PA Y/4

Constructors

# class ClassName
{
Members part
public:
Test ()
{
cout << "hello good afternoon";
}
~Test ()
{
cout << "good bye";
}
};
void main ()
{
clrscr ();
Test t;
getch ();
}

destructor

O/P
hello good afternoon

Class Test
{ public:
Test (int a, int b
{
x = a;
y = b;
cout << x << y;
}
private:
int x, y;
};

```
void main ()
 {
  clrscr ();
 Test t (2,3);
 getch ();
  }
```

O/P 23

{ Rubric

#

```
void display ()
 {
 cout << x << "  " << y << endl;
 }
Test (Test &t++)
  {
   x = tt. x;
   y = tt. y;
   }};
void main ()
 {
  clrscr ()
  Test t(2,3);
  Test t1(t);
  Test t2 = t;
  t. display ();
  t1. display();
  t2. display();
  }
```

O/P

```
2 3
2 3
2 3
```

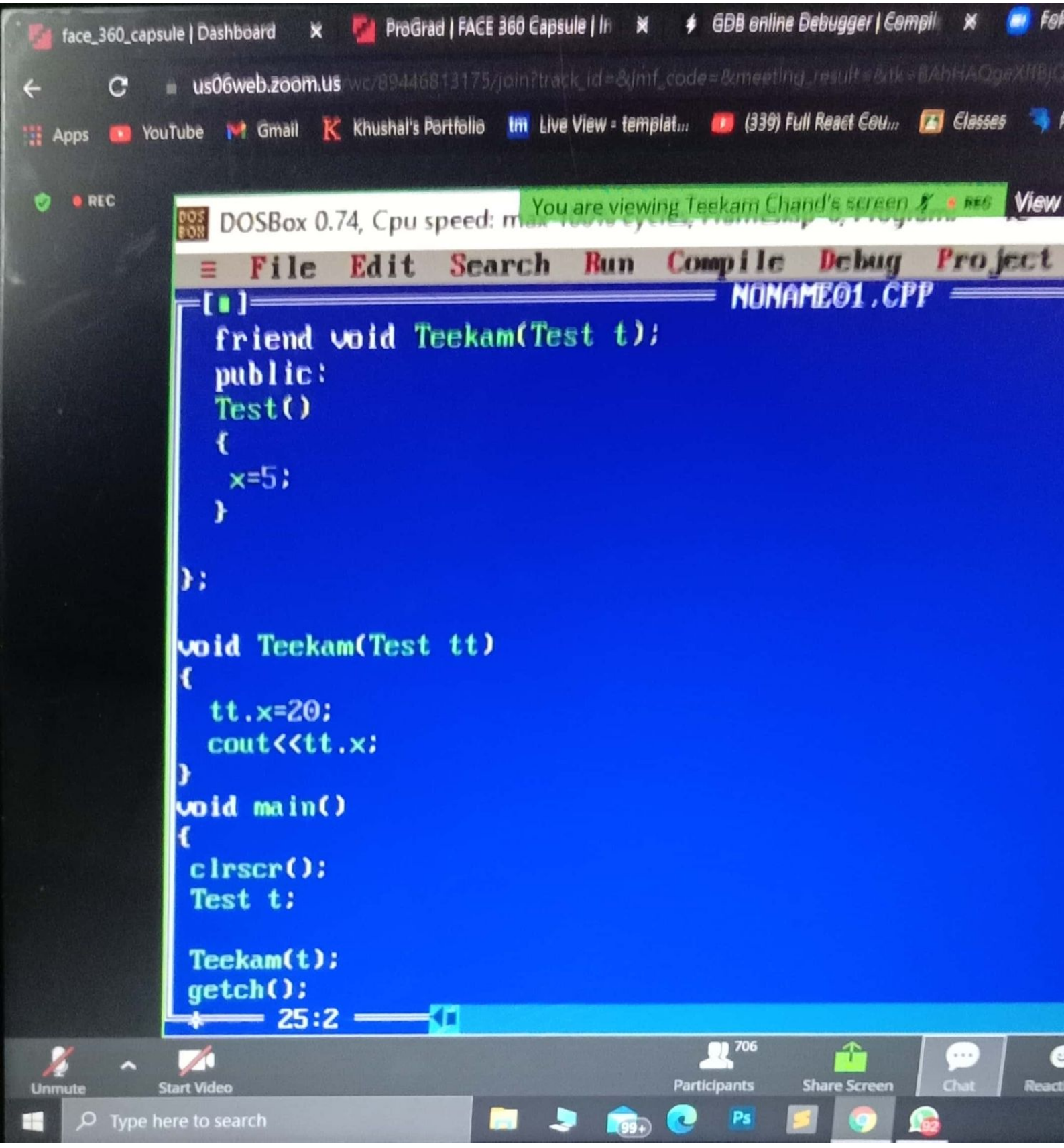
Date
Page No.

A friend function :-

```
class Test
{
private
  int x;
  friend void Teekam(Test t);
};
void Teekam(Test t)
{
  t.x = 10;
}
void main()
{
  clrscr();
  Test t;
  t.x = 5;
  getch();
}
```

friend

2 snapshot


face_360_capsule | Dashboard
ProGrad | FACE 360 Capsule | In
GDB online Debugger | Compil
us06web.zoom.us
Apps
YouTube
Gmail
Khushal's Portfolio
Live View - templat...
(339) Full React Cou...
Classes
REC
You are viewing Teekam Chand's screen
View
DOSBox 0.74, Cpu speed:
File Edit Search Run Compile Debug Project
NONAME01.CPP
friend void Teekam(Test t);
public:
Test()
{
x=5;
}
};
void Teekam(Test tt)
{
tt.x=20;
cout<<tt.x;
}
void main()
{
clrscr();
Test t;
Teekam(t);
getch();
25:2
Unmute
Start Video
706
Participants
Share Screen
Chat
Type here to search

