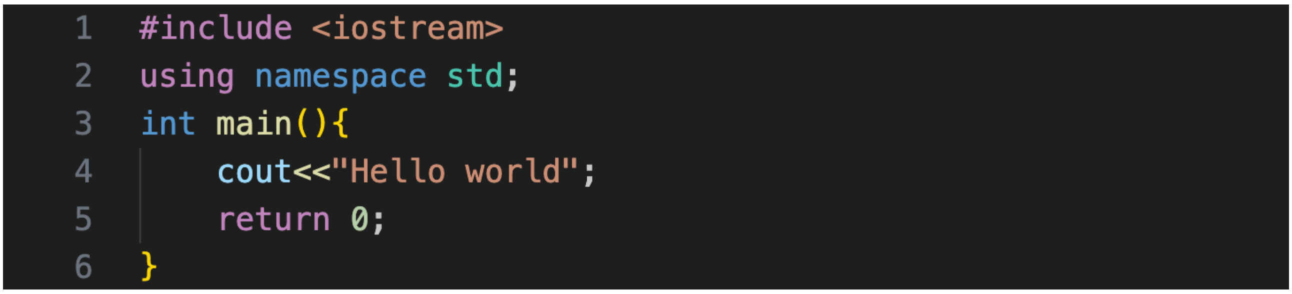
OBJECT ORIENTED PROGRAMMING
LAB ASSIGNMENT -1

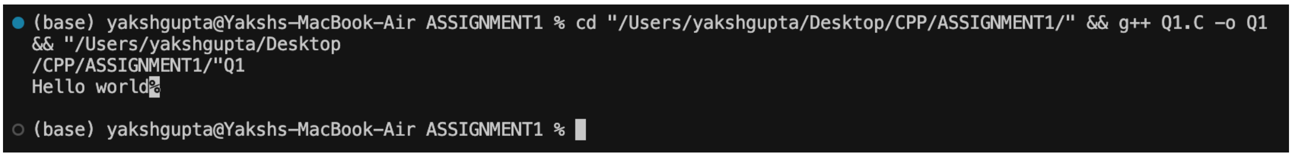
1. Write a program (WAP) to display "Hello World" on console display.

CODE:



```cpp
#include <iostream>
using namespace std;
int main(){
    cout<<"Hello world";
    return 0;
}
```

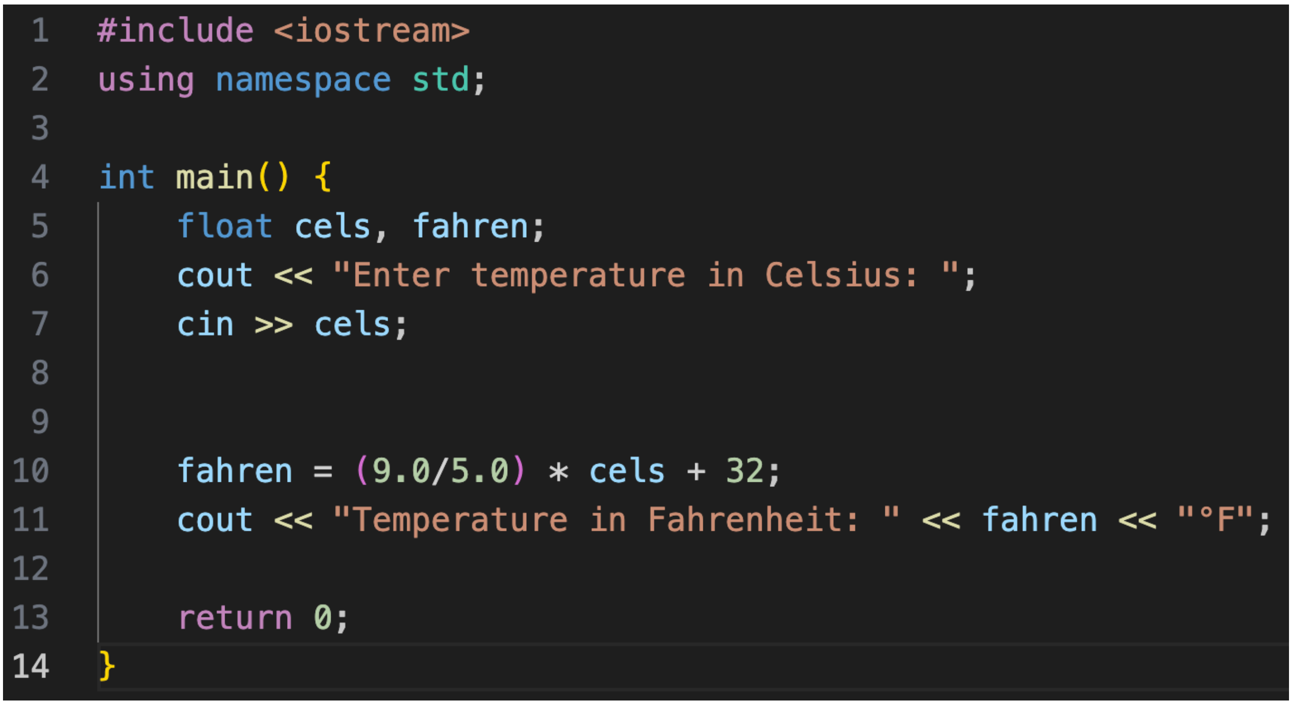
OUTPUT:



```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/" && g++ Q1.C -o Q1
  && "/Users/yakshgupta/Desktop
  /CPP/ASSIGNMENT1/"Q1
  Hello world%

○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % 
```

