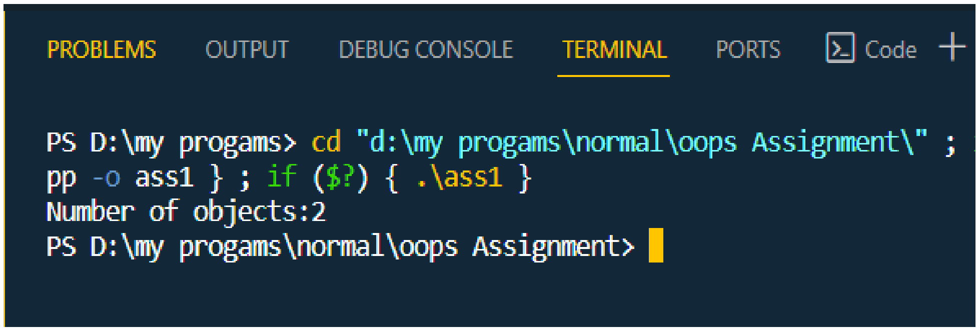
//assignment 1

```
#include<bits/stdc++.h>
using namespace std;

class Person
{
    string name;
    int age;
    static int count;

public:
    Person(string n, int a)
    {
        name = n;
        age = a;
        count++;
    }
    // function to show count of objects
    static void showCount()
    {
        cout << "Number of objects:" << count << endl;
    }
};
int Person::count = 0;

int main()
{
    Person p1("AYUSH", 21);
    Person p2("RAM", 23);
    p2.showCount();
    return 0;
}
```

Output ->

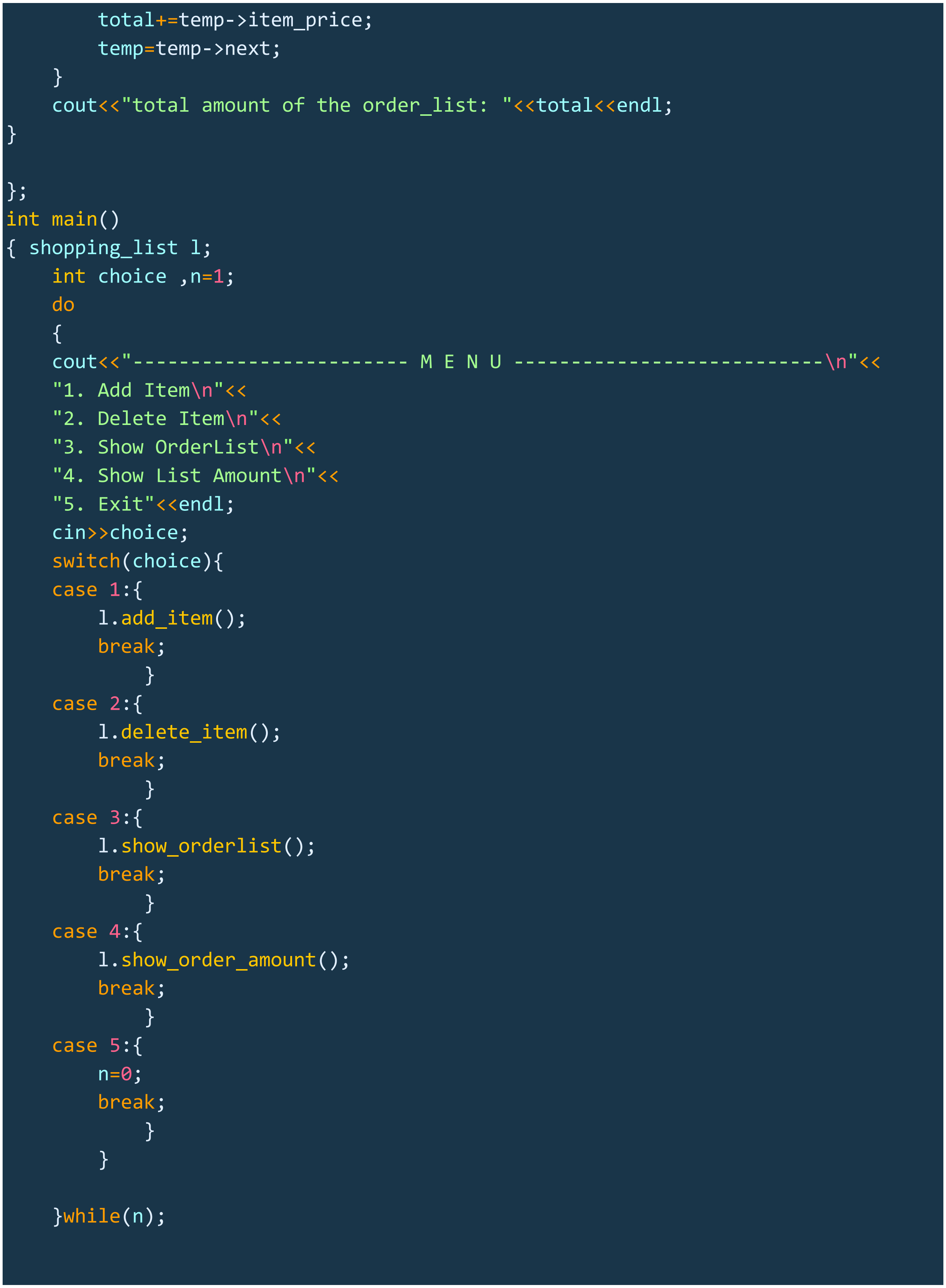


```
PS D:\my progams> cd "d:\my progams\normal\oops Assignment\" ;
pp -o ass1 } ; if ($?) { .\ass1 }
Number of objects:2
PS D:\my progams\normal\oops Assignment>
```

//assignment 2

```
#include <bits/stdc++.h>
using namespace std;
 class item
 {
   public:
   string item_name;
   int item_code;
   int item_price;
   item *next;
   item(int c,string s,int p)
   {
    item_code=c;
    item_name=s;
    item_price=p;
    next=NULL;
   }
 };
class shopping_list{
private:
item *start=NULL;
public:
void add_item()
{
    int c,p;
    string s;
    cout<<"enter the item code:"<<endl; cin>>c;
    cout<<"enter the item name:"<<endl; cin>>s;
    cout<<"enter the item price:"<<endl; cin>>p;
    item *newitem = new item(c,s,p);
    if(start==NULL)
    {
        start=newitem;
        return;
    }
   item *temp;
   temp=start;
    while(temp->next!=NULL)
        temp=temp->next;
    temp->next=newitem;
```

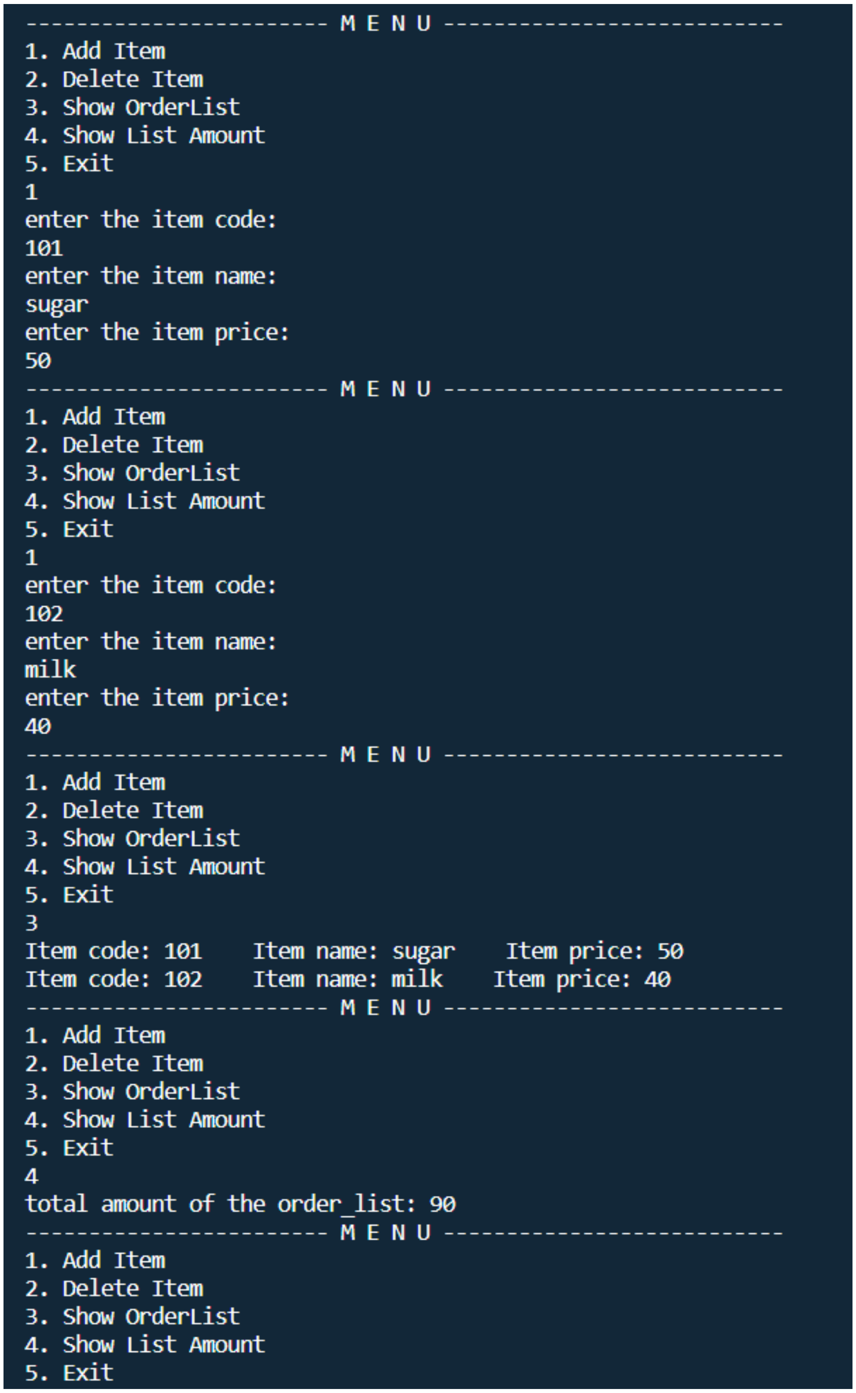
```
}
void delete_item()
{
    if(start==NULL)
    {
        cout<<"list is empty"<<endl;
        return;
    }
    int c;
    cout<<"enter item code"<<endl;
    cin>>c;
    item *temp,*prev=NULL;
    temp=start;
    while(temp!=NULL)
    {
        if(temp->item_code==c)
        break;
        prev=temp;
        temp=temp->next;
    }
    prev->next=temp->next;
    delete temp;

}
void show_orderlist()
{
    item *temp;
    temp=start;
    while(temp!=NULL)
    {
        cout<<"Item code: "<<temp->item_code<<"    ";
        cout<<"Item name: "<<temp->item_name<<"    ";
        cout<<"Item price: "<<temp->item_price<<"    ";
        cout<<endl;
        temp=temp->next;
    }

}
void show_order_amount()
{
    int total=0;
     item *temp;
    temp=start;
    while(temp!=NULL)
    {
```



```
        total+=temp->item_price;
        temp=temp->next;
    }
    cout<<"total amount of the order_list: "<<total<<endl;
}

};
int main()
{ shopping_list l;
    int choice ,n=1;
    do
    {
    cout<<"------------------------ M E N U ---------------------------\n"<<
    "1. Add Item\n"<<
    "2. Delete Item\n"<<
    "3. Show OrderList\n"<<
    "4. Show List Amount\n"<<
    "5. Exit"<<endl;
    cin>>choice;
    switch(choice){
    case 1:{
        l.add_item();
        break;
            }
    case 2:{
        l.delete_item();
        break;
            }
    case 3:{
        l.show_orderlist();
        break;
            }
    case 4:{
        l.show_order_amount();
        break;
            }
    case 5:{
        n=0;
        break;
            }
        }

    }while(n);
```

```
    return 0;
}
```