≡ File Edit Search Run Compile Deb

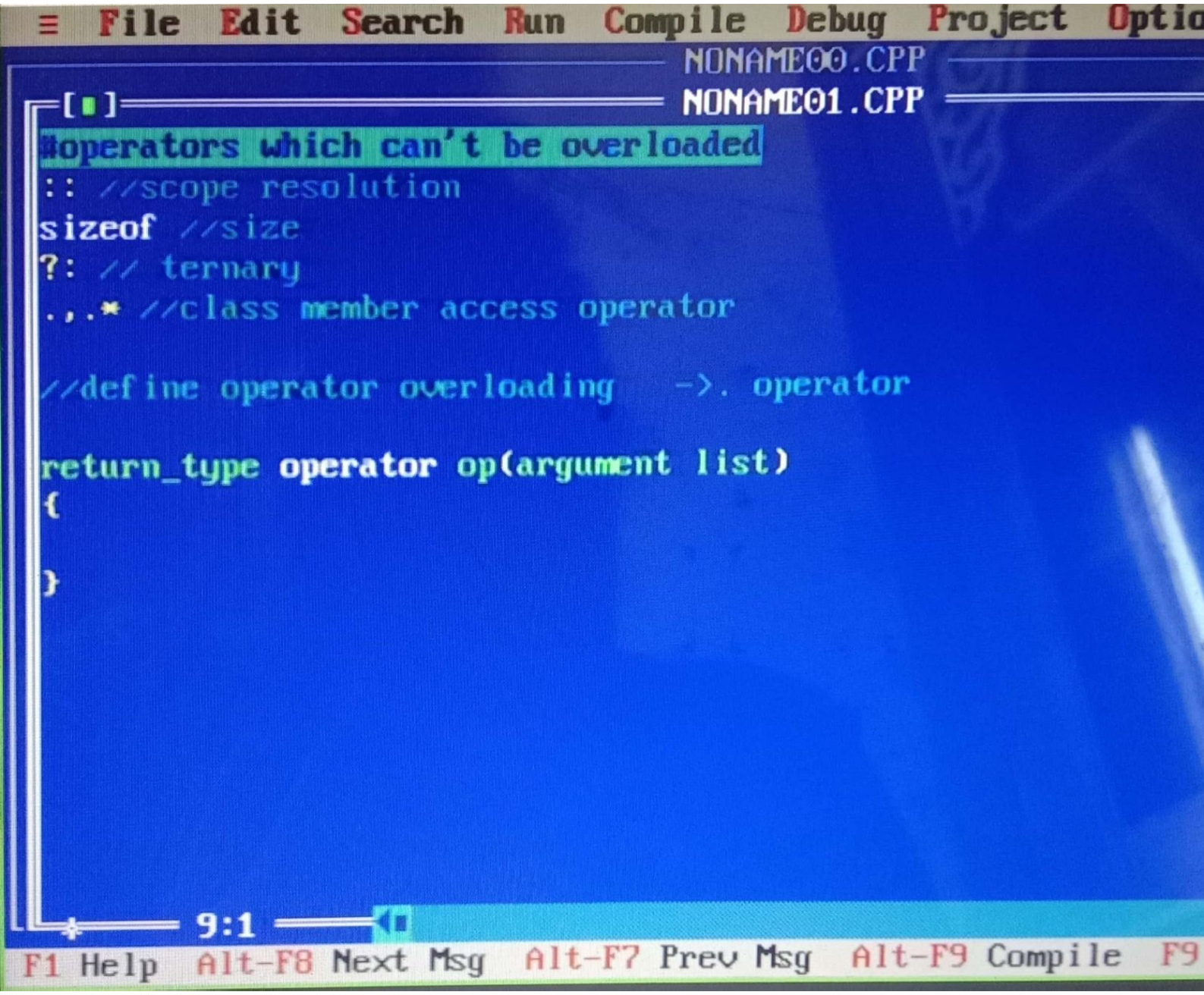
[■] NONAME00

```
#include<iostream.h>
#include<conio.h>
class TestMe
{
   int x;
  public:
  friend void Teekam(TestMe t1);
};

void Teekam(TestMe t1)
{
   t1.x=40;
   cout<<t1.x;
}
void main()
{
 clrscr();
 TestMe t;
 //t.x=20;
 Teekam(t);
getch();
```

☆ Operator which can't be overloaded

i) Scope resolution

ii)


File Edit Search Run Compile Debug Project Optio
NONAME00.CPP
NONAME01.CPP
#operators which can't be overloaded
:: //scope resolution
sizeof //size
?: // ternary
.,.* //class member access operator
//define operator overloading ->. operator
return_type operator op(argument list)
{
}
9:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9

