LAB 4
STANDARD TEMPLATE LIBRARY

Data Structures

AGENDA

- Introducing STL
- Vectors
- Vector Example
- Task1: Vector Course Application.
- Task 2: Stack Application

STL (STANDARD TEMPLATE LIBRARY)

a powerful set of C++ template classes that implement many popular and commonly used algorithms and data structures like

- vectors
- Lists
- Map
- Queues
- stacks.

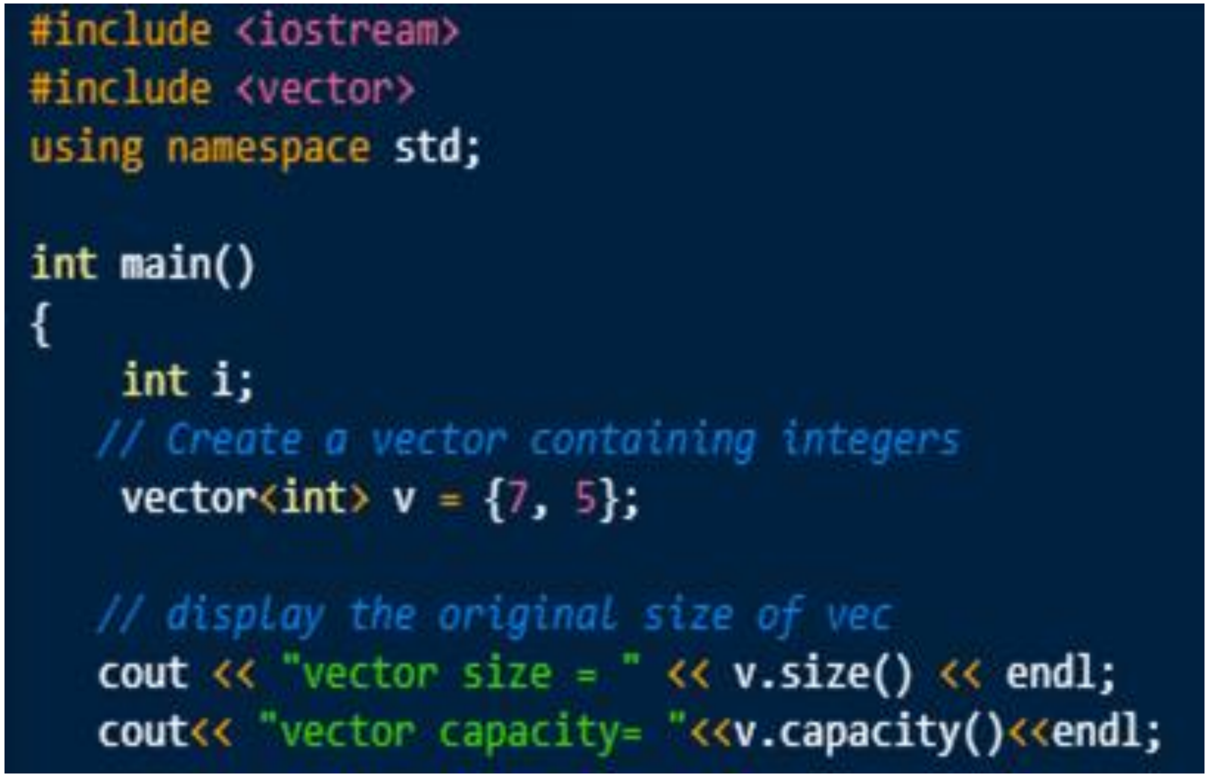
VECTOR

- Its is the Array list Data Structure in STL
- Vector is a template-based container that behaves just like a Dynamic Array.
- It can expand its memory at run time

VECTOR FUNCTIONS AND OPERATORS

size	Return size (public member function)
capacity	Return size of allocated storage capacity(public member function
empty	Test whether vector is empty (public member function)
operator[]	Access element (public member function)
at	Access element (public member function)
front	Access first element (public member function)
back	Access last element (public member function)
push_back	Add element at the end (public member function)
pop_back	Delete last element (public member function)
insert	Insert elements (public member function)