Output –>

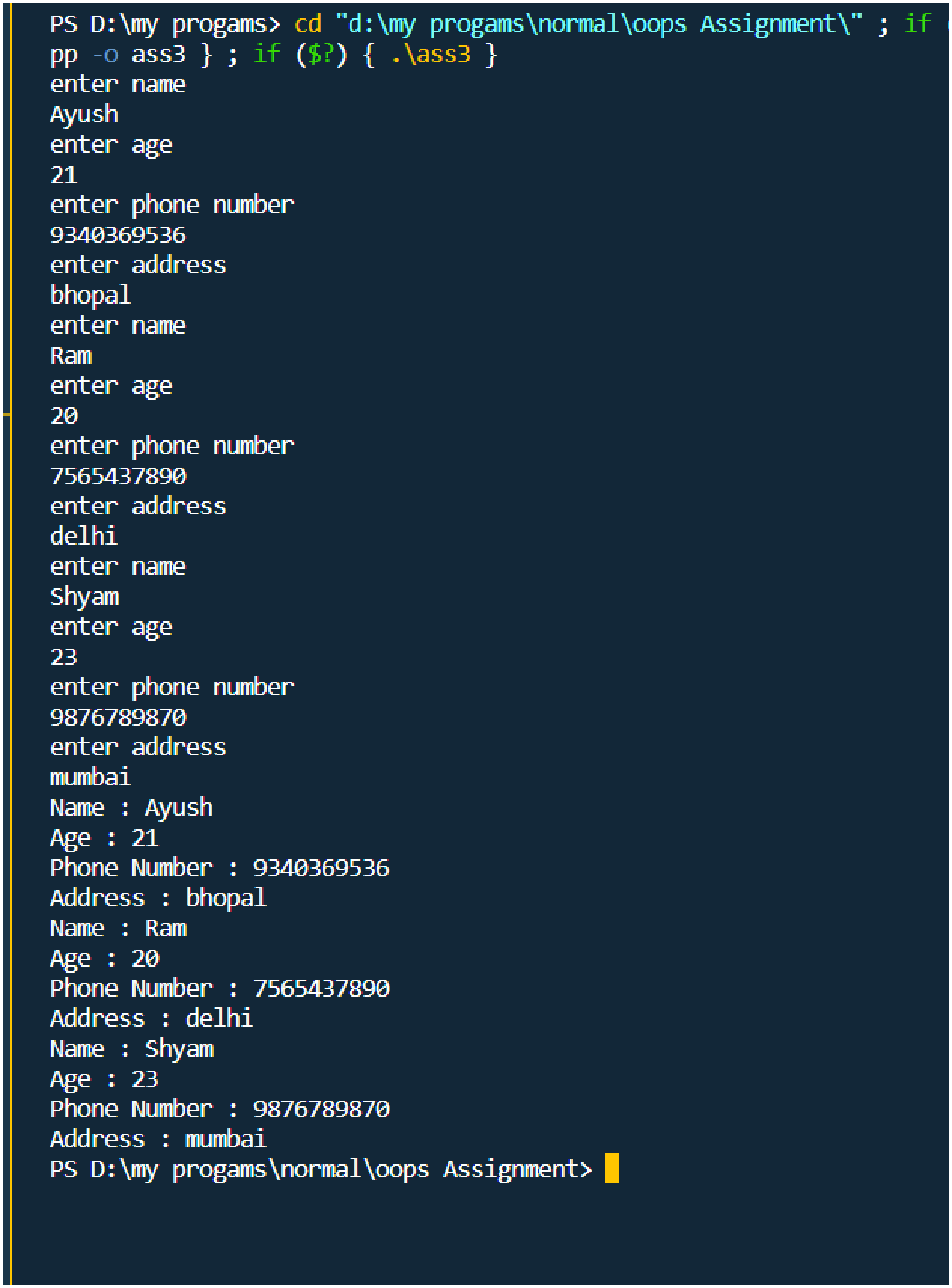


```
------------------------ M E N U ---------------------------
1. Add Item
2. Delete Item
3. Show OrderList
4. Show List Amount
5. Exit
1
enter the item code:
101
enter the item name:
sugar
enter the item price:
50
------------------------ M E N U ---------------------------
1. Add Item
2. Delete Item
3. Show OrderList
4. Show List Amount
5. Exit
1
enter the item code:
102
enter the item name:
milk
enter the item price:
40
------------------------ M E N U ---------------------------
1. Add Item
2. Delete Item
3. Show OrderList
4. Show List Amount
5. Exit
3
Item code: 101    Item name: sugar    Item price: 50
Item code: 102    Item name: milk    Item price: 40
------------------------ M E N U ---------------------------
1. Add Item
2. Delete Item
3. Show OrderList
4. Show List Amount
5. Exit
4
total amount of the order_list: 90
------------------------ M E N U ---------------------------
1. Add Item
2. Delete Item
3. Show OrderList
4. Show List Amount
5. Exit
```

//assignment 3

```
// Online C++ compiler to run C++ program online
#include <bits/stdc++.h>
using namespace std;
class manager{
    private:
    int age;
    string phone,name,address;
    public:
    void set_data()
    {
        cout<<"enter name"<<endl;
        cin>>name;
        cout<<"enter age"<<endl;
        cin>>age;
        cout<<"enter phone number"<<endl;
        cin>>phone;
        cout<<"enter address"<<endl;
        cin>>address;
    }
    void show_data()
    {
        cout<<"Name : "<<name<<endl;
        cout<<"Age : "<<age<<endl;
        cout<<"Phone Number : "<<phone<<endl;
        cout<<"Address : "<<address<<endl;
    }

};
int main() {

    manager m[3];
    for(int i=0;i<3;i++)
    {
        m[i].set_data();
    }
     for(int i=0;i<3;i++)
    {
        m[i].show_data();
    }
    return 0;
}
```

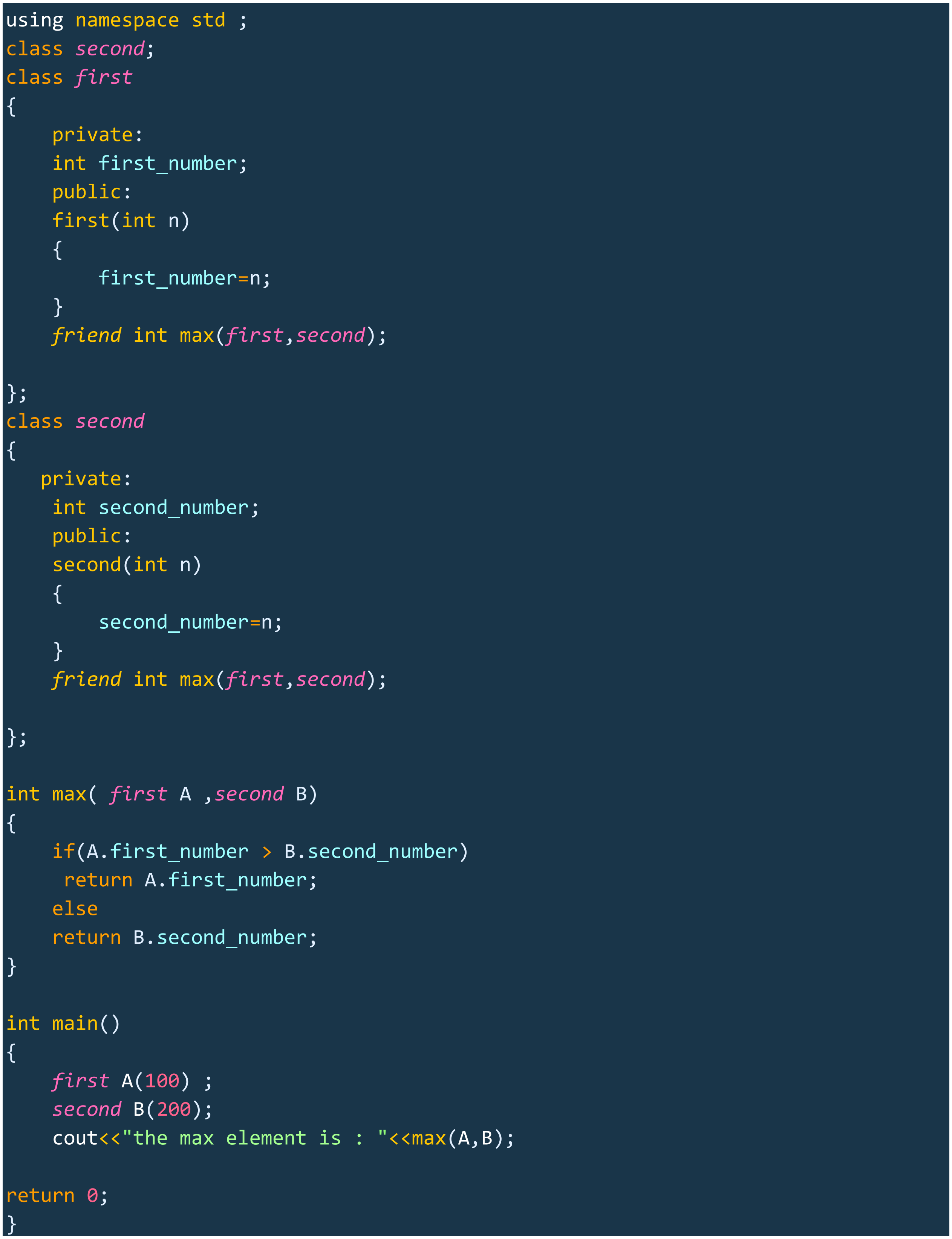
Output ->



```
PS D:\my progams> cd "d:\my progams\normal\oops Assignment\" ; if
pp -o ass3 } ; if ($?) { .\ass3 }
enter name
Ayush
enter age
21
enter phone number
9340369536
enter address
bhopal
enter name
Ram
enter age
20
enter phone number
7565437890
enter address
delhi
enter name
Shyam
enter age
23
enter phone number
9876789870
enter address
mumbai
Name : Ayush
Age : 21
Phone Number : 9340369536
Address : bhopal
Name : Ram
Age : 20
Phone Number : 7565437890
Address : delhi
Name : Shyam
Age : 23
Phone Number : 9876789870
Address : mumbai
PS D:\my progams\normal\oops Assignment>
```

