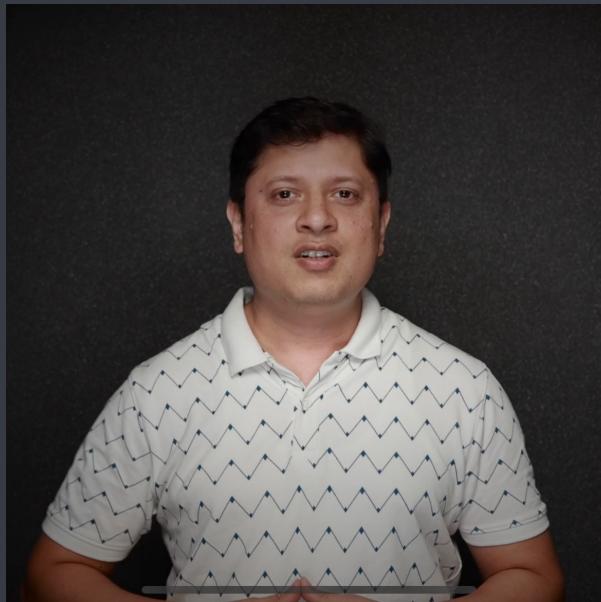


C++ in depth

# Constructor & destructor in Inheritance

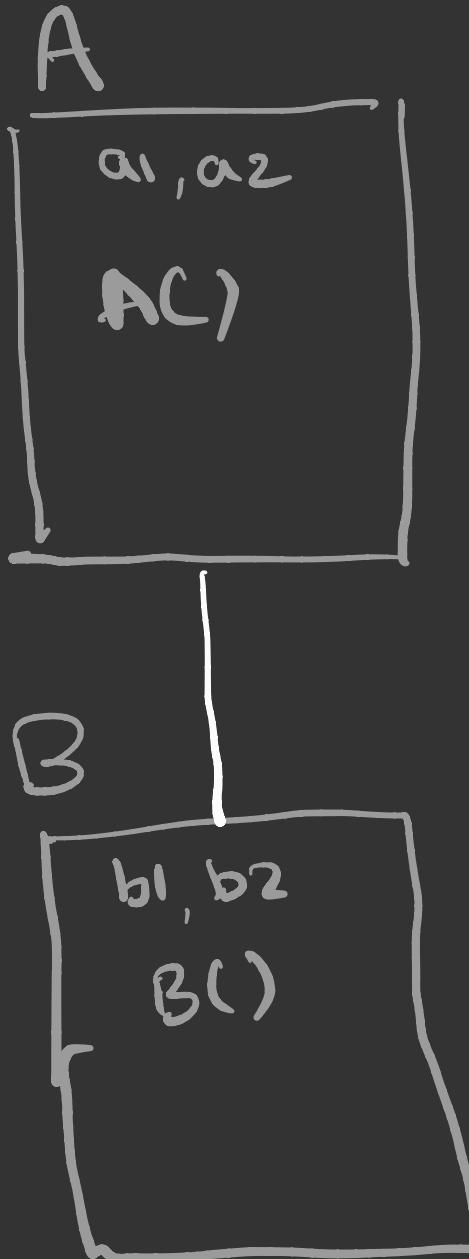


Saurabh Shukla (MySirG)

## Agenda

- ① Constructor in inheritance
- ② Destructor in inheritance

# Constructors in Inheritance



What is the purpose of constructor?  
To initialize properties of  
newly created object

obj  
a, a2  
b1, b2

Constructor never inherits.  
child class constructor implicitly or  
explicitly invokes constructor of parent  
class

## Case 1

When no explicit constructor is defined by the programmer in class A and class B

class A

{

public:  
A()  
{ }

}

class B : public A

{

public:  
B() : A()  
{ }

}

B obj; ← It calls constructor of B class.

∴ order of constructor calls B() then A()

but execution order is  
A() then B()

## Case 2

class A

```
{ public:  
    A()  
    { ... }  
}
```

When A class contains non parameterized constructor (explicitly defined by programmer) and no explicit constructor is defined in child class B.

3:

Class B : public A

{

```
public:  
    B(): A()  
    {}  
}
```

order of call      B() then A()

order of execution      A() then  
                            B()

### Case 3

Class A

{

public:

A(int a)  
{ ... }

}

class B : public A argument.

{

public:

B():A()  
{ }

};

when explicit parameterized constructor in class A is provided by the programmer and no explicit constructor is provided in class B.

Note: There is no constructor in class A which takes zero

B obj; Error

Yadi parent class mein non parameterized constructor hai hi nahi to yeh compulsory hai ki child class mein programmer constructor banaye, sath hi parent class ke constructor ko call karne ka kaam bhi kare

#### Case 4:

class A

{

public:

A(int a){...}

}

class B : public A

{

public:

B(): A()

{...}

When class A has an explicit parameterized constructor and no constructor with zero arguments. Class B has an explicit constructor but call to parent class constructor is not explicitly written.

B obj; Error

}

## Case 5

Class A

```
{  
public:  
    A(int a)  
};
```

y;

Class B : public A

```
{  
public:  
    B(): A(5)  
};
```

y;

Class A has only parameterized constructor and no non parameterized constructor. Class B has explicit constructor which calls parameterized constructor of parent class.

B obj; Correct

## Case 6

Class A

```
{ public:
```

```
    A()
```

```
{ ... }
```

```
    A(int a)
```

```
{ ... }
```

when parent class has multiple constructors along with no argument constructor then it is programmer's choice to provide explicit constructor in child class or not.

```
}
```

```
<class B : public A
```

```
{ public:
```

```
    B() : A()
```

```
{ ... }
```

```
    B(int x, int y) : A(x)
```

```
{ ... }
```

```
}
```

```
B obj(5,4);
```

## Destructor in Inheritance

Destructor never inherits.

No need to call parent's destructor explicitly.

```
Class A
{
    public:
        ~A()
    {
        =
    }
};
```

```
Class B : public A
{
    public:
        ~B()
    {
        ~A() ← call to destructor of
        parent class.
    }
}
```

{     B obj;  
=====  
}  
} ←  
order of call ~B() then ~A()  
execution order ~B() then ~A()