



BITS F232: FOUNDATIONS OF DATA STRUCTURES & ALGORITHMS (1ST SEMESTER 2023-24) C++ CONTINUED...

Chittaranjan Hota, PhD
Sr. Professor of Computer Sc.
BITS-Pilani Hyderabad Campus
[hota\[AT\]hyderabad.bits-pilani.ac.in](mailto:hota[AT]hyderabad.bits-pilani.ac.in)

CONSTRUCTORS IN C++

- A constructor in C++ is a **member function** of a class which **initializes** objects of a class and allocates storage. Automatically called (with the **same class name**) when the object is created.

```
1 #include <iostream>
2 using namespace std;
3 class Demo
4 {
5     public:
6         int a;
7     private:
8         Demo() {
9             a=10;
10        }
11    public:
12        static Demo getobj() {
13            Demo obj;
14            return obj;
15        }
16 };
17 int main()
18 {
19     Demo obj=Demo::getobj();
20     cout<<obj.a;
21     return 0;
22 }
```

main.cpp:10 private within this context

Result

CPU Time: 0.00 sec(s), Memory: 3508 kilobyte(s)

area of rectangle: 45

Default constructor (No args.)

Parameterized (Takes args.)

Copy constructor (initializes one object using another object)

Are there any return types for constructors?

EXAMPLES OF DEFAULT AND COPY CONSTRUCTORS

```
1 #include <iostream>
2 using namespace std;
3 class Area {
4 public:
5     int area;
6     Area() {
7         area = 0;
8     }
9     Area(int side) {
10         area = side * side;
11     }
12     Area(int length, int width) {
13         area = length * width;
14     }
15     int disp() {
16         return area;
17     }
18 };
19 int main() {
20     Area obj1;
21     Area obj2(3);
22     Area obj3(7, 5);
23     cout << "Area of obj1: " << obj1.disp() << endl;
24     cout << "Area of obj2: " << obj2.disp() << endl;
25     cout << "Area of obj3: " << obj3.disp() << endl;
26     return 0;
27 }
```

Constructor Overloading

```
Area of obj1: 0
Area of obj2: 9
Area of obj3: 35
```

Result

CPU Time: 0.00 sec(s), Memory: 3508 kilobyte(s)

```
a= 55
b= 78
```

A copy constructor is a member function that initializes an object using another object of the same class.

Result

CPU Time: 0.00 sec(s), Memory: 3276 kilobyte(s)

```
230
```

```
1 #include <iostream>
2 using namespace std;
3 class A {
4 public:
5     int x;
6     A(int a)
7     {
8         x=a;
9     }
10    A(A &i)
11    {
12        x = i.x;
13    }
14 };
15 int main() {
16     A a1(230);
17     A a2(a1);
18     cout<<a2.x;
19     return 0;
20 }
21
```

DESTRUCTORS IN C++

- A destructor is a member function that is automatically called when a class object ceases to exist (delete operator is called, execution ends, scope of local variable ends)
- It takes no arguments and has no return type.

```
BITS Pilani  
BITS Goa  
BITS Hyderabad
```

Is destructor overloading allowed in C++?

```
1  #include <iostream>  
2  using namespace std;  
3  
4  class Example1{  
5      public:  
6          Example1(){  
7              cout << "BITS Pilani" << endl;  
8          }  
9          ~Example1(){  
10             cout << "BITS Hyderabad" << endl;  
11         }  
12  
13         void display()  
14         {  
15             cout << "BITS Goa" << endl;  
16         }  
17     };  
18  
19     int main() {  
20         Example1 ex;  
21         ex.display();  
22     }
```

CLASS INHERITANCE IN C++

Why is inheritance used in C++?

```
class Person {  
    private:  
        string name;  
        int Aadhaar;  
    public:  
        void print();  
        string getName();  
};
```

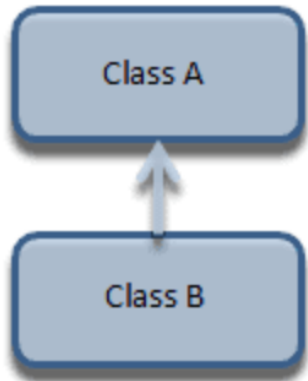
(Base/parent/superclass)

```
class Student : public Person {  
    private:  
        string branch;  
        int gradYear;  
        double cgpa;  
        string idNo;  
    public:  
        void print();  
};
```

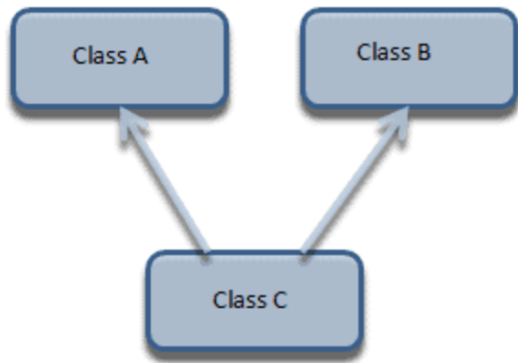
(Derived/child/subclass)

How will you draw the class inheritance diagram?