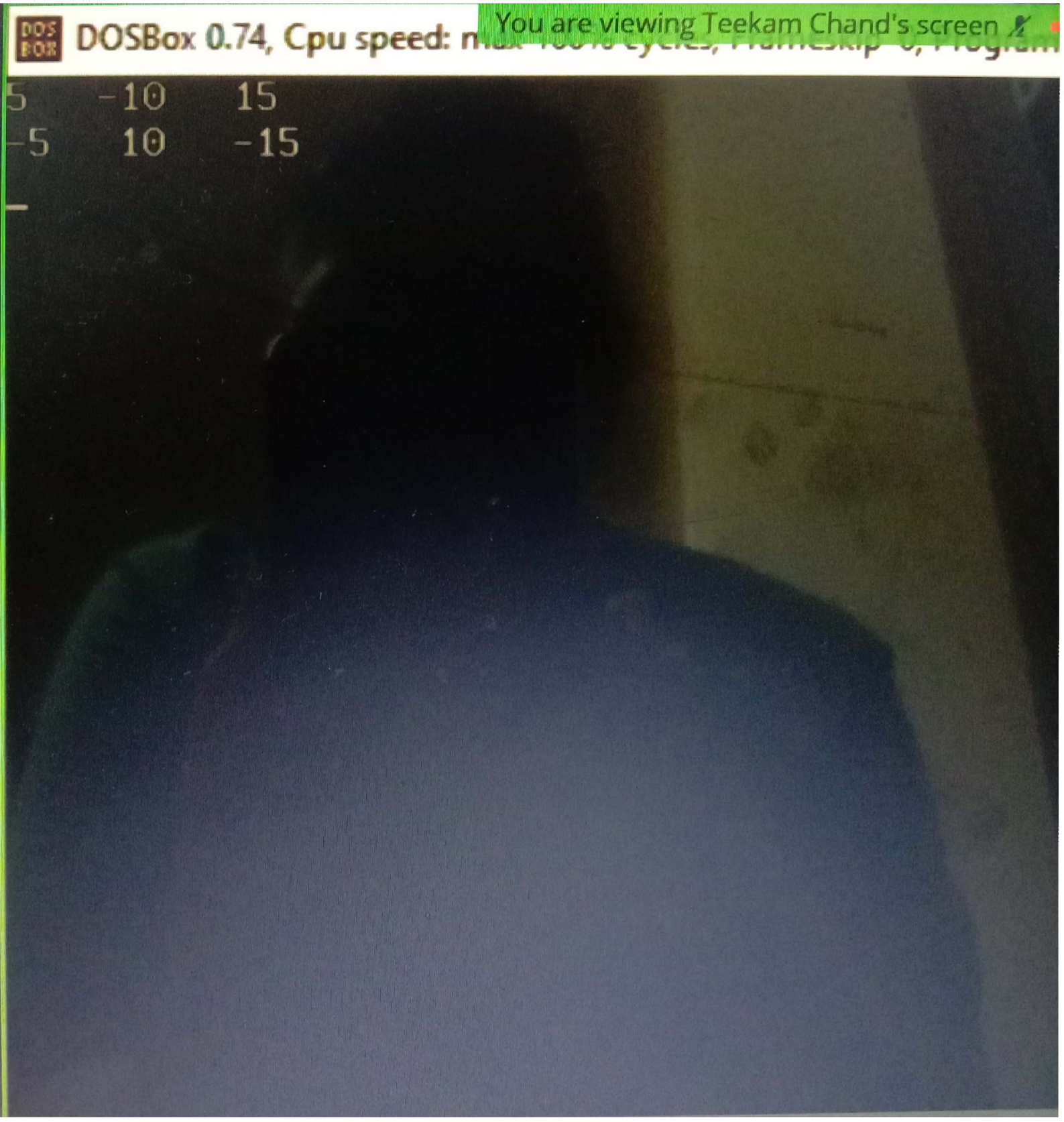
```
File  Edit  Search  Run  Compile  Debug  Proje
[■]                                NONAME01.CPP
#include<iostream.h>
#include<conio.h>
class TestMe
{
   private:
   int x,y,z;
   public:
   void getdata(int a,int b,int c);
   void display();
   void operator -();
};

void TestMe :: getdata(int a,int b,int c)
{
  x=a;
  y=b;
  z=c;
}
void TestMe :: display()
{
  cout<<x<<"   "<<y<<"   "<<z<<endl;
                                1:41
```

File Edit Search Run Compile Debug Proje

NONAME01.CPP

```
{
  cout<<x<<"   "<<y<<"   "<<z<<endl;
}

void TestMe :: operator -()
{
   x= -x;
   y= -y;
   z= -z;
}

void main()
{
 clrscr();
 TestMe t;
 t.getdata(5,-10,15);
 t.display();
 -t; //operator call
 t.display();
 getch();
}
```