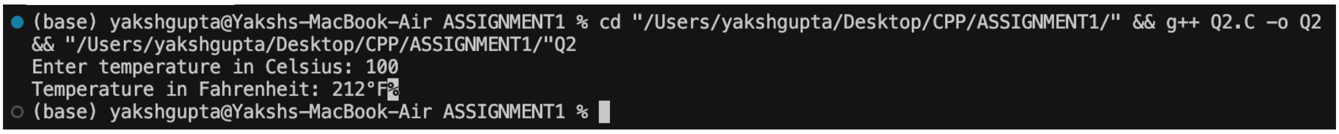
2. Write a C++ program that will ask for a temperature in Celsius and display it in degree Fahrenheit.[F=9C/5+32]

CODE:



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

int main() {
    float cels, fahren;
    cout << "Enter temperature in Celsius: ";
    cin >> cels;


    fahren = (9.0/5.0) * cels + 32;
    cout << "Temperature in Fahrenheit: " << fahren << "°F";

    return 0;
}
```

OUTPUT:



```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/" && g++ Q2.C -o Q2
  && "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/"Q2
  Enter temperature in Celsius: 100
  Temperature in Fahrenheit: 212°F%
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % 
```

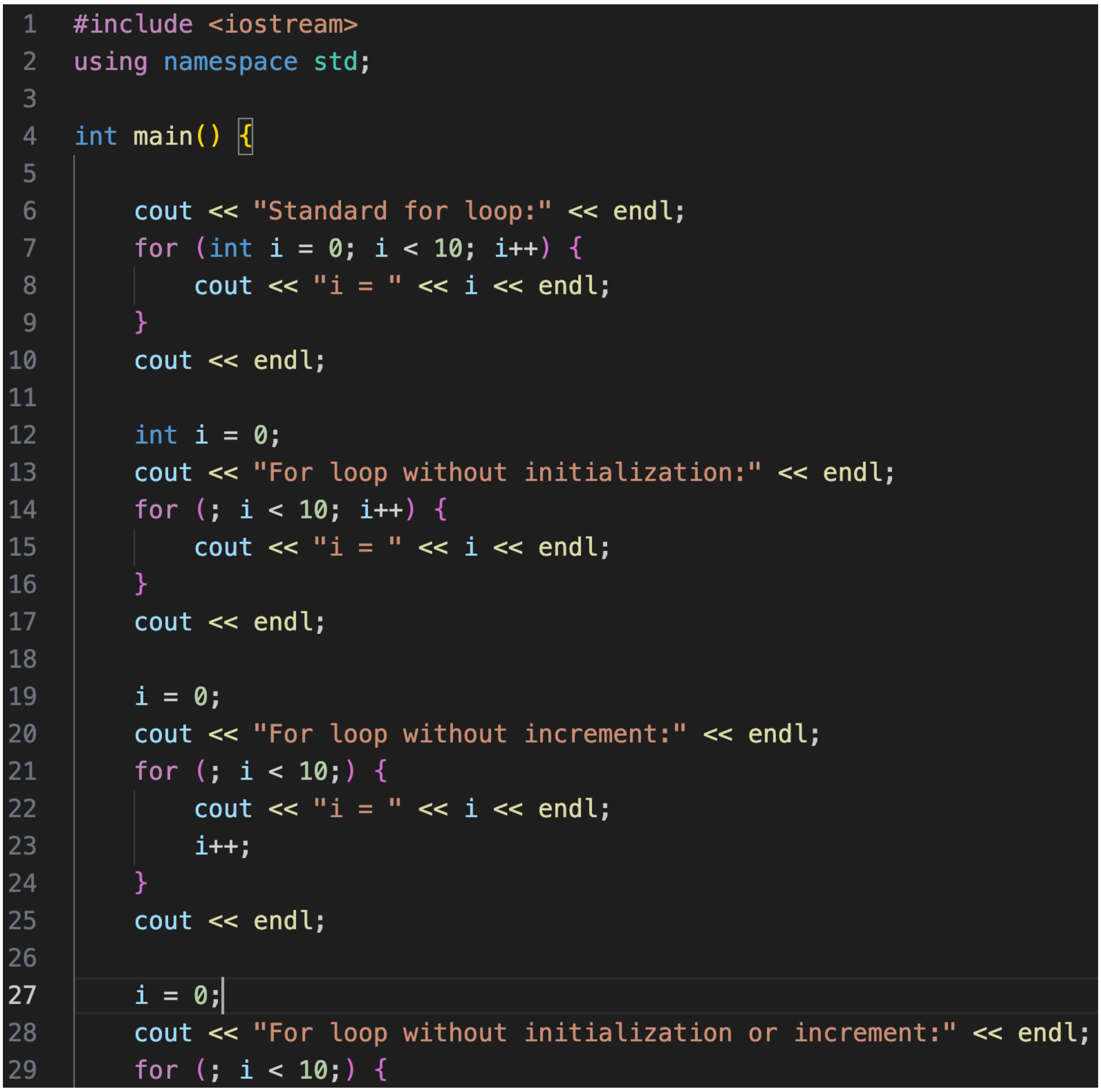