EXAMPLES



(Single Inheritance)



(Multiple Inheritance)

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class Animal
5 {
6     string name=" ";
7     public:
8     int tail = 1;
9 };
10
11 class Dog : public Animal
12 {
13     public:
14     void voiceAction()
15     {
16         cout<<"Barks!";
17     }
18 };
19 int main()
20 {
21     Dog d;
22     cout<<"Dog has "<<d.tail<<" tail"<<endl;
23     cout<<"Dog ";
24     d.voiceAction();
25 }
```

```
Dog has 1 tail
Dog Barks!
```

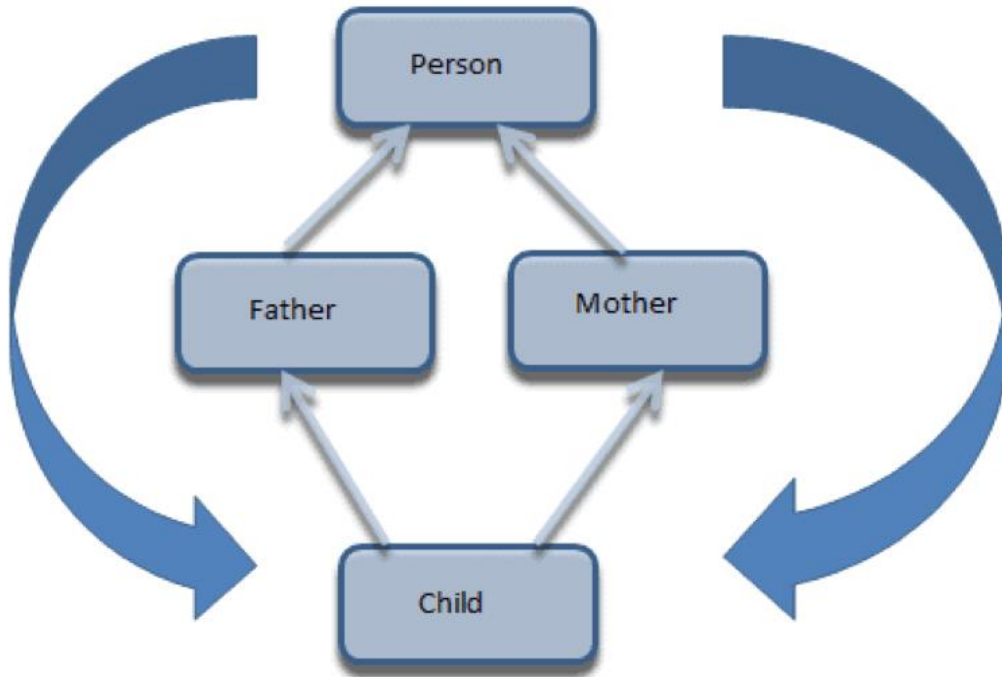
```
1 #include <iostream>
2 using namespace std;
3 class student_marks {
4     protected:
5     int rollNo, marks1, marks2;
6     public:
7     void get() {
8         cout << "Enter the ID No.: "; cin >> rollNo;
9         cout << "Enter the Midsem and Compre marks: "; cin >> marks1 >> marks2;
10    }
11 };
12 class lab_marks {
13     protected:
14     int lmarks;
15     public:
16     void getlm() {
17         cout << "Enter the mark for lab exam: "; cin >> lmarks;
18    }
19 };
20
21 class Result : public student_marks, public lab_marks {
22     int total_marks;
23     public:
24     void display()
25     {
26         total_marks = (marks1 + marks2 + lmarks);
27         cout << "\nID No: " << rollNo << "\nTotal marks: " << total_marks;
28    }
29 };
30 int main()
31 {
32     Result res;
33     res.get();
34     res.getlm();
35     res.display();
36 }
```

```
input
Enter the Midsem and Compre marks: 50 80
Enter the mark for lab exam: 30

ID No: 1
Total marks: 160
```

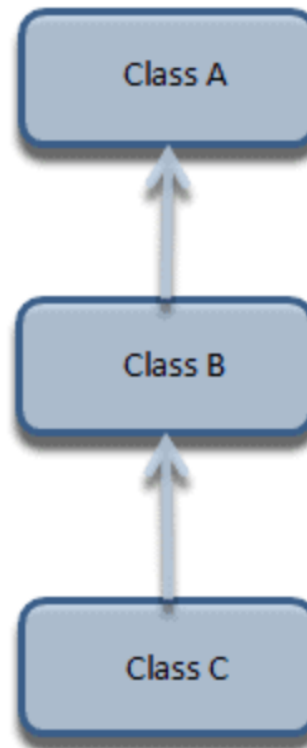
EXAMPLES CONTINUED...

(Sol: Make base class **virtual**)



(The Diamond Problem in C++)

(Img. Source: www.softwaretestinghelp.com)



(Multi-level Inheritance)

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class Animal
5 {
6     string name=" ";
7     public:
8     int tail = 1;
9 };
10 class Dog : public Animal
11 {
12     public:
13     void voiceAction()
14     {
15         cout<<"Barks!";
16     }
17 };
18 class Puppy : public Dog{
19     public:
20     void weeping()
21     {
22         cout<<"Sheds tears!";
23     }
24 };
25 int main()
26 {
27     Puppy p;
28     cout<<"Puppy has "<<p.tail<<" tail"<<endl;
29     cout<<"Puppy ";
30     p.voiceAction();
31     cout<<" Puppy ";
32     p.weeping();
33 }
```

Puppy has 1 tail
Puppy Barks! Puppy Sheds tears!

POLYMORPHISM IN C++