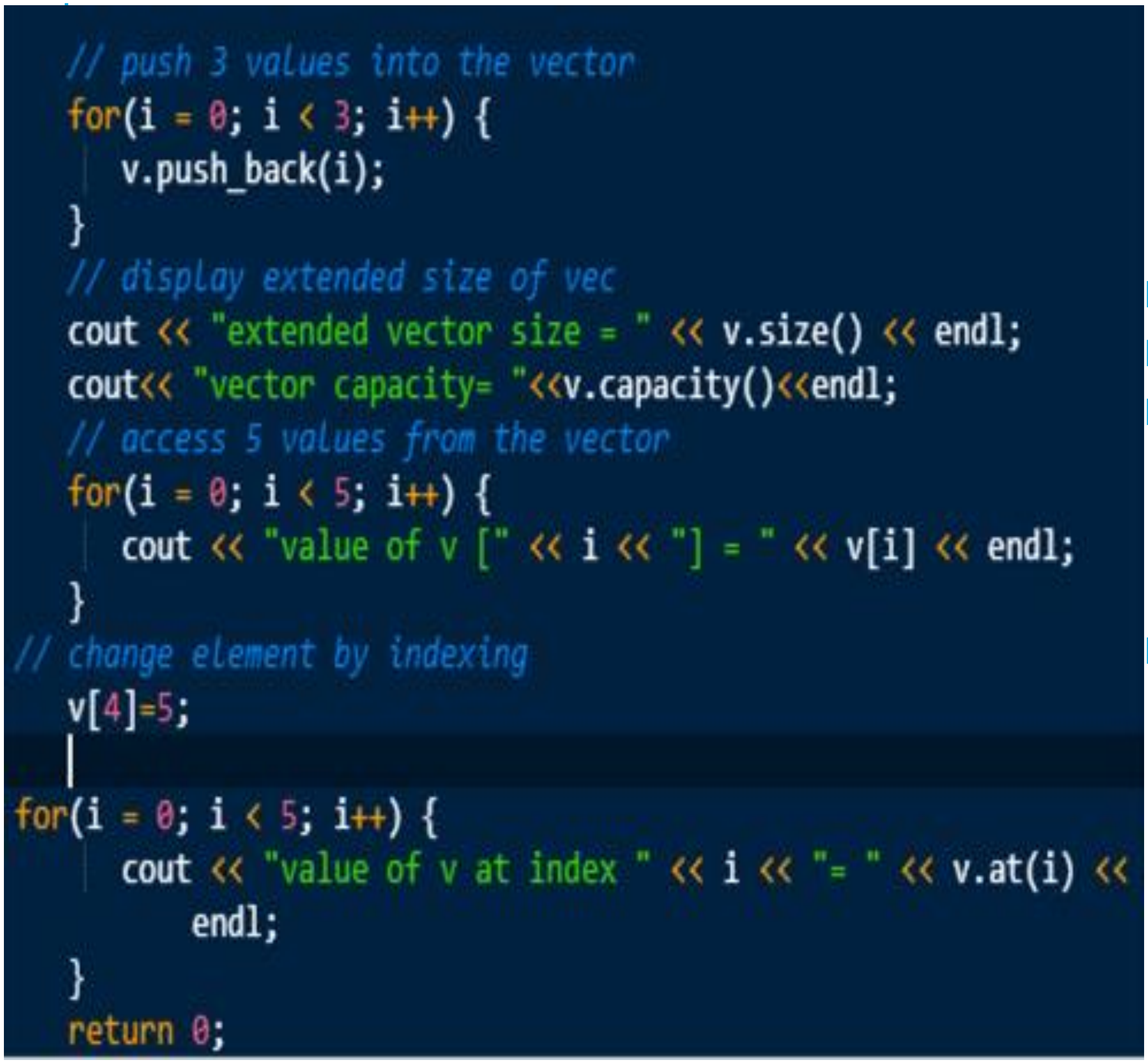
VECTOR EXAMPLE



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

int main()
{
    int i;
   // Create a vector containing integers
    vector<int> v = {7, 5};

   // display the original size of vec
   cout << "vector size = " << v.size() << endl;
   cout<< "vector capacity= "<<v.capacity()<<endl;
```

Output:

- vector size = 2
- vector capacity= 2