37:3


DOSBox 0.74, Cpu speed:
You are viewing Teekam Chand's screen
5 -10 15
-5 10 -15

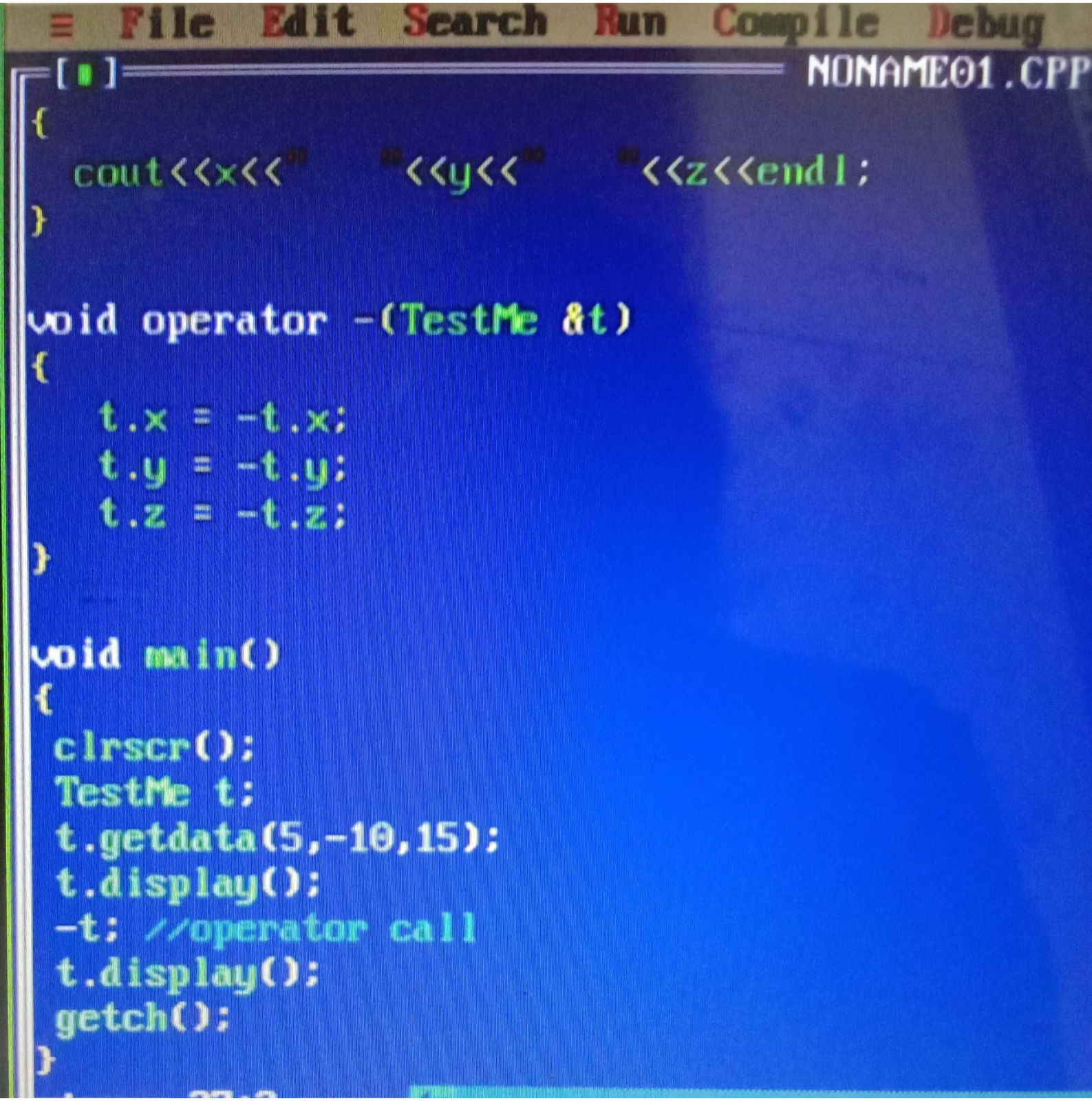
Operator overloading using friend f

File Edit Search Run Compile Debug Project

NONAME01.CPP

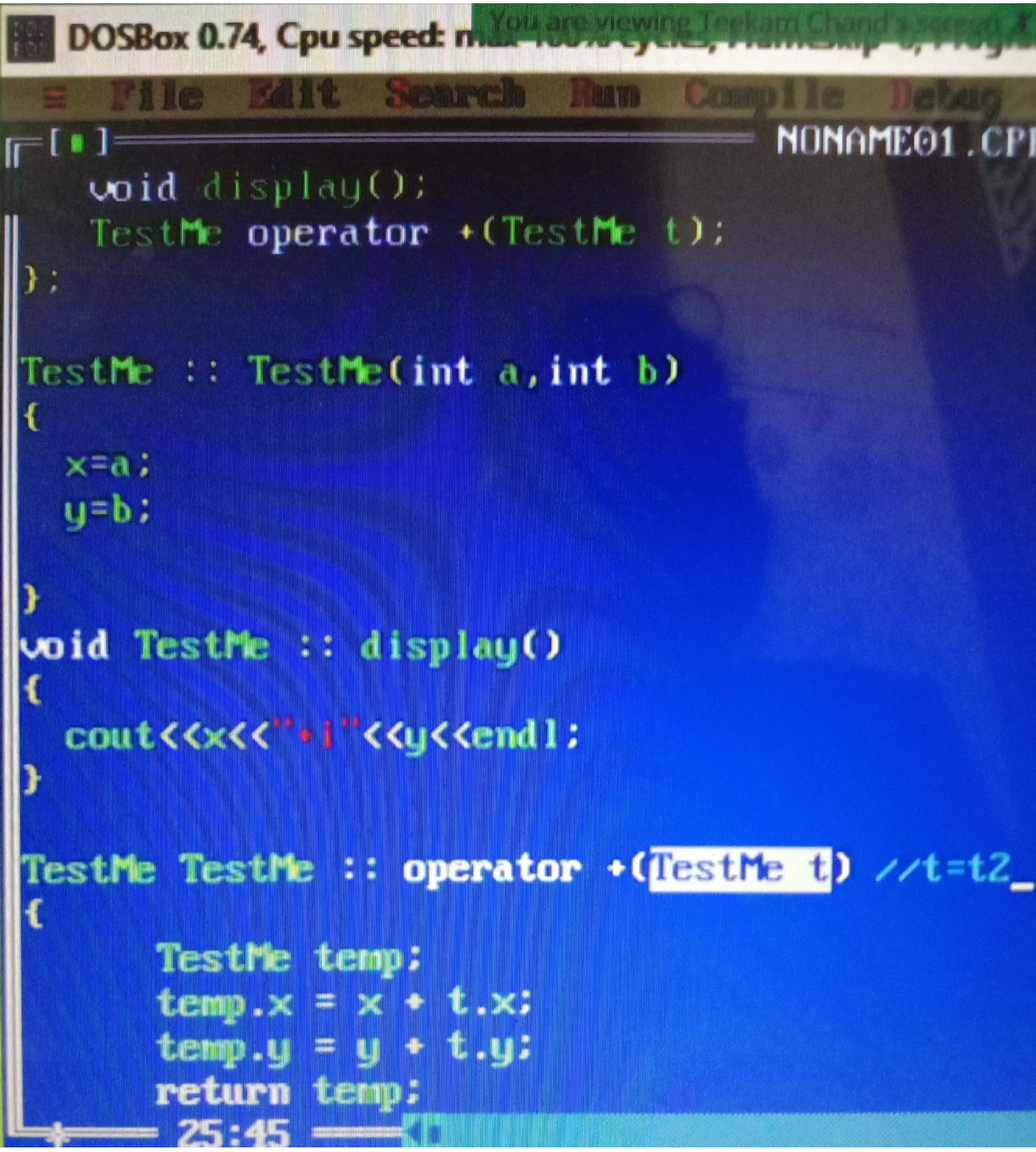
```
    public:
    void getdata(int a,int b,int c);
    void display();
    friend void operator -(TestMe &t);
};

void TestMe :: getdata(int a,int b,int c)
{
   x=a;
   y=b;
   z=c;
}
void TestMe :: display()
{
   cout<<x<<"    "<<y<<"    "<<z<<endl;
}

void operator -(TestMe &t)
{
    t.x = -t.x;
    t.y = -t.y;
```