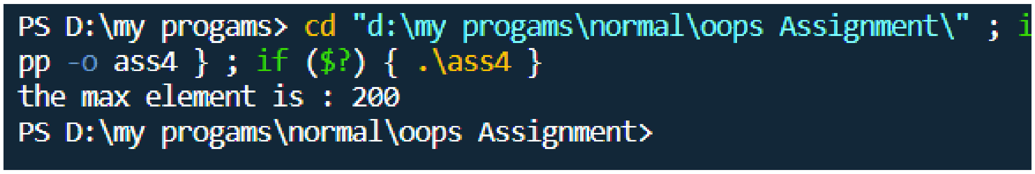
//assignment 4

```
#include<bits/stdc++.h>
```



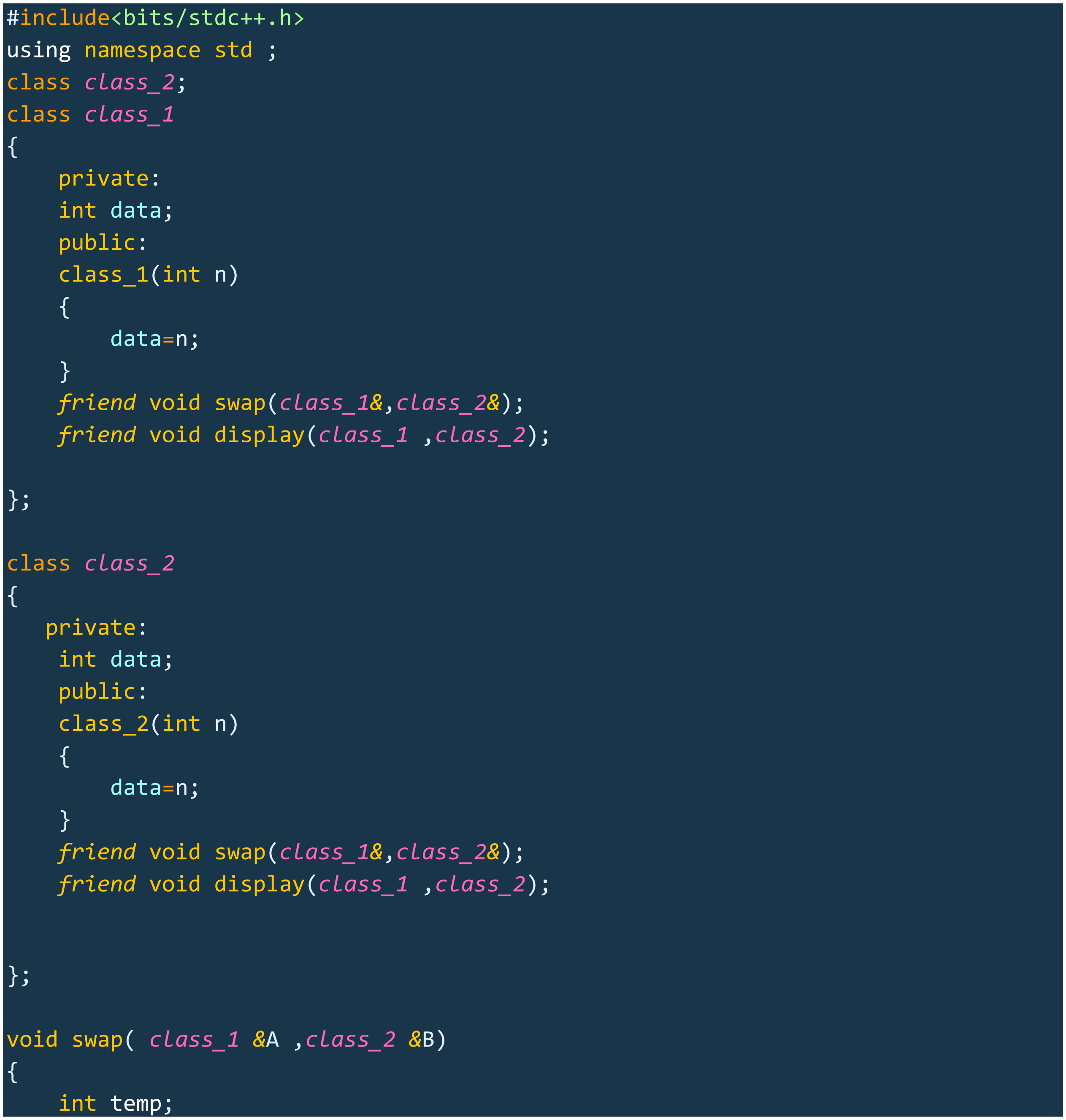
```
using namespace std ;
class second;
class first
{
    private:
    int first_number;
    public:
    first(int n)
    {
        first_number=n;
    }
    friend int max(first,second);

};
class second
{
   private:
    int second_number;
    public:
    second(int n)
    {
        second_number=n;
    }
    friend int max(first,second);

};

int max( first A ,second B)
{
    if(A.first_number > B.second_number)
     return A.first_number;
    else
    return B.second_number;
}

int main()
{
    first A(100) ;
    second B(200);
    cout<<"the max element is : "<<max(A,B);

return 0;
}
```

Output ->



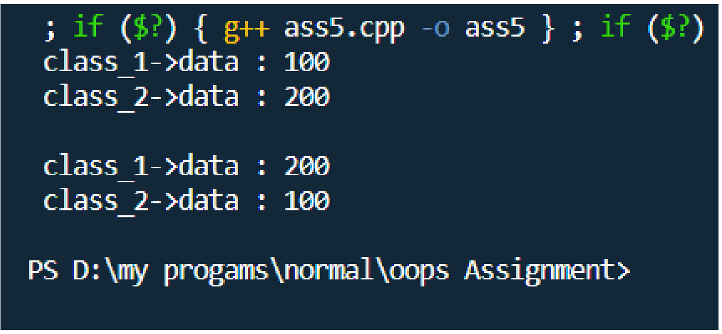
```
PS D:\my progams> cd "d:\my progams\normal\oops Assignment\" ; i
pp -o ass4 } ; if ($?) { .\ass4 }
the max element is : 200
PS D:\my progams\normal\oops Assignment>
```

//assignment 5



```
#include<bits/stdc++.h>
using namespace std ;
class class_2;
class class_1
{
    private:
    int data;
    public:
    class_1(int n)
    {
        data=n;
    }
    friend void swap(class_1&,class_2&);
    friend void display(class_1 ,class_2);

};

class class_2
{
   private:
    int data;
    public:
    class_2(int n)
    {
        data=n;
    }
    friend void swap(class_1&,class_2&);
    friend void display(class_1 ,class_2);


};

void swap( class_1 &A ,class_2 &B)
{
    int temp;
    temp=A.data;
    A.data=B.data;
    B.data=temp;
}
void display(class_1 A,class_2 B)
{
    cout<<" class_1->data : "<<A.data<<endl;
    cout<<" class_2->data : "<<B.data<<endl;
    cout<<endl;
}

int main()
{
    class_1 A(100) ;
    class_2 B(200);
    display(A,B);
    swap(A,B);
    display(A,B);


return 0;
}
```

Output –

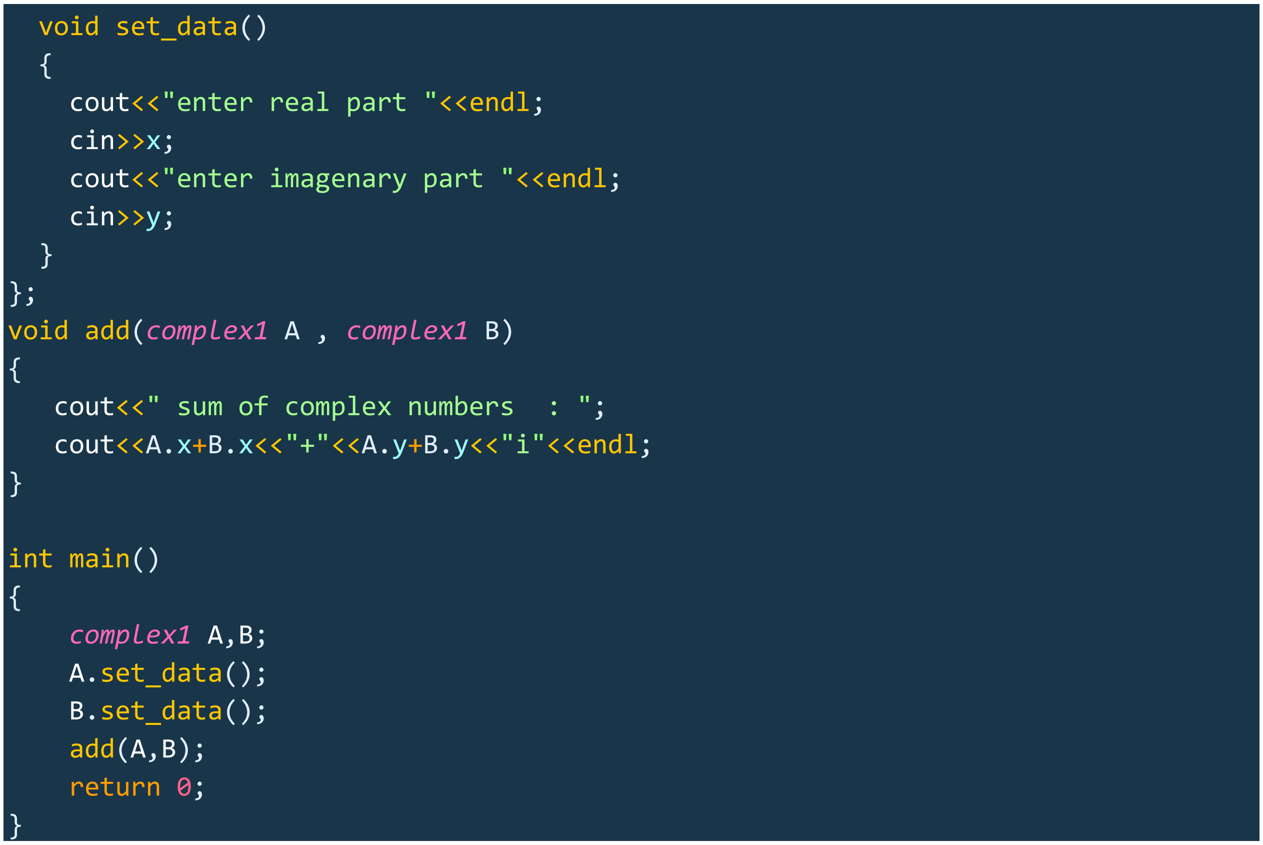


```
  ; if ($?) { g++ ass5.cpp -o ass5 } ; if ($?)
  class_1->data : 100
  class_2->data : 200

  class_1->data : 200
  class_2->data : 100

 PS D:\my progams\normal\oops Assignment>
```