```
    // push 3 values into the vector
    for(i = 0; i < 3; i++) {
        v.push_back(i);
    }
    // display extended size of vec
    cout << "extended vector size = " << v.size() << endl;
    cout<< "vector capacity= "<<v.capacity()<<endl;
    // access 5 values from the vector
    for(i = 0; i < 5; i++) {
        cout << "value of v [" << i << "] = " << v[i] << endl;
    }
// change element by indexing
    v[4]=5;
    
for(i = 0; i < 5; i++) {
        cout << "value of v at index " << i << "= " << v.at(i) <<
            endl;
    }
    return 0;
```

Output:

- extended vector size = 5
- vector capacity= 8
- value of v [0] = 7
  value of v [1] = 5
  value of v [2] = 0
  value of v [3] = 1
  value of v [4] = 2
- value of v at index 0= 7
  value of v at index 1= 5
  value of v at index 2= 0
  value of v at index 3= 1
  value of v at index 4= 5

Vector: Course Application

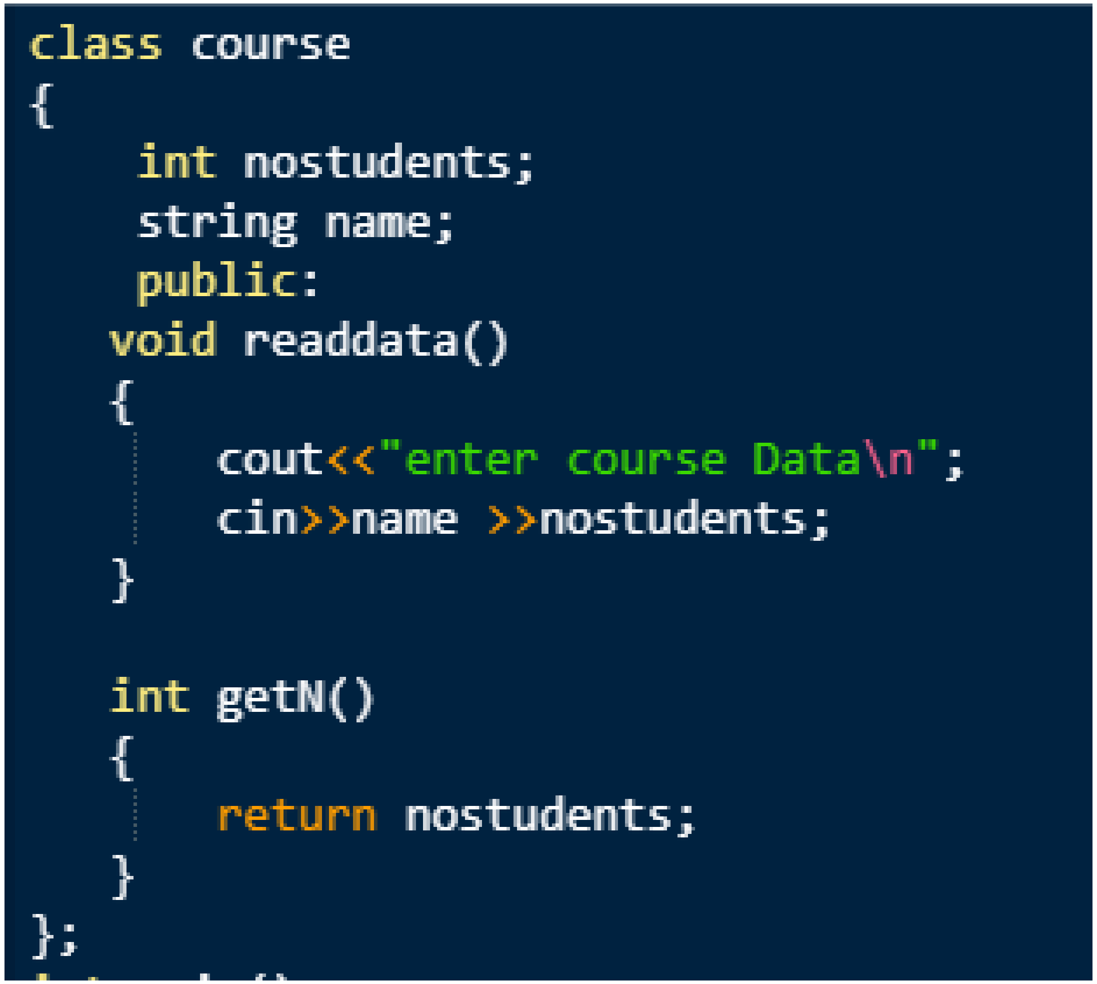
Implement a program using Vectors to take form user courses till he stops then display the total number of students enrolled in courses.
Where each course has Name and number of students.

15 minutes

Sample Run

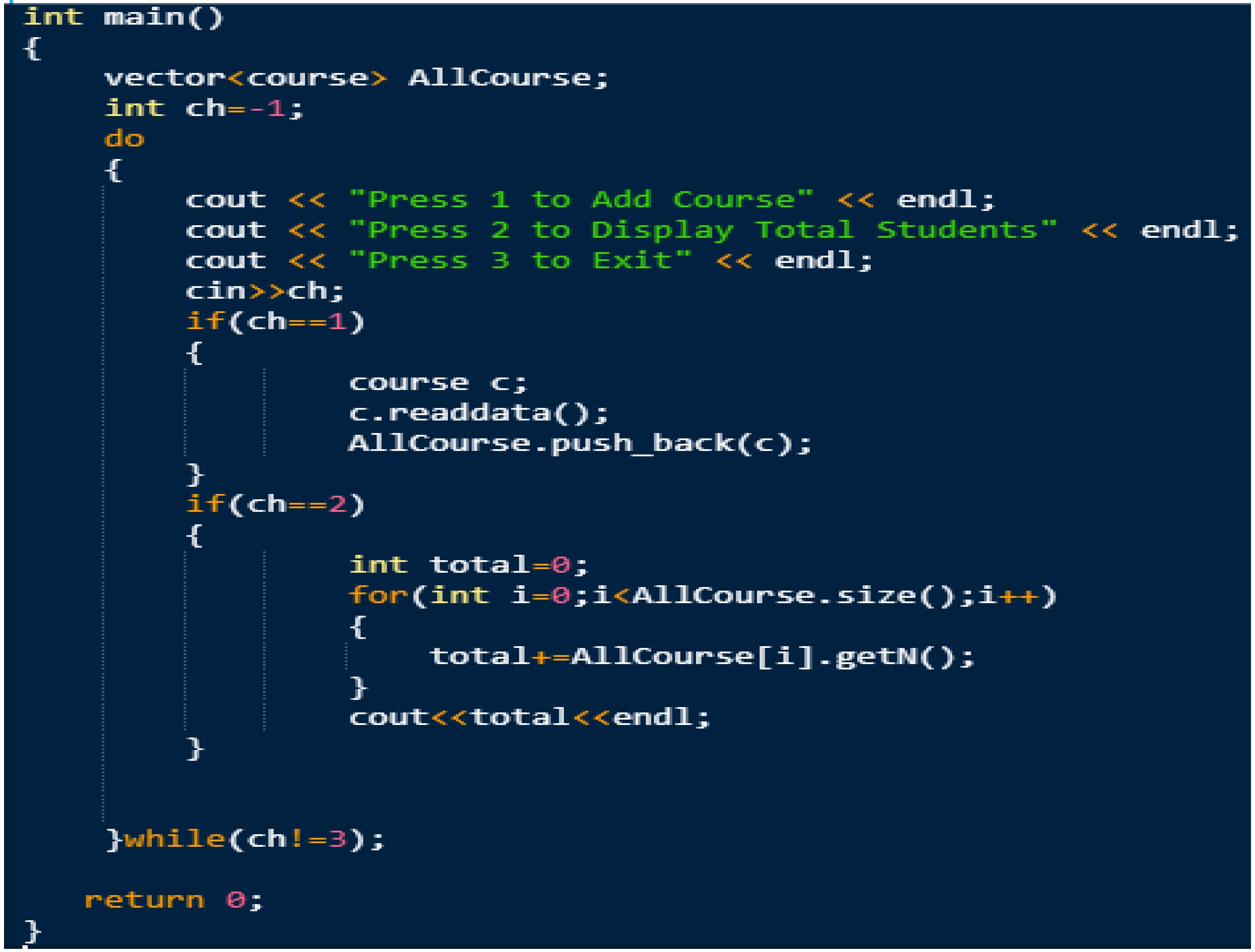
```
Press 1 to Add Course  Press 2 to Display Total Students  Press 3 to Exit
1
enter course Data

