



AWESH ISLAM
BUET, CSE

BATCH RECURSION

C AND C++

PROGRAMMING MASTERCLASS

C++ Class-09



SHAROARE HOSAN EMON
BME, BUET

আমাদের সবগুলো ক্লাস দেখার জন্য ভিজিট করো
<https://www.hsccrackers.com/>



SCAN ME

Namespace and Others

Namespace

```
8  #include <iostream>
9  using namespace std;
10 namespace firstNS {
11 class demo{
12     int i;
13 public:
14     demo(int x){ i = x; }
15     void seti(int x) { i = x; }
16     int geti() {return i;}
17 };
18 char str[] = "Illustrating namespace";
19 int counter;
20 }
21 namespace secondNS {
22 int x,y;
23 }
24 int main(int argc, const char * argv[]) {
25     firstNS::demo ob(10);
26     cout<<"Value of ob is: "<<ob.geti()<<endl;
27     using firstNS::str;
28     cout<<str<<endl;
29
30     using namespace firstNS;
31     for(counter = 10;counter;counter--){
32         cout<<counter<<" ";
33     }
34     cout<<endl;
35     secondNS::x = 10;
36     secondNS::y = 20;
37     cout<<" x,y : "<<secondNS::x<<" "<<secondNS::y<<endl;
38     using namespace secondNS;
39     demo obx(x),oby(y);
40     cout<<"obx,oby: "<<obx.geti()<<" "<<oby.geti()<<endl;
41     return 0;
42 }
```

Namespace Can be split

```
8  #include <iostream>
9  using namespace std;
10 namespace demo{
11  int a;
12 }
13 namespace demo {
14  int b;
15 }
16 int x;
17 int main(){
18     using namespace demo;
19     a = b = x = 10;
20     cout<<a<<" "<<b<<" "<<x<<endl;
21 }
```

Namespace std

```
8  #include <iostream>
9  int main(){
10     double val;
11     std::cout<<" Enter a number: "<<std::endl;
12     std::cin>>val;
13     std::cout<<" Your number is: "<<std::endl;
14     std::cout<<val<<std::endl;
15 }
```

Namespace std

```
8  #include <iostream>
9  int main(){
10     double val;
11     std::cout<<" Enter a number: "<<std::endl;
12     std::cin>>val;
13     std::cout<<" Your number is: "<<std::endl;
14     std::cout<<val<<std::endl;
15 }
```

```
8  #include <iostream>
9  using std::cin;
10 using std::cout;
11 using std::endl;
12 int main(){
13     double val;
14     cout<<" Enter a number: "<<endl;
15     cin>>val;
16     cout<<" Your number is: "<<endl;
17     cout<<val<<endl;
18 }
```

Static Class member

```
8  #include <iostream>
9  using namespace std;
10 class myclass{
11 public:
12     static int i;
13     void seti(int x){ i = x; }
14     int geti() { return i; }
15 };
16 int myclass::i;
17 int main(){
18     myclass ob1,ob2;
19     ob1.seti(10);
20     cout<<ob1.geti()<<endl;
21     cout<<ob2.geti()<<endl;
22 }
23 |
```

10

10

Program ended with exit code: 0

Static Class member

```
8  #include <iostream>
9  using namespace std;
10 class myclass{
11 public:
12     static int i;
13     void seti(int x){ i = x; }
14     int geti() { return i; }
15 };
16 int myclass::i;
17 int main(){
18     myclass::i = 100;
19     myclass ob1,ob2;
20     cout<<ob1.geti()<<endl;
21     cout<<ob2.geti()<<endl;
22 }
23
```

100

100

Program ended with exit code: 0

Static Class member function

```
8  #include <iostream>
9  using namespace std;
10 class myclass{
11 public:
12     static int i;
13     static void seti(int x){ i = x; }
14     int geti() { return i; }
15 };
16 int myclass::i;
17 int main(){
18     myclass::seti(100);
19     myclass ob1,ob2;
20     cout<<ob1.geti()<<endl;
21     cout<<ob2.geti()<<endl;
22 }
```


Const Member function

```
8  #include <iostream>
9  using namespace std;
10 class myclass{
11 public:
12     int i;
13     void seti(int x) const{ i = x; }
14     int geti() const{ return i; }
15 };
16
17 int main(){
18     myclass ob1,ob2;
19     cout<<ob1.geti()<<endl;
20     cout<<ob2.geti()<<endl;
21 }
```



Cannot assign to non-static data member within const member function 'seti'

Const Member function & mutable

```
8  #include <iostream>
9  using namespace std;
10 class myclass{
11 public:
12     mutable int i;
13     void seti(int x) const{ i = x; }
14     int geti() const{ return i; }
15 };
16
17 int main(){
18     myclass ob1,ob2;
19     cout<<ob1.geti()<<endl;
20     cout<<ob2.geti()<<endl;
21 }
```


A last look at constructor

```
8  #include <iostream>
9  using namespace std;
10 class myclass{
11 public:
12     int i;
13     myclass(int x){
14         i = x;
15     }
16     void seti(int x) { i = x; }
17     int geti() { return i; }
18 };
19
20 int main(){
21     myclass ob1 = 4, ob2 = 5;
22     cout<<ob1.geti()<<endl;
23     cout<<ob2.geti()<<endl;
24 }
25
```

4

5

Program ended with exit code: 0

Casting operator overloading

```
8  #include <iostream>
9  using namespace std;
10 class coord{
11     int x,y;
12 public:
13     coord(int a,int b){
14         x = a;
15         y = b;
16     }
17     operator int(){
18         return x*y;
19     }
20 };
21
22 int main(){
23     coord a(10,20);
24     int i = a;
25     cout<<i<<endl;
26 }
```

200

Program ended with exit code: 0

Casting operator overloading

Printing operator overloading

```
8  #include <iostream>
9  using namespace std;
10 class coord{
11     int x,y;
12 public:
13     coord(int a,int b){
14         x = a;
15         y = b;
16     }
17     operator int(){
18         return x*y;
19     }
20     friend ostream & operator << (ostream &out, const coord &c);
21     friend istream & operator >> (istream &in, coord &c);
22 };
23 ostream & operator << (ostream &out, const coord &c)
24 {
25     out << "("<<c.x;
26     out << "," << c.y << ")" <<endl;
27     return out;
28 }
29 int main(){
30     coord a(10,20);
31     cout<<a<<endl;
32 }
```

(10,20)

Program ended with exit code: 0

Input operator overloading

```
10 class coord{
20     friend ostream & operator << (ostream &out, const coord &c);
21     friend istream & operator >> (istream &in, coord &c);
22 };
23 istream & operator >> (istream &in, coord &c)
24 {
25     cout << "Enter x :";
26     in >> c.x;
27     cout << "Enter y:";
28     in >> c.y;
29     return in;
30 }
31 ostream & operator << (ostream &out, const coord &c)
32 {
33     out << "(" << c.x;
34     out << "," << c.y << ")" << endl;
35     return out;
36 }
37 int main(){
38     coord a(0,0);
39     cin>>a;
40     cout<<a<<endl;
41 }
42
43
44
```

Enter x :10

Enter y:20

(10,20)

Program ended with exit code: 0