>

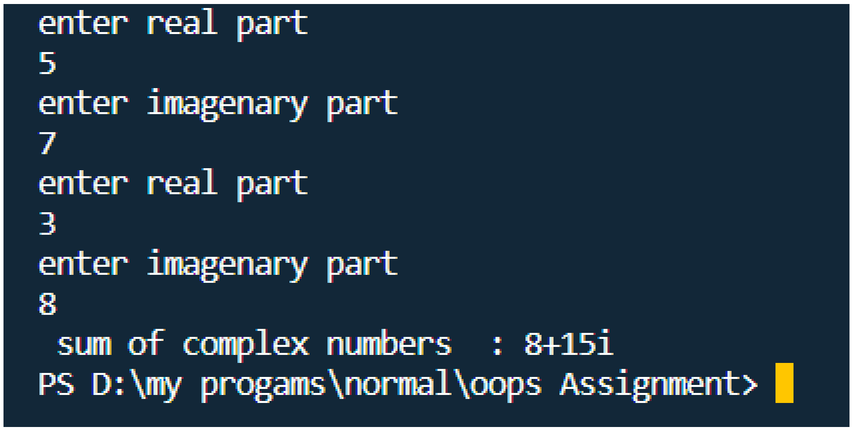
//assignment 6

```
#include<bits/stdc++.h>
using namespace std ;

 class complex1
{
  int x,y;
  public:

  friend void add(complex1 , complex1 );
```



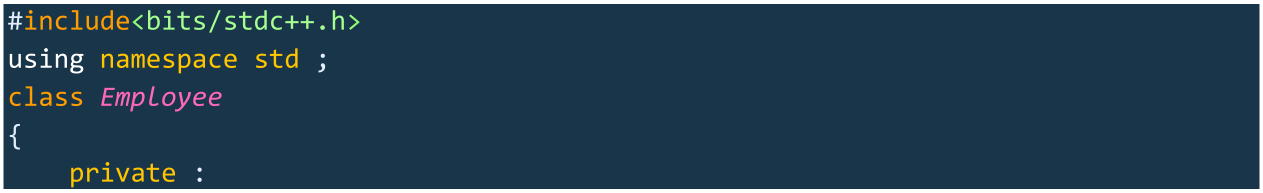
```
  void set_data()
  {
    cout<<"enter real part "<<endl;
    cin>>x;
    cout<<"enter imagenary part "<<endl;
    cin>>y;
  }
};
void add(complex1 A , complex1 B)
{
   cout<<" sum of complex numbers  : ";
   cout<<A.x+B.x<<"+"<<A.y+B.y<<"i"<<endl;
}

int main()
{
    complex1 A,B;
    A.set_data();
    B.set_data();
    add(A,B);
    return 0;
}
```

Output->

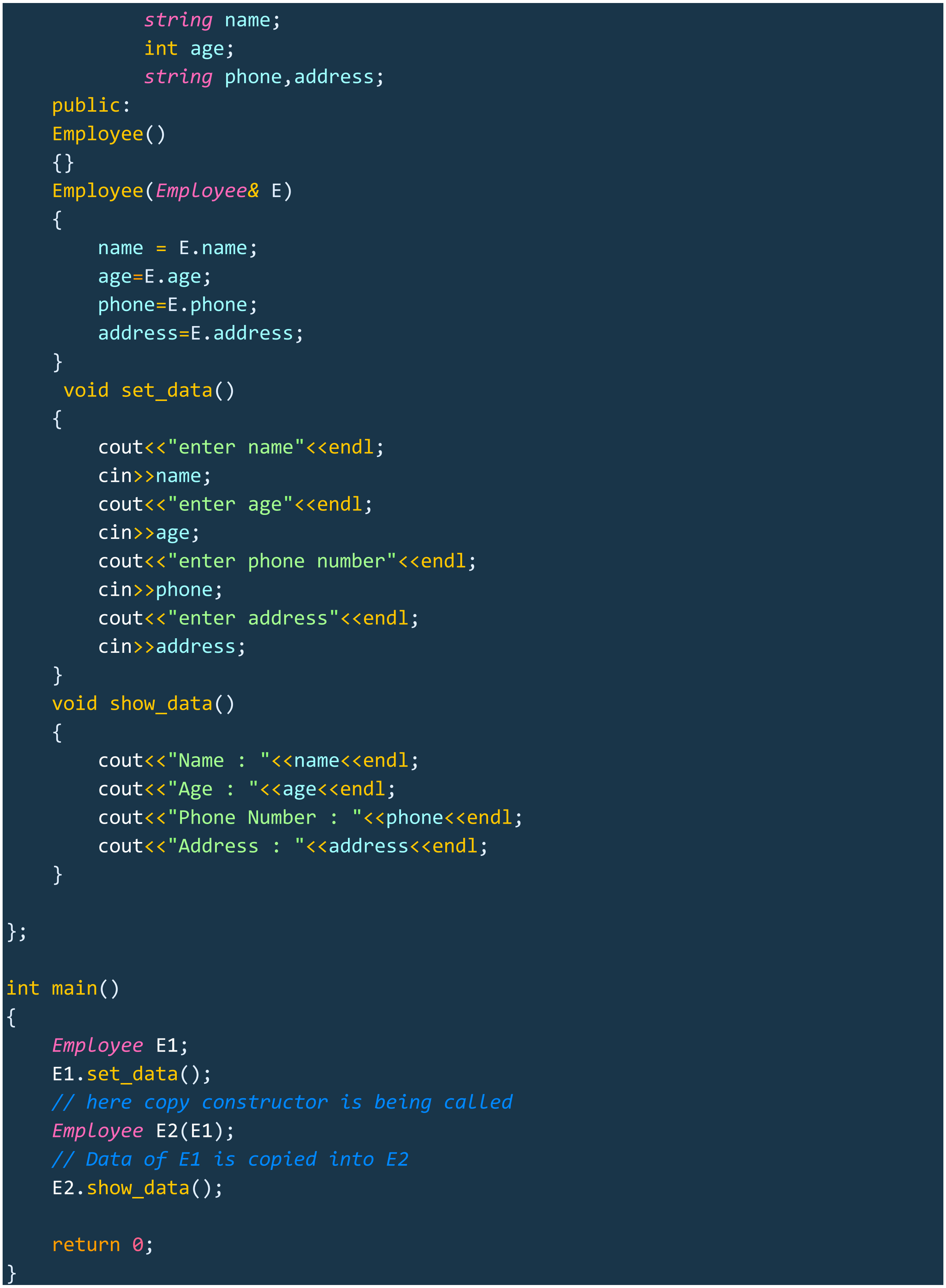


```
enter real part
5
enter imagenary part
7
enter real part
3
enter imagenary part
8
 sum of complex numbers  : 8+15i
PS D:\my progams\normal\oops Assignment> 
```

//assignment 7

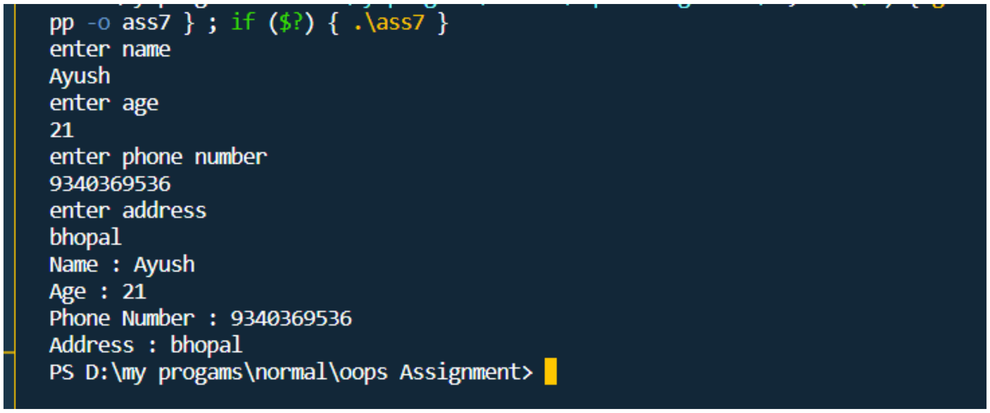


```
#include<bits/stdc++.h>
using namespace std ;
class Employee
{
    private :
```



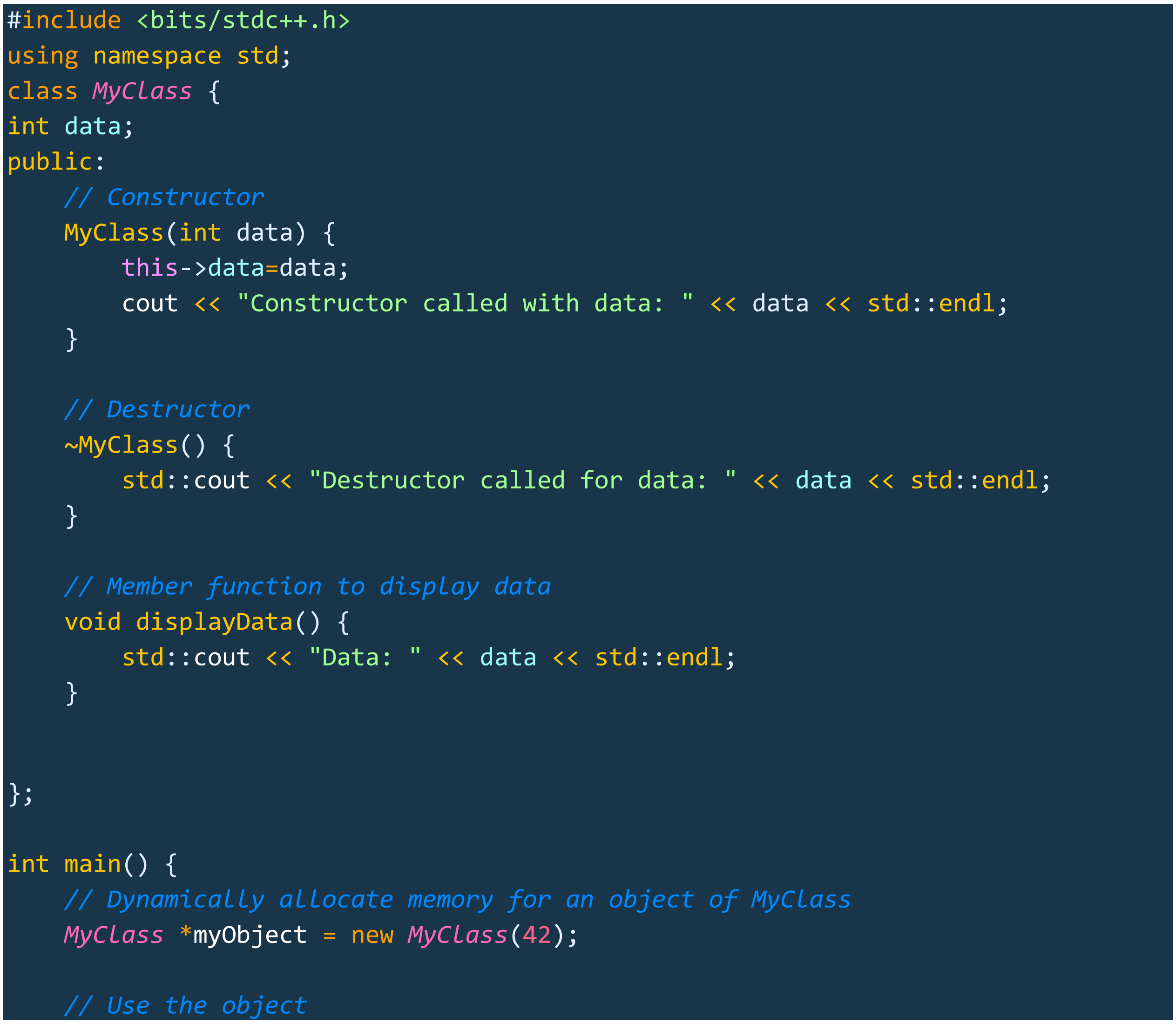
```
            string name;
            int age;
            string phone,address;
    public:
    Employee()
    {}
    Employee(Employee& E)
    {
        name = E.name;
        age=E.age;
        phone=E.phone;
        address=E.address;
    }
     void set_data()
    {
        cout<<"enter name"<<endl;
        cin>>name;
        cout<<"enter age"<<endl;
        cin>>age;
        cout<<"enter phone number"<<endl;
        cin>>phone;
        cout<<"enter address"<<endl;
        cin>>address;
    }
    void show_data()
    {
        cout<<"Name : "<<name<<endl;
        cout<<"Age : "<<age<<endl;
        cout<<"Phone Number : "<<phone<<endl;
        cout<<"Address : "<<address<<endl;
    }

};

int main()
{
    Employee E1;
    E1.set_data();
    // here copy constructor is being called
    Employee E2(E1);
    // Data of E1 is copied into E2
    E2.show_data();

    return 0;
}
```