3. WAP to demonstrate for, while, do-while (with all possible variations), like for loop can be demonstrated without giving initialization in for construct or without giving increment in for construct.

   Sample:
   for (**int** i=0; i<10; i++)

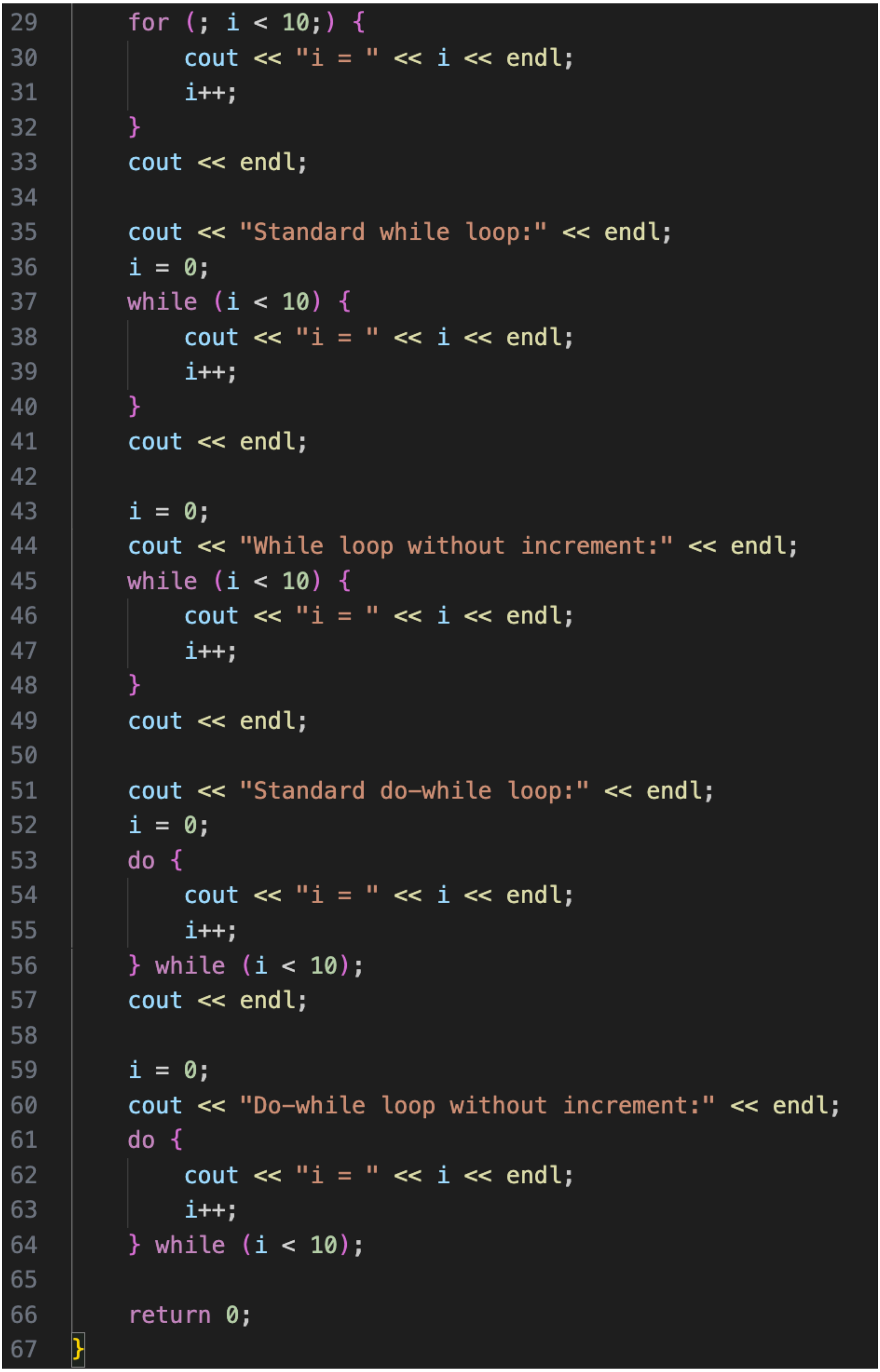
   i=0
   for (; i<10; i++)

   i=0
   for (; i<10;)
   i++

CODE:

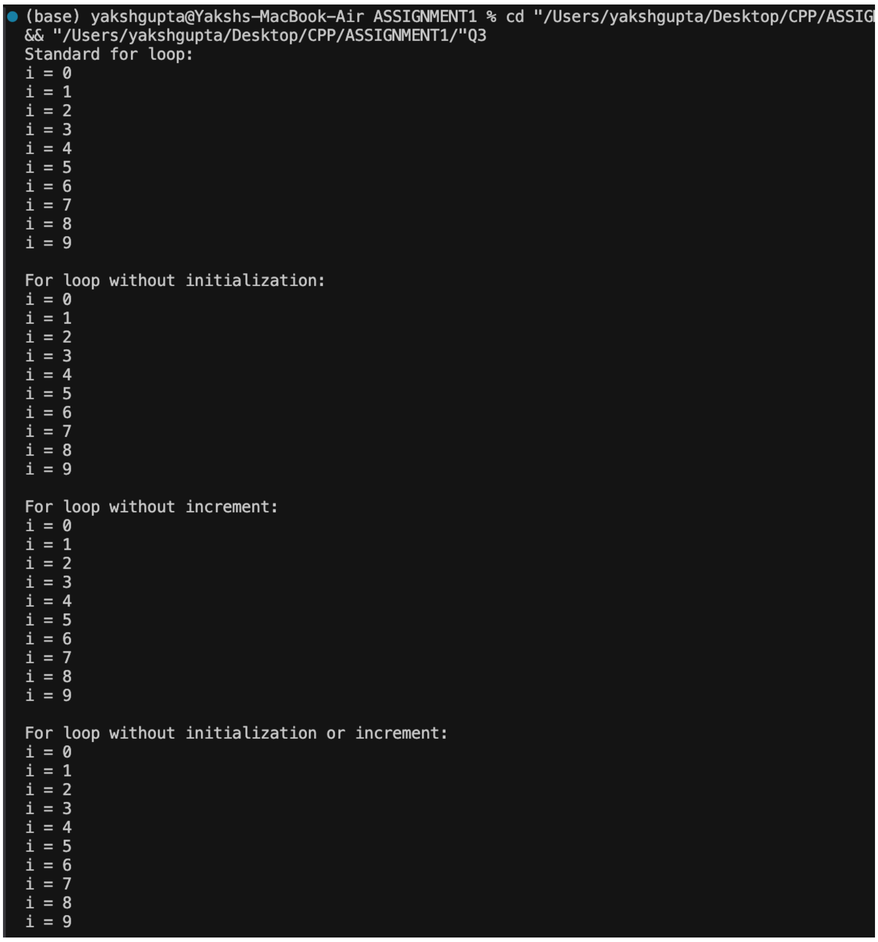