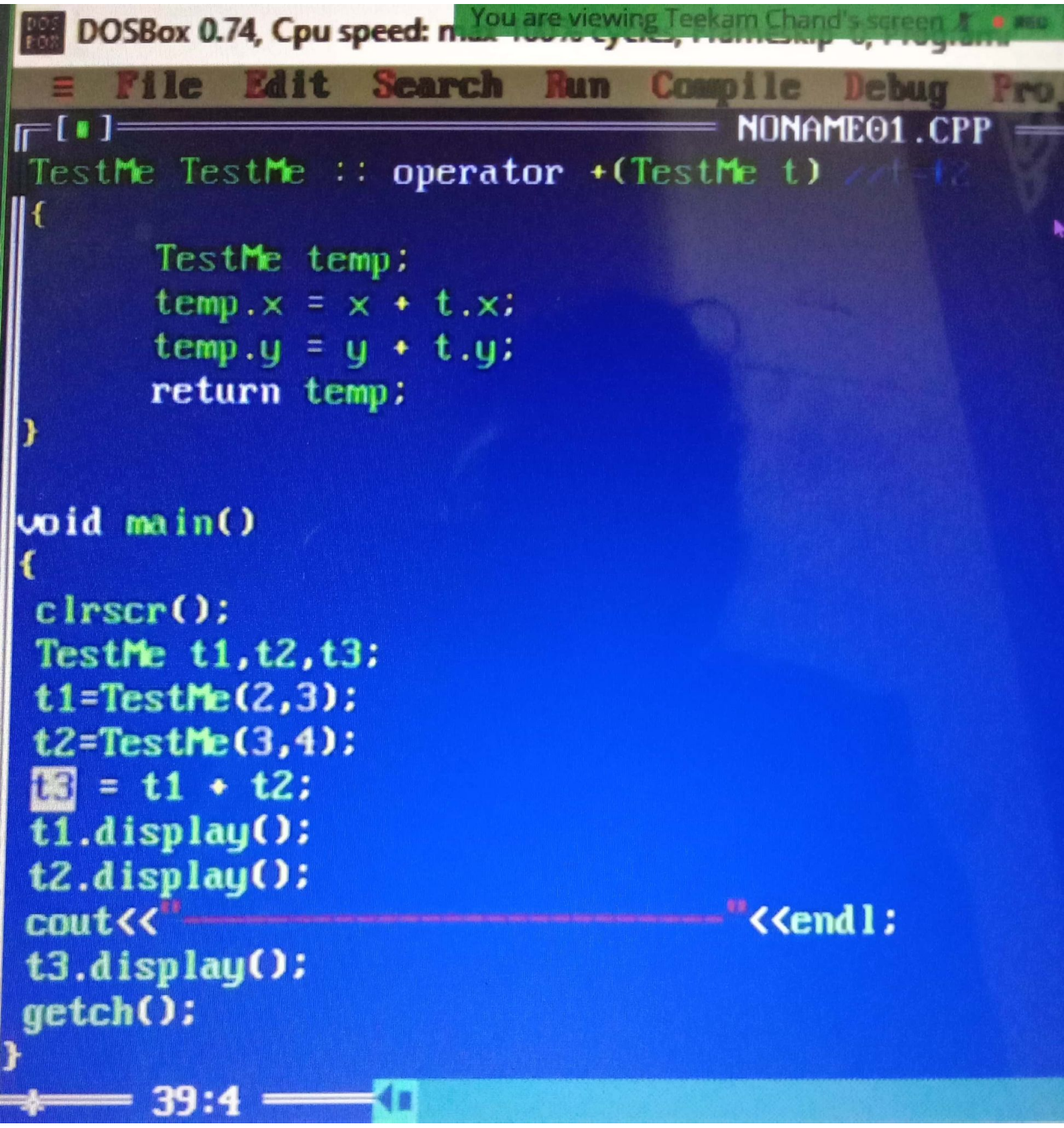
22:29

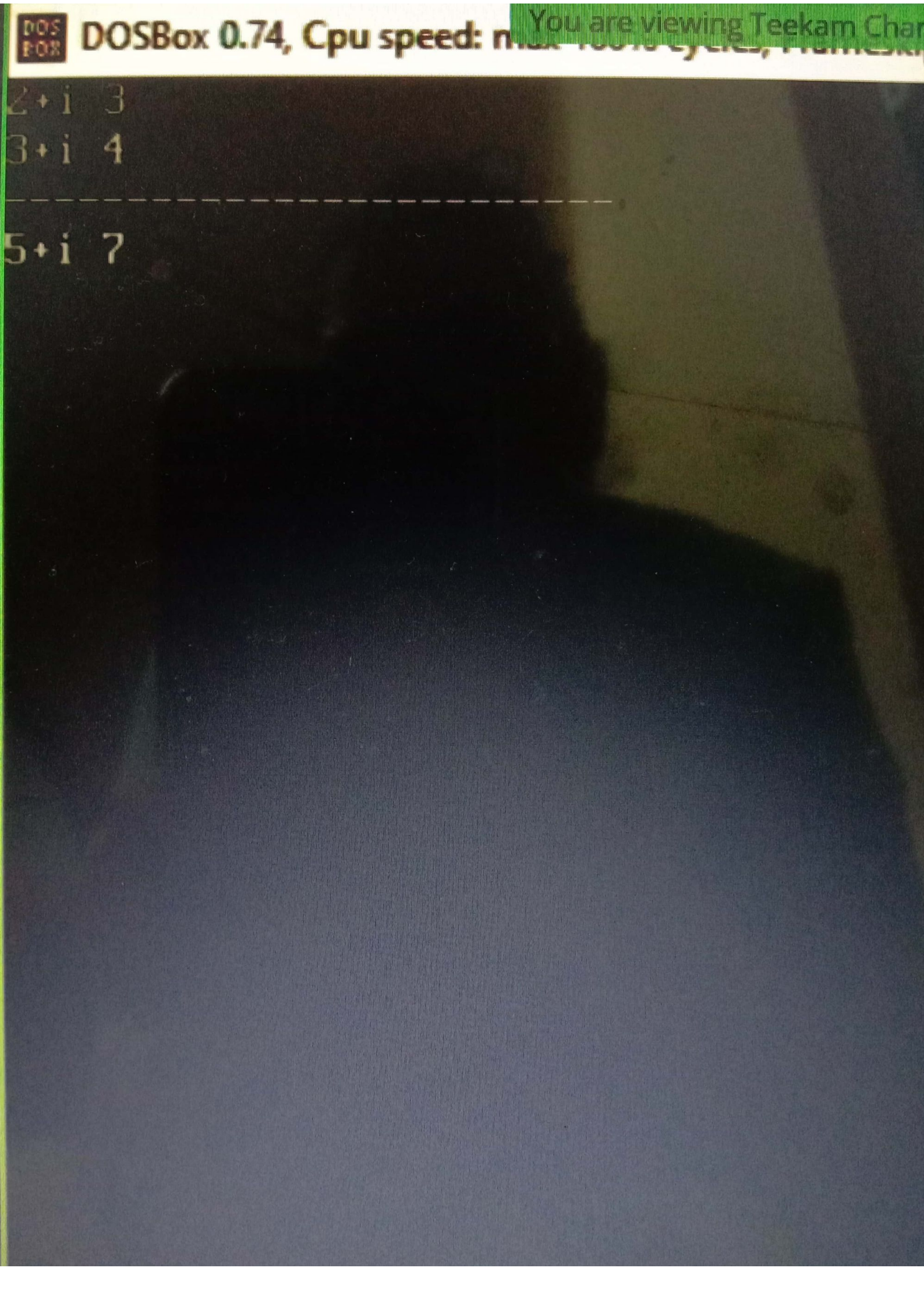

File Edit Search Run Compile Debug
NONAME01.CPP
{
cout<<x<<" "<<y<<" "<<z<<endl;
}
void operator -(TestMe &t)
{
t.x = -t.x;
t.y = -t.y;
t.z = -t.z;
}
void main()
{
clrscr();
TestMe t;
t.getdata(5,-10,15);
t.display();
-t; //operator call
t.display();
getch();
}