Output->



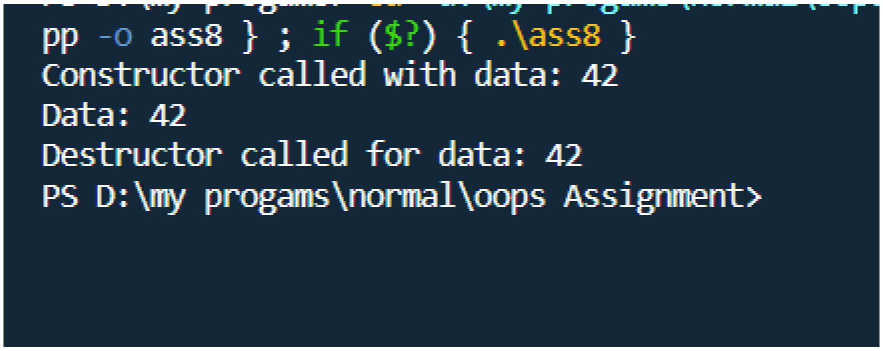
```
pp -o ass7 } ; if ($?) { .\ass7 }
enter name
Ayush
enter age
21
enter phone number
9340369536
enter address
bhopal
Name : Ayush
Age : 21
Phone Number : 9340369536
Address : bhopal
PS D:\my progams\normal\oops Assignment>
```

//assignment 8



```
#include <bits/stdc++.h>
using namespace std;
class MyClass {
int data;
public:
    // Constructor
    MyClass(int data) {
        this->data=data;
        cout << "Constructor called with data: " << data << std::endl;
    }

    // Destructor
    ~MyClass() {
        std::cout << "Destructor called for data: " << data << std::endl;
    }

    // Member function to display data
    void displayData() {
        std::cout << "Data: " << data << std::endl;
    }


};

int main() {
    // Dynamically allocate memory for an object of MyClass
    MyClass *myObject = new MyClass(42);

    // Use the object
    myObject->displayData();

    // Deallocate the memory when done using the object
    delete myObject;

    return 0;
}
```

Output->

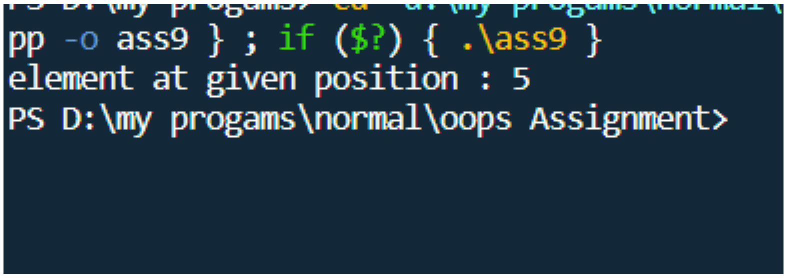


```
pp -o ass8 } ; if ($?) { .\ass8 }
Constructor called with data: 42
Data: 42
Destructor called for data: 42
PS D:\my progams\normal\oops Assignment>
```

//assignment 9

```
#include <bits/stdc++.h>
using namespace std;
#define MAX 10

class Matrix
{
    int arr[MAX][MAX];

public:
    Matrix(int row, int col)
    {
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                arr[i][j] = 0;
            }
        }
    }

    void insert(int row, int col, int val)
    {
        if (row < 0 || row >= MAX || col < 0 || col >= MAX)
        {
            return;
        }
        arr[row][col] = val;
    }

    int get(int row, int col)
    {
        if (row < 0 || row >= MAX || col < 0 || col >= MAX)
        {
            return -1;
        }
        return arr[row][col];
    }
};

int main()
{
    Matrix obj(3, 3);
    obj.insert(0, 0, 1);
    obj.insert(0, 1, 2);
    obj.insert(0, 2, 3);
    obj.insert(1, 0, 4);
    obj.insert(1, 1, 5);
    cout<<"element at given position : "<<obj.get(1, 1);
    return 0;
}
```

Output->

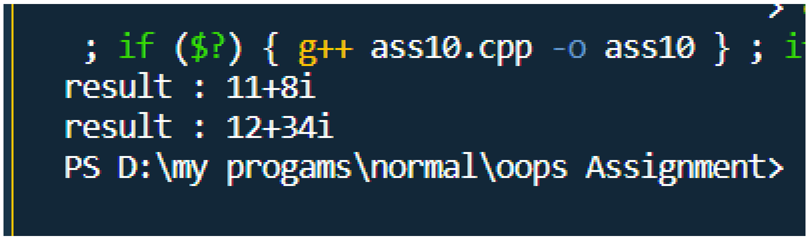


```
PS D:\my progams> cd "d:\my progams\normal\
pp -o ass9 } ; if ($?) { .\ass9 }
element at given position : 5
PS D:\my progams\normal\oops Assignment>
```

//assignment 10

```
#include <iostream>
using namespace std;

class Copmlex
{
    int real;
    int img;

public:
    Copmlex(int r = 0, int i = 0)
    {
        real = r;
        img = i;
    }
    Copmlex operator+(Copmlex obj)
    {
        Copmlex temp;
        temp.real = real + obj.real;
        temp.img = img + obj.img;
        return temp;
    }
    Copmlex operator*(Copmlex obj)
    {
        Copmlex temp;
        temp.real = real * obj.real - img * obj.img;
        temp.img = real * obj.img + img * obj.real;
        return temp;
    }
    void display()
    {
        cout<<"result : " << real << "+" << img << "i" << endl;
    }
};

int main()
{
    Copmlex c1(8, 6), c2(3, 2);
    Copmlex c3 = c1 + c2;
    c3.display();
    Copmlex c4 = c1 * c2;
    c4.display();
    return 0;
}
```

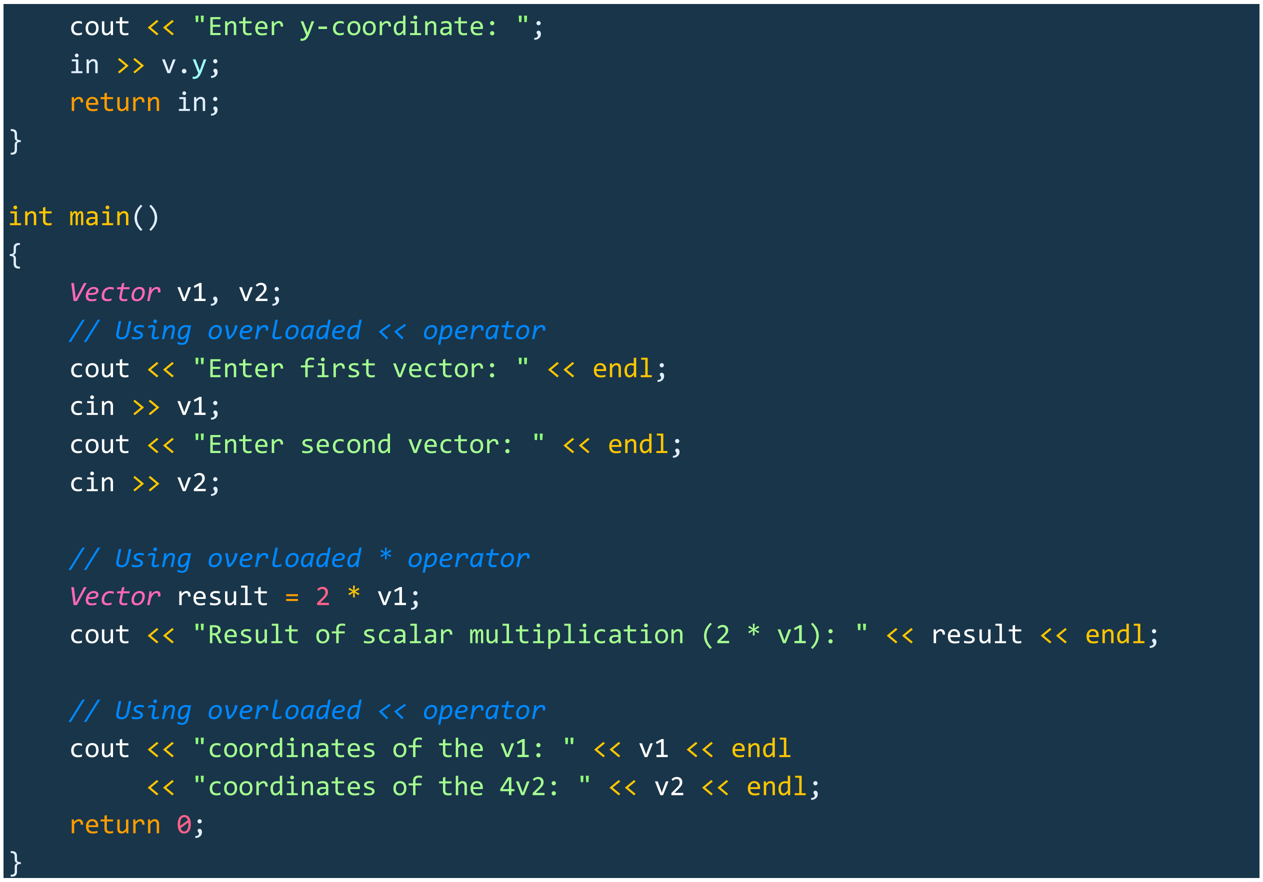
Output->



```
 ; if ($?) { g++ ass10.cpp -o ass10 } ; i
result : 11+8i
result : 12+34i
PS D:\my progams\normal\oops Assignment>
```

