest will pop up the "save file" option for user.

- - - - - - - - - - - - - -

Q When is a dest called?

A Just b4 obj is destoyed.
It can't be called b4 that.

Q When does the obj gets destroyed?

A When its scope gets over.

eg1
```
int main
{
        obj1, obj2   // created here



} // scope over - obj destroyed.
```

```
eg2 int main()
    {
        obj1, obj2
        —
        —
        —
        —
        if (...)
        {
            obj3, obj4
            —
            —
        } // Scope for obj 3, 4 over
                          - two dest will
                            be called.
        obj5

    } // Scope for obj 1, 2, 5 over
        three dest will be called.
```

Order of dest will be reverse of their creation i.e obj 5 destoyed first, then obj 2, then obj 1.

Q Who calls the dest? Doesnt it need to be called explicitly?

A. No. Its called auto^{y}.

. User can't call dest.

Q How many times can you call a dest ?

A Only once. when the obj is destroyed.

Q How do you call a dest?

A User can't call it. Its called auto^{y}.

Q ~~A~~ Char. of dest ?

A
- Same name as Class. Use ~ sign.
  eg ~student ( )
- No return type. Not even void.
- Called auto^{y}.
- Called only once
- eg ~student ( );

Q Steps to use a Const & Dest.
WAP to show the use of a Const & dest.?

A
```
#...
using...
class student
{
        private:
        int rn, m1, m2;
        public:
        void set (int x, int y, int z);
        //You can have both set &
        // Const in your program.
        void get ( );
        void add ( );
//Const //   student (int x, int y, int z);
//dest //.   ~student ( );

 };
void student :: set (int x, int y, int z)
{
        rn = x;
        m1 = y;
        m2 = z;
}
void student :: get ( )
{
        cout << rn << m1 << m2;
}
void student :: add ( )
{
        cout << m1 + m2;
}
```

~~void Stu~~

```
□ Student :: student (int x, it y, int z)
↑          ↑          ↑
No void    class      fun          // Const.
int etc    name       name            definition
{
    rn = x;
    m1= y;
    m2=z;
    cout << "const is called";
}

□ student :: ~student ( )   // dest definition
{
    cout << "dest is called" << endl;

}
```

```
int main
{
① student obj1(1, 10, 20), obj2(2,20,30);
```

// when you use const, it is mandatory to pass values when obj is created.

// otherwise, there will be an error.

// Here following events occur

1) obj created
2) const called auto.
3) "Const is called" printed
4) values are assigned to var;

//Now no need to call set fun.

```
② obj1.get();
③ obj1.add();
④ obj2.get();
⑤ obj2.add();
```

```
if (1)  // if(1) → 1 is written so
        // that the block inside if
{       //is always run.
(6)  Student obj3 (3,50,60), obj4(4,90,100);
(7)     obj3.get();
(8)     obj3.add();
(9)     ob4.get();
(10a)   ob4.add();
(10b) }  // Obj3 & 4 destroyed here.
(11) Student obj5(5,10,20); // new obj
                               created.
(12)   obj5.get();
(13)   ob5.add();
(14) } //Main ends here.
   // Obj5 destoyed.
   // Obj2   "
   //obj1    "
```

<u>Output</u> :-

```
Line-1: Const is called
        Const is called.

Line-2: 1, 10, 20
     3: 30
     4: 2, 20, 30
     5: 50
     6: Const is called
        Const is called
     7: 3,50,60
     8: 110
```

9: 4, 90, 100

10a: 190.

10b: dest is called
dest is called.

11: Const is called

12: 5, 10, 20

13: 30

14: dest is called
dest is called
dest is called.

Practice Ques:-

Q Write differences b/w Const & dest.