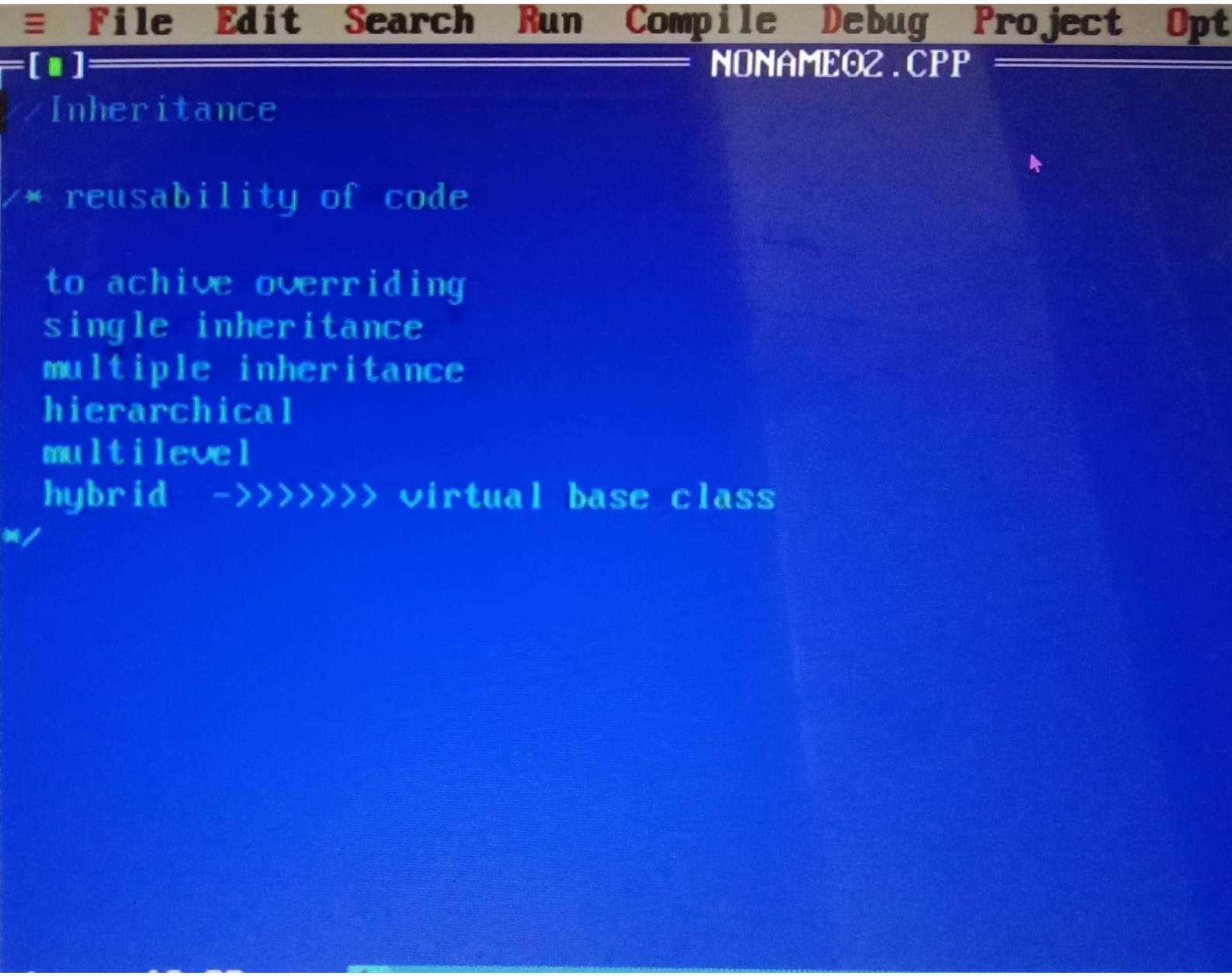
* operator overloading by binary fun

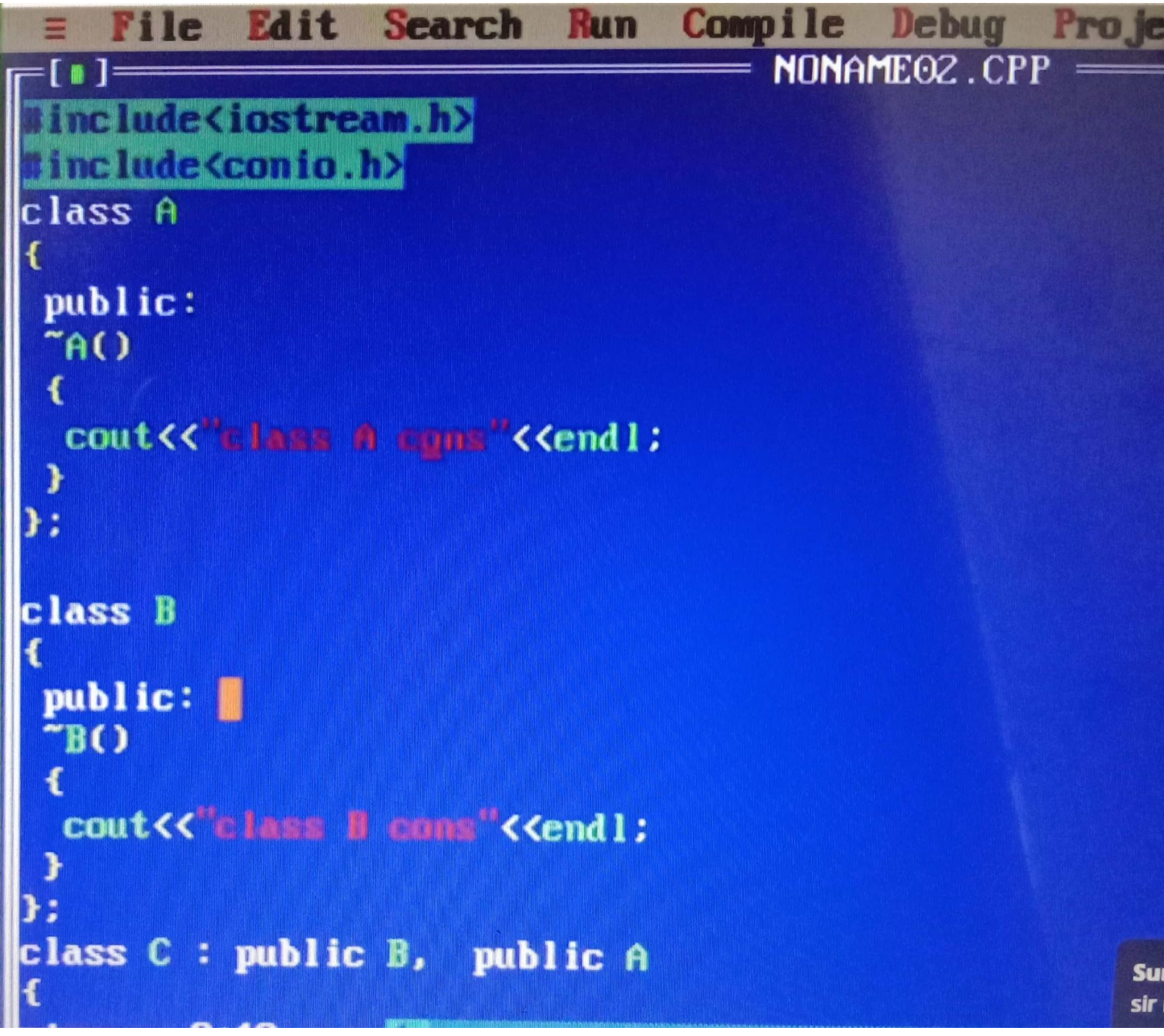
```
#include<iostream.h>
#include<conio.h>
class TestMe
{
    private:
    int x,y;
    public:
    TestMe(){}
    TestMe(int a,int b);
    void display();
    TestMe operator +(TestMe t);
};

TestMe :: TestMe(int a,int b)
{
  x=a;
  y=b;

}
void TestMe :: display()
{
```