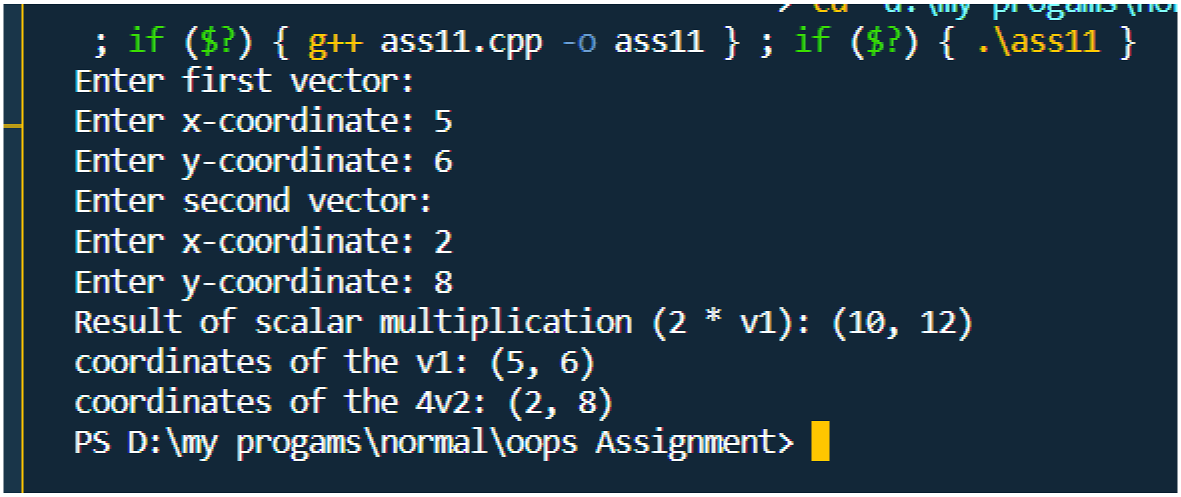
//assignment 11

```
#include <iostream>
using namespace std;

class Vector
{
    int x, y;

public:
    Vector() {}
    Vector(int a, int b)
    {
        x = a;
        y = b;
    }

    friend Vector operator*(int, Vector &);
    friend ostream &operator<<(ostream &, Vector &);
    friend istream &operator>>(istream &, Vector &);
};

Vector operator*(int a, Vector &v)
{
    return Vector(a * v.x, a * v.y);
}

ostream &operator<<(ostream &out, Vector &v)
{
    out << "(" << v.x << ", " << v.y << ")";
    return out;
}

istream &operator>>(istream &in, Vector &v)
{
    cout << "Enter x-coordinate: ";
    in >> v.x;
```



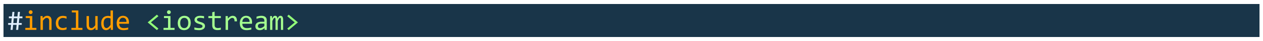
```
    cout << "Enter y-coordinate: ";
    in >> v.y;
    return in;
}

int main()
{
    Vector v1, v2;
    // Using overloaded << operator
    cout << "Enter first vector: " << endl;
    cin >> v1;
    cout << "Enter second vector: " << endl;
    cin >> v2;

    // Using overloaded * operator
    Vector result = 2 * v1;
    cout << "Result of scalar multiplication (2 * v1): " << result << endl;

    // Using overloaded << operator
    cout << "coordinates of the v1: " << v1 << endl
         << "coordinates of the 4v2: " << v2 << endl;
    return 0;
}
```

Output->

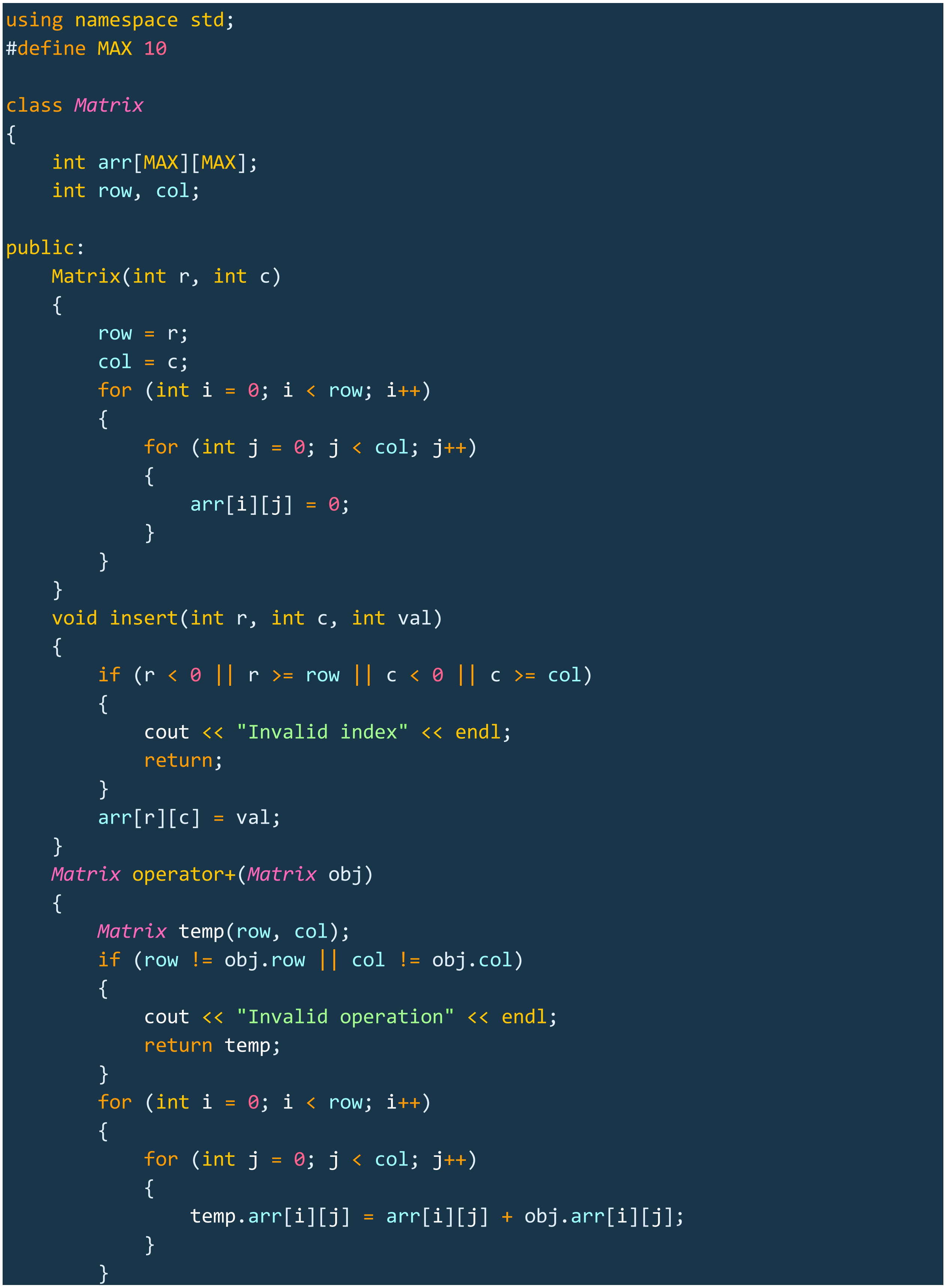


```
 ; if ($?) { g++ ass11.cpp -o ass11 } ; if ($?) { .\ass11 }
Enter first vector:
Enter x-coordinate: 5
Enter y-coordinate: 6
Enter second vector:
Enter x-coordinate: 2
Enter y-coordinate: 8
Result of scalar multiplication (2 * v1): (10, 12)
coordinates of the v1: (5, 6)
coordinates of the 4v2: (2, 8)
PS D:\my progams\normal\oops Assignment>
```