DS 30

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit

1
enter course Data

SP 40

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit

2

Total =70

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit

3
```

Vector: Course Application *“SOLUTION”*



```
class course
{
     int nostudents;
     string name;
     public:
    void readdata()
    {
        cout<<"enter course Data\n";
        cin>>name >>nostudents;
    }

    int getN()
    {
        return nostudents;
    }
};
```

Vector: Course Application "SOLUTION"



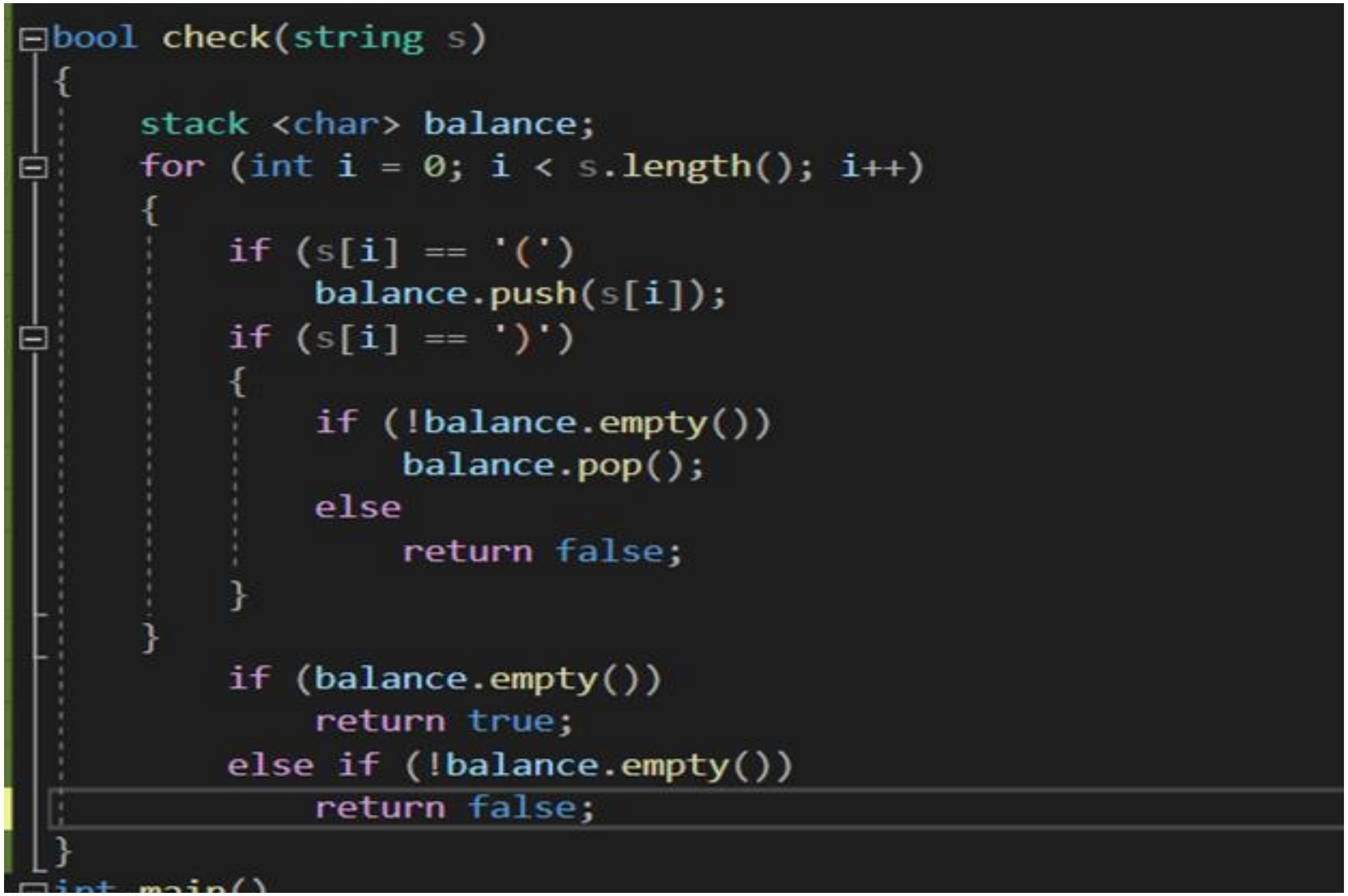
```
int main()
{
    vector<course> AllCourse;
    int ch=-1;
    do
    {
        cout << "Press 1 to Add Course" << endl;
        cout << "Press 2 to Display Total Students" << endl;
        cout << "Press 3 to Exit" << endl;
        cin>>ch;
        if(ch==1)
        {
                course c;
                c.readdata();
                AllCourse.push_back(c);
        }
        if(ch==2)
        {
                int total=0;
                for(int i=0;i<AllCourse.size();i++)
                {
                    total+=AllCourse[i].getN();
                }
                cout<<total<<endl;
        }