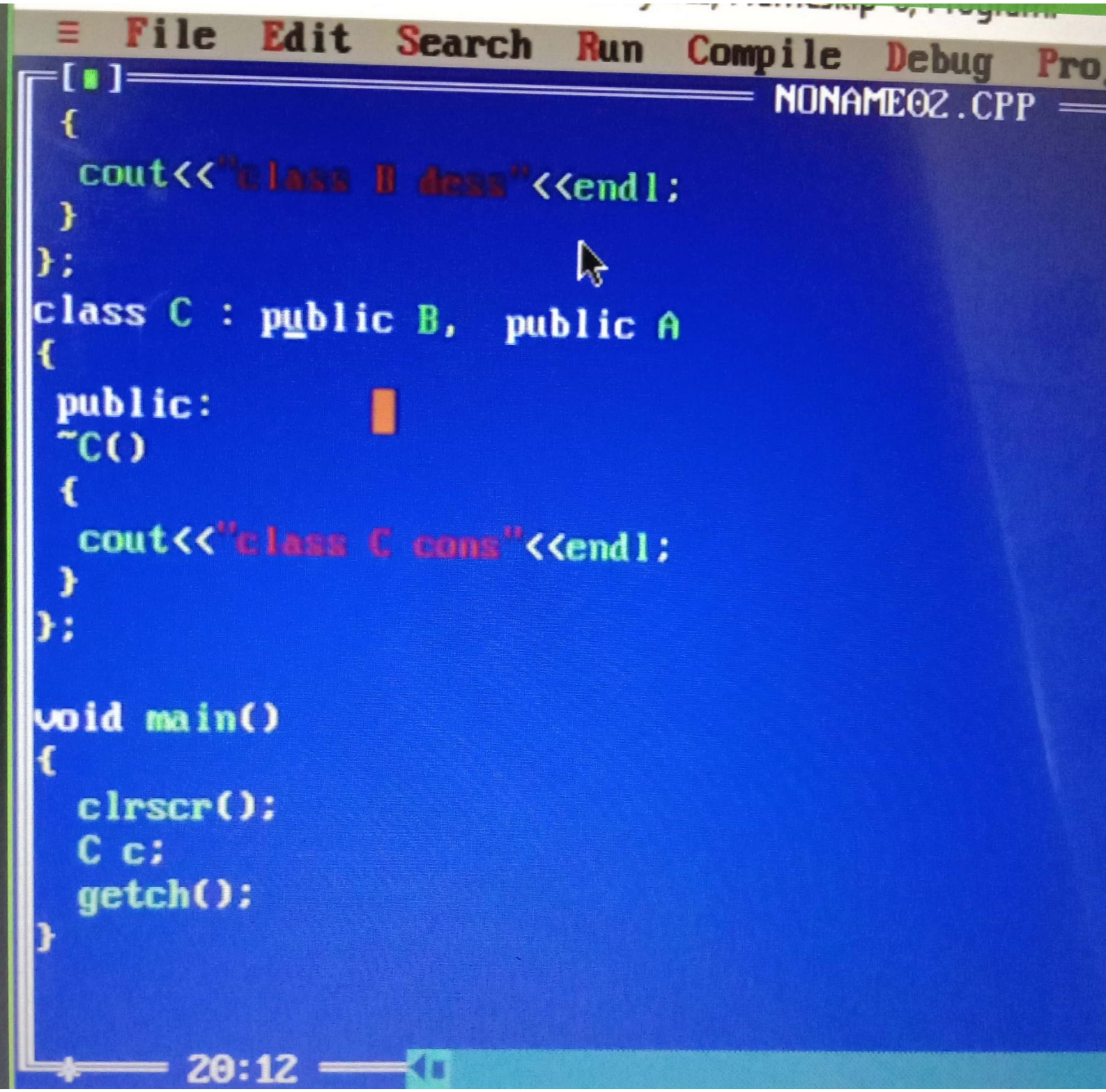

DOSBox 0.74, Cpu speed:
You are viewing Teekam Chand's screen
File Edit Search Run Compile Debug
NONAME01.CPP
void display();
TestMe operator +(TestMe t);
};
TestMe :: TestMe(int a,int b)
{
x=a;
y=b;
}
void TestMe :: display()
{
cout<<x<<"+j"<<y<<endl;
}
TestMe TestMe :: operator +(TestMe t) //t=t2_
{
TestMe temp;
temp.x = x + t.x;
temp.y = y + t.y;
return temp;
25:45


DOSBox 0.74, Cpu speed:
You are viewing Teekam Chand's screen
File Edit Search Run Compile Debug
NONAME01.CPP
TestMe TestMe :: operator +(TestMe t)
{
TestMe temp;
temp.x = x + t.x;
temp.y = y + t.y;
return temp;
}
void main()
{
clrscr();
TestMe t1,t2,t3;
t1=TestMe(2,3);
t2=TestMe(3,4);
t3 = t1 + t2;
t1.display();
t2.display();
cout<<"----------"<<endl;
t3.display();
getch();
}
39:4


DOSBox 0.74, Cpu speed:
You are viewing Teekam Char
2+i 3
3+i 4
5+i 7


File Edit Search Run Compile Debug Project Opt
NONAME02.CPP
//Inheritance
/* reusability of code
to achive overriding
single inheritance
multiple inheritance
hierarchical
multilevel
hybrid ->>>>>>> virtual base class
*/


File Edit Search Run Compile Debug Proje
NONAME02.CPP
#include<iostream.h>
#include<conio.h>
class A
{
public:
~A()
{
cout<<"class A cons"<<endl;
}
};
class B
{
public:
~B()
{
cout<<"class B cons"<<endl;
}
};
class C : public B, public A
{


File Edit Search Run Compile Debug Pro
NONAME02.CPP
{
cout<<"class B dess"<<endl;
}
};
class C : public B, public A
{
public:
~C()
{
cout<<"class C cons"<<endl;
}
};
void main()
{
clrscr();
C c;
getch();
}
20:12