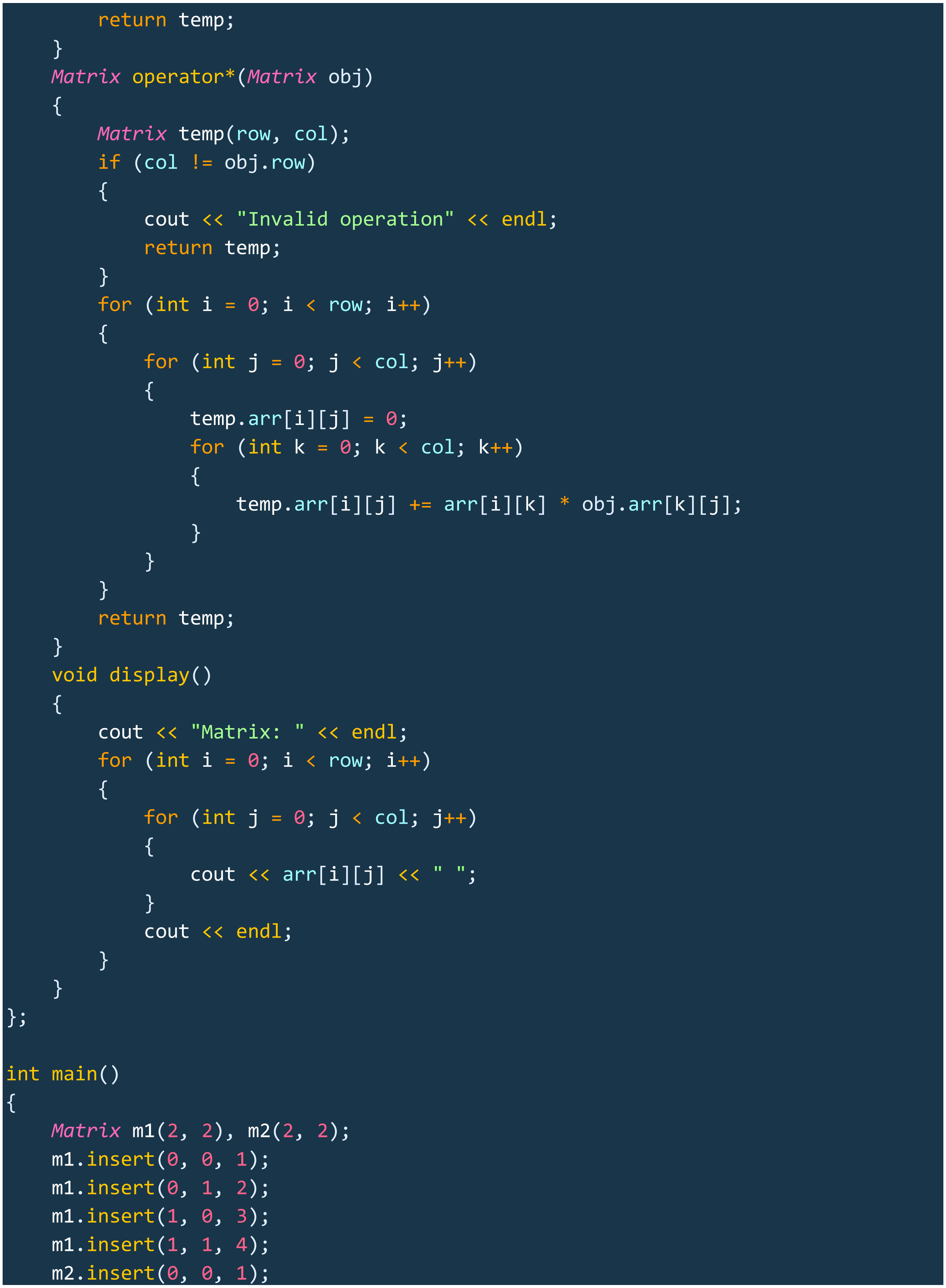
//assignment 12



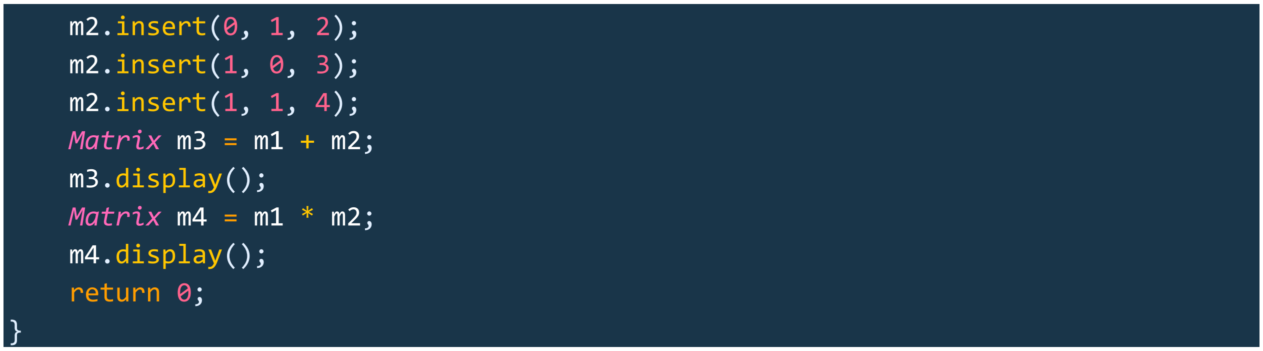
```
#include <iostream>
```



```
using namespace std;
#define MAX 10

class Matrix
{
    int arr[MAX][MAX];
    int row, col;

public:
    Matrix(int r, int c)
    {
        row = r;
        col = c;
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                arr[i][j] = 0;
            }
        }
    }
    void insert(int r, int c, int val)
    {
        if (r < 0 || r >= row || c < 0 || c >= col)
        {
            cout << "Invalid index" << endl;
            return;
        }
        arr[r][c] = val;
    }
    Matrix operator+(Matrix obj)
    {
        Matrix temp(row, col);
        if (row != obj.row || col != obj.col)
        {
            cout << "Invalid operation" << endl;
            return temp;
        }
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                temp.arr[i][j] = arr[i][j] + obj.arr[i][j];
            }
        }
```

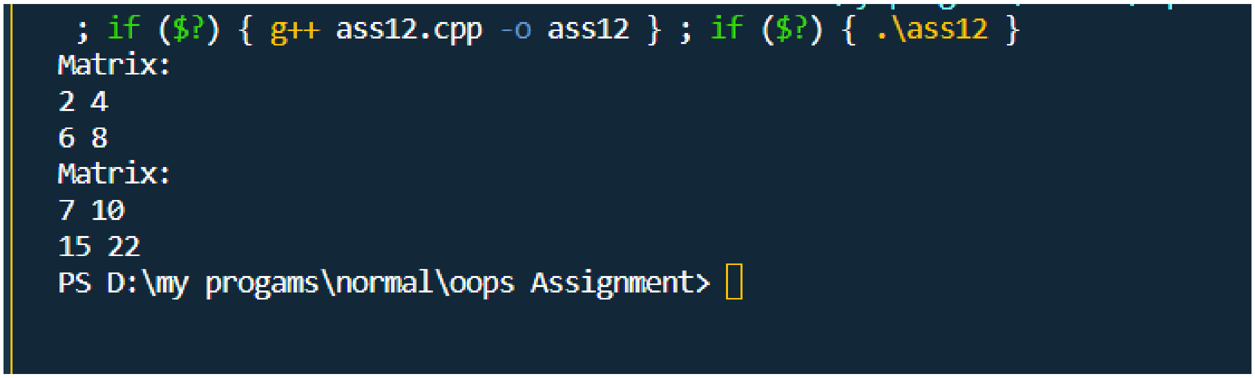


```
        return temp;
    }
    Matrix operator*(Matrix obj)
    {
        Matrix temp(row, col);
        if (col != obj.row)
        {
            cout << "Invalid operation" << endl;
            return temp;
        }
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                temp.arr[i][j] = 0;
                for (int k = 0; k < col; k++)
                {
                    temp.arr[i][j] += arr[i][k] * obj.arr[k][j];
                }
            }
        }
        return temp;
    }
    void display()
    {
        cout << "Matrix: " << endl;
        for (int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
                cout << arr[i][j] << " ";
            }
            cout << endl;
        }
    }
};

int main()
{
    Matrix m1(2, 2), m2(2, 2);
    m1.insert(0, 0, 1);
    m1.insert(0, 1, 2);
    m1.insert(1, 0, 3);
    m1.insert(1, 1, 4);
    m2.insert(0, 0, 1);
```



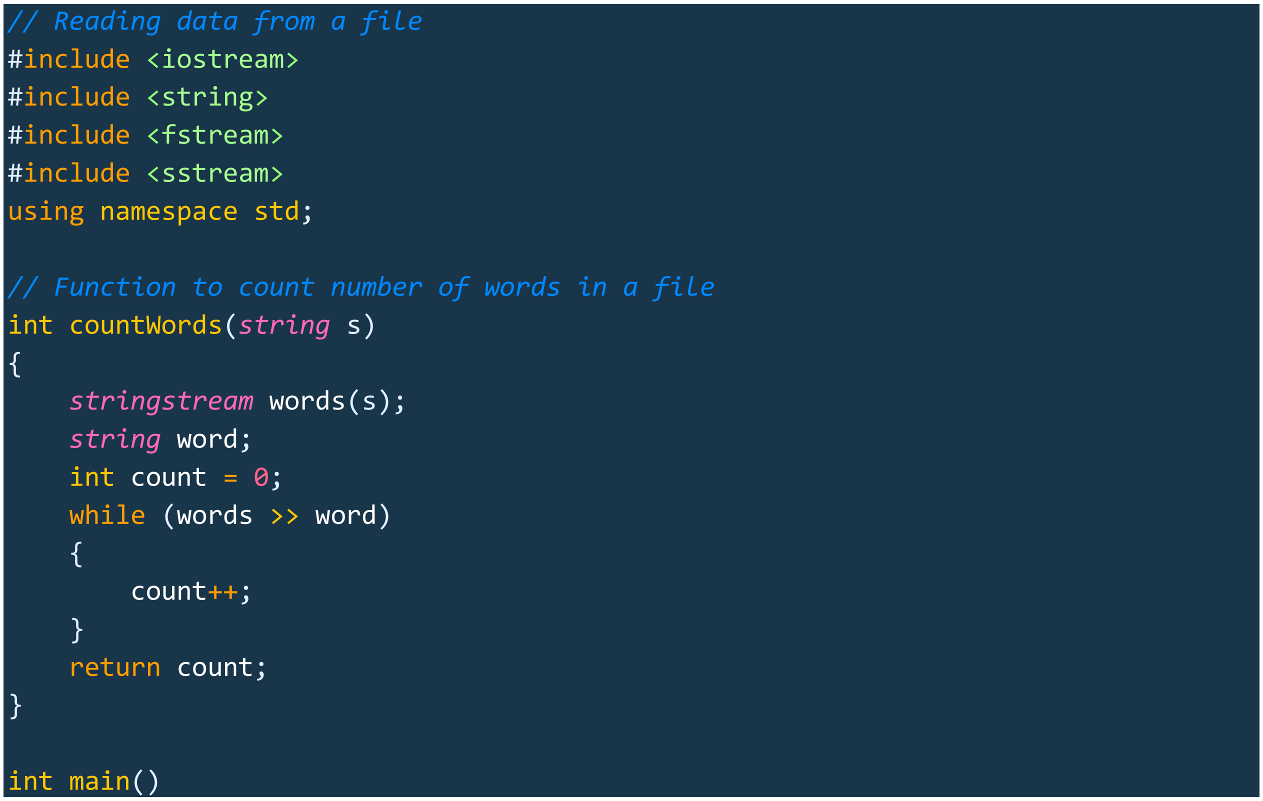
```
    m2.insert(0, 1, 2);
    m2.insert(1, 0, 3);
    m2.insert(1, 1, 4);
    Matrix m3 = m1 + m2;
    m3.display();
    Matrix m4 = m1 * m2;
    m4.display();
    return 0;
}
```

Output->



```
 ; if ($?) { g++ ass12.cpp -o ass12 } ; if ($?) { .\ass12 }
Matrix:
2 4
6 8
Matrix:
7 10
15 22
PS D:\my progams\normal\oops Assignment> 
```

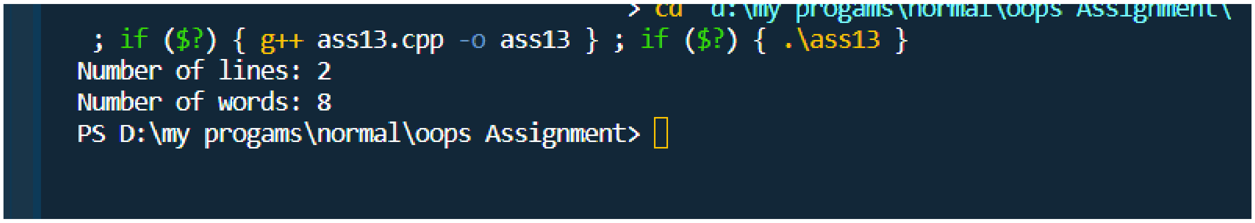
//assignment 13



```
// Reading data from a file
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
using namespace std;

// Function to count number of words in a file
int countWords(string s)
{
    stringstream words(s);
    string word;
    int count = 0;
    while (words >> word)
    {
        count++;
    }
    return count;
}

int main()
```

```
{
    ifstream infile("test.txt");
    string line;
    int lineCount = 0;
    int wordCount = 0;

    // Count number of lines and words
    while (getline(infile, line))
    {
        lineCount++;
        wordCount += countWords(line);
    }

    cout << "Number of lines: " << lineCount << endl
         << "Number of words: " << wordCount << endl;
    return 0;
}
```