    }while(ch!=3);

   return 0;
}
```

Stack: Application

Implement a program that reads in a sequence of characters, and determines whether its parentheses are "balanced." using function

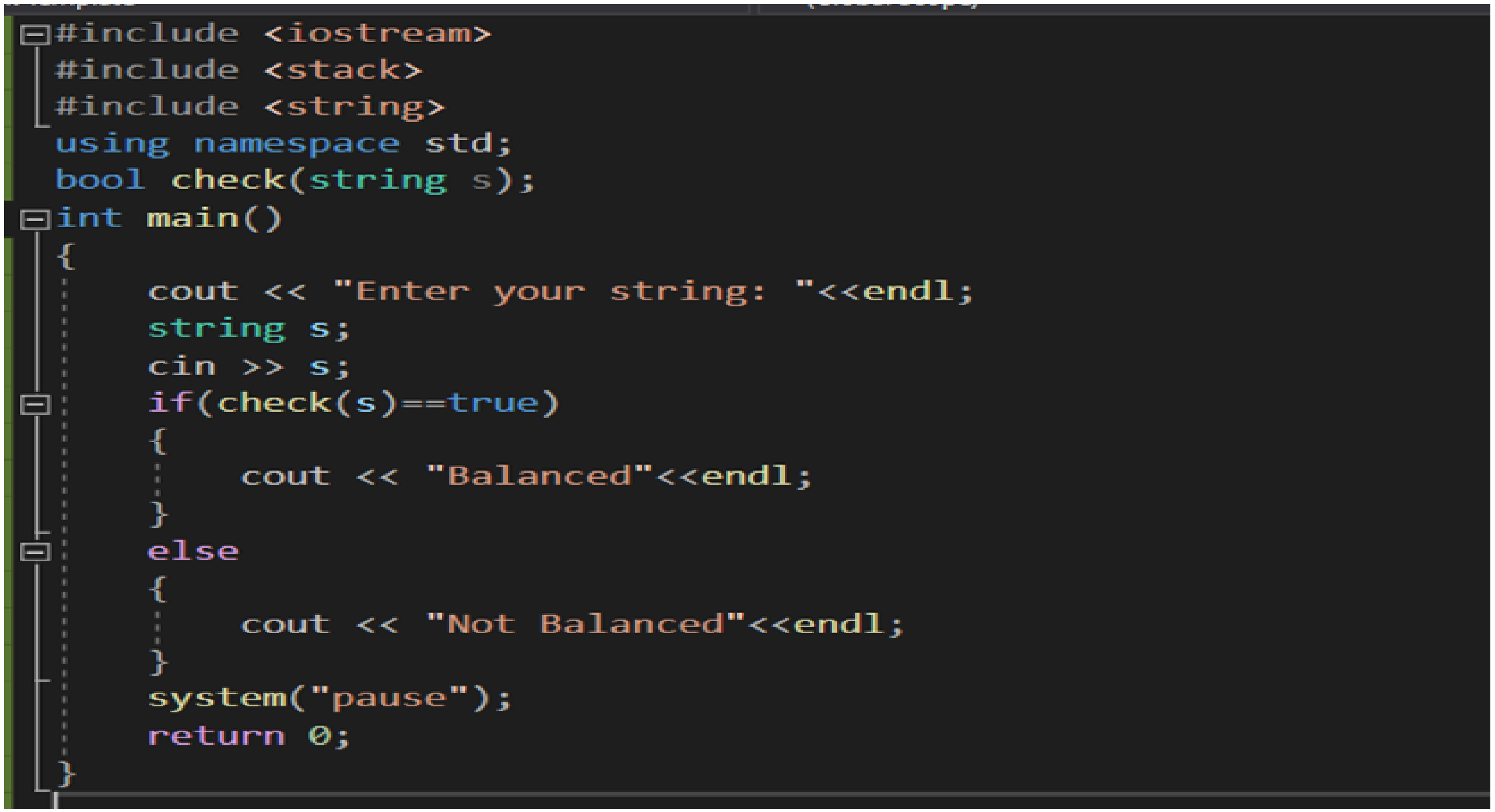
15 minutes

Stack:Application *"SOLUTION"*



```
bool check(string s)
{
    stack <char> balance;
    for (int i = 0; i < s.length(); i++)
    {
        if (s[i] == '(')
            balance.push(s[i]);
        if (s[i] == ')')
        {
            if (!balance.empty())
                balance.pop();
            else
                return false;
        }
    }
        if (balance.empty())
            return true;
        else if (!balance.empty())
            return false;
}
int main()
```

Stack:Application *"SOLUTION"*



```cpp
#include <iostream>
#include <stack>
#include <string>
using namespace std;
bool check(string s);
int main()
{
    cout << "Enter your string: "<<endl;
    string s;
    cin >> s;
    if(check(s)==true)
    {
        cout << "Balanced"<<endl;
    }
    else
    {
        cout << "Not Balanced"<<endl;
    }
    system("pause");
    return 0;
}
```

thank you