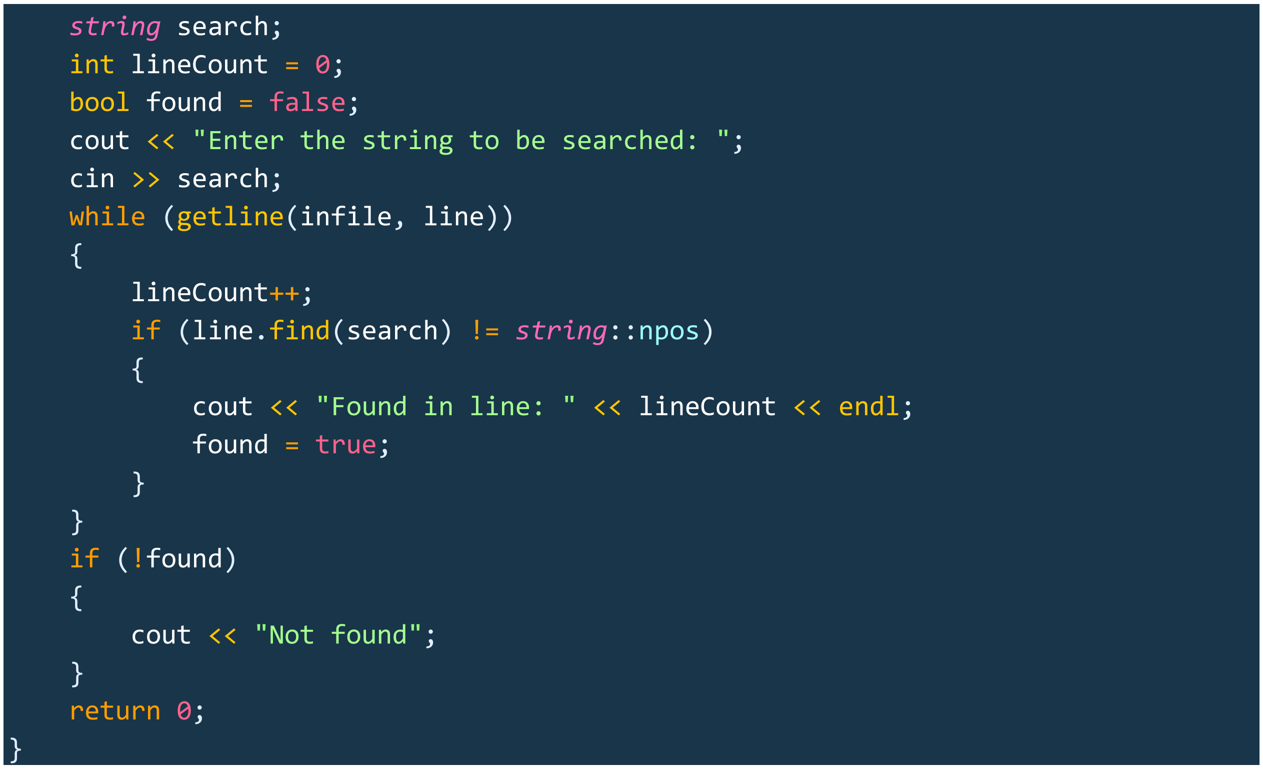
Output->



```
                                        > cd  d:\my progams\normal\oops Assignment\
 ; if ($?) { g++ ass13.cpp -o ass13 } ; if ($?) { .\ass13 }
Number of lines: 2
Number of words: 8
PS D:\my progams\normal\oops Assignment> 
```

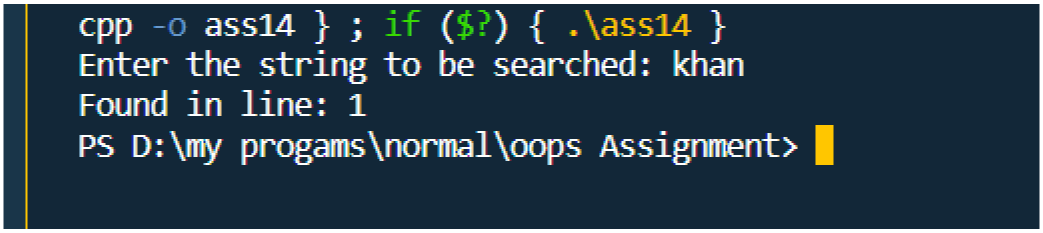
//assignment 14

```
// Finding specific string in a file
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

int main()
{
    ifstream infile("test.txt");
    if (!infile)
    {
        cout << "File not found";
        return 0;
    }
    string line;
```



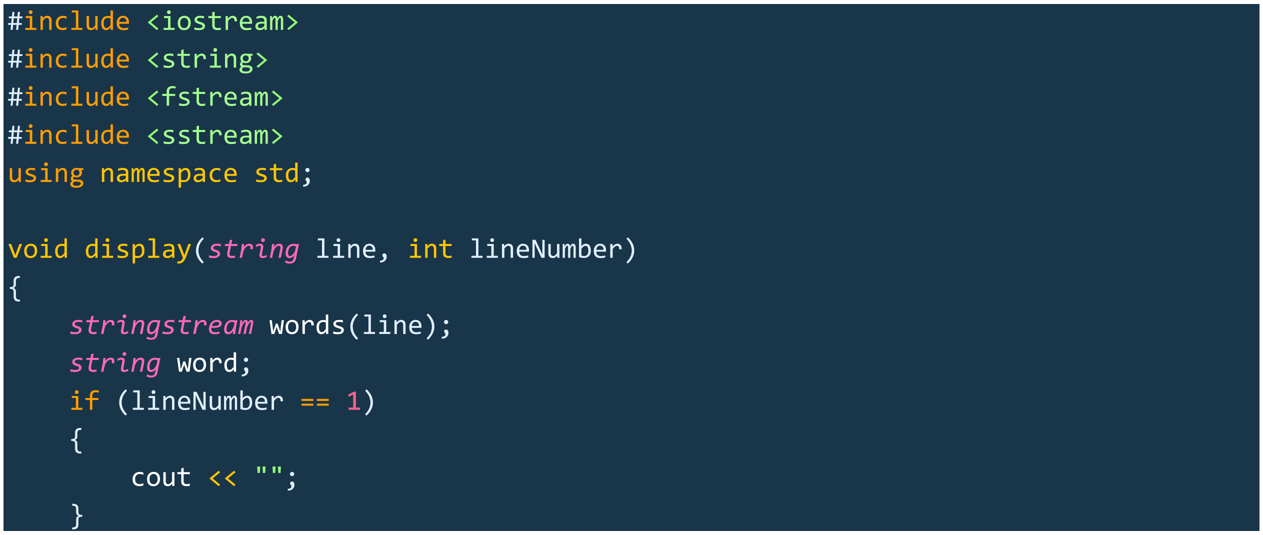
```
    string search;
    int lineCount = 0;
    bool found = false;
    cout << "Enter the string to be searched: ";
    cin >> search;
    while (getline(infile, line))
    {
        lineCount++;
        if (line.find(search) != string::npos)
        {
            cout << "Found in line: " << lineCount << endl;
            found = true;
        }
    }
    if (!found)
    {
        cout << "Not found";
    }
    return 0;
}
```

Output ->

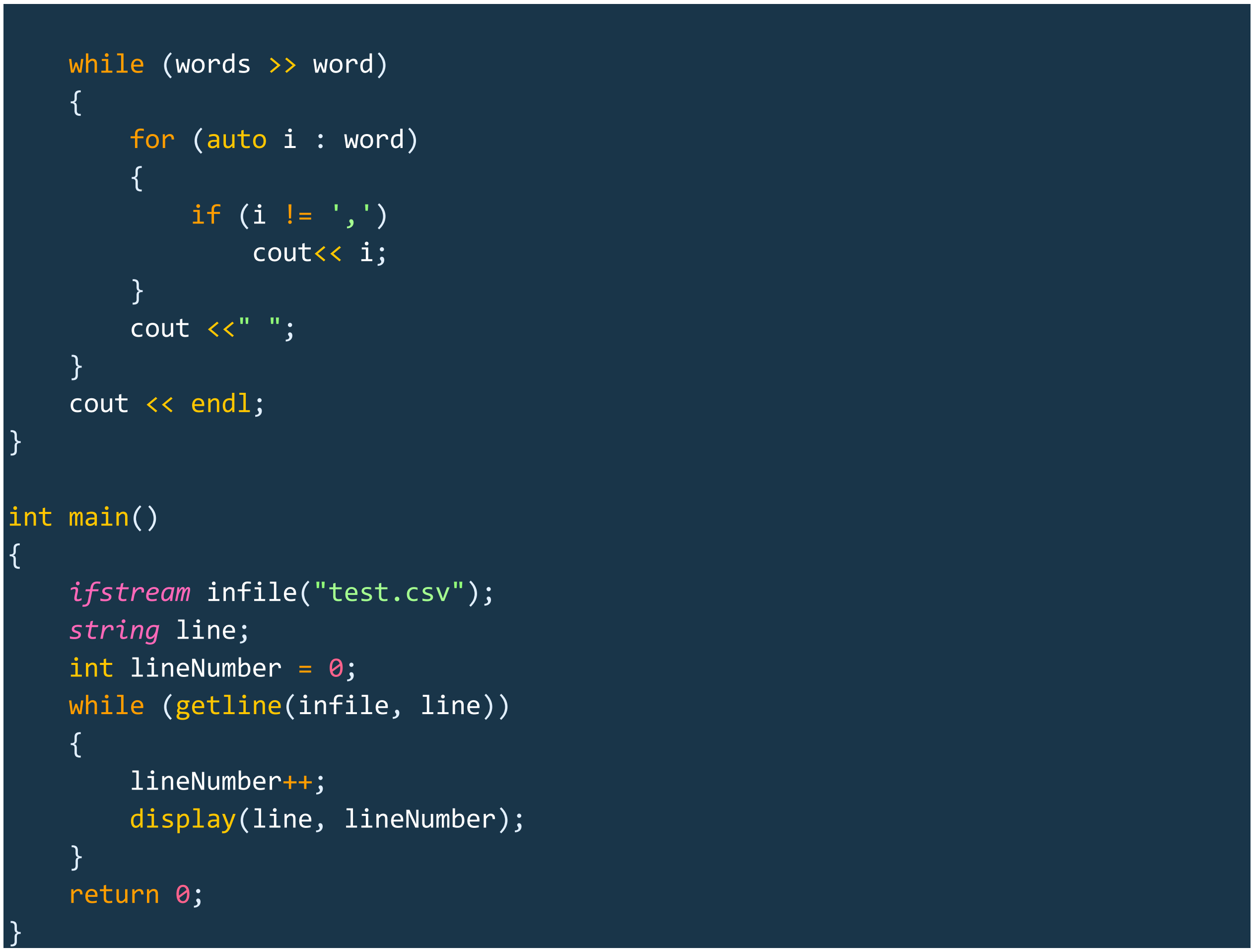


```
cpp -o ass14 } ; if ($?) { .\ass14 }
Enter the string to be searched: khan
Found in line: 1
PS D:\my progams\normal\oops Assignment>
```

//assignment 15

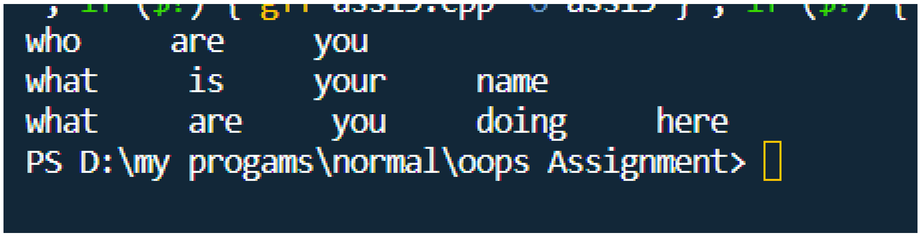


```
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
using namespace std;

void display(string line, int lineNumber)
{
    stringstream words(line);
    string word;
    if (lineNumber == 1)
    {
        cout << "";
    }
```



```
    while (words >> word)
    {
        for (auto i : word)
        {
            if (i != ',')
                cout<< i;
        }
        cout <<" ";
    }
    cout << endl;
}

int main()
{
    ifstream infile("test.csv");
    string line;
    int lineNumber = 0;
    while (getline(infile, line))
    {
        lineNumber++;
        display(line, lineNumber);
    }
    return 0;
}
```

Output ->



```
 ; if ($?) { g++ ass15.cpp -o ass15 } ; if ($?) {
who     are     you
what     is     your     name
what     are     you     doing     here
PS D:\my progams\normal\oops Assignment> 
```