```cpp
#include <iostream>
using namespace std;

int main() {

    cout << "Standard for loop:" << endl;
    for (int i = 0; i < 10; i++) {
        cout << "i = " << i << endl;
    }
    cout << endl;

    int i = 0;
    cout << "For loop without initialization:" << endl;
    for (; i < 10; i++) {
        cout << "i = " << i << endl;
    }
    cout << endl;

    i = 0;
    cout << "For loop without increment:" << endl;
    for (; i < 10;) {
        cout << "i = " << i << endl;
        i++;
    }
    cout << endl;

    i = 0;
    cout << "For loop without initialization or increment:" << endl;
    for (; i < 10;) {
```

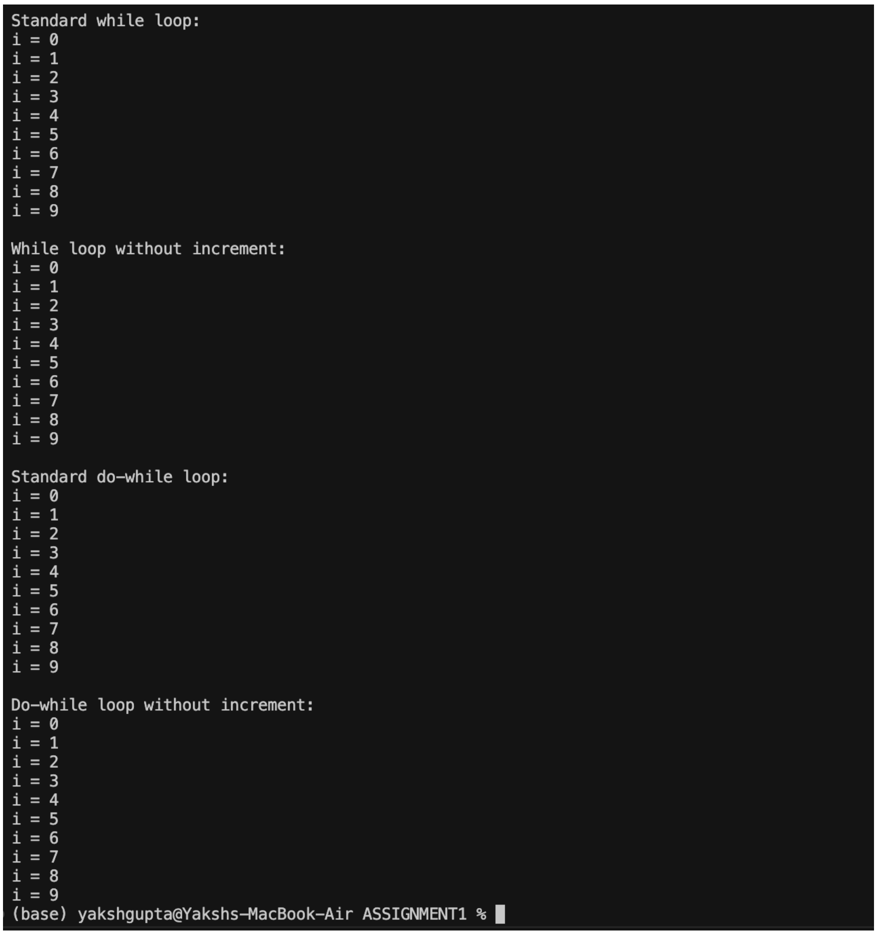


```
    for (; i < 10;) {
        cout << "i = " << i << endl;
        i++;
    }
    cout << endl;

    cout << "Standard while loop:" << endl;
    i = 0;
    while (i < 10) {
        cout << "i = " << i << endl;
        i++;
    }
    cout << endl;

    i = 0;
    cout << "While loop without increment:" << endl;
    while (i < 10) {
        cout << "i = " << i << endl;
        i++;
    }
    cout << endl;

    cout << "Standard do-while loop:" << endl;
    i = 0;
    do {
        cout << "i = " << i << endl;
        i++;
    } while (i < 10);
    cout << endl;

    i = 0;
    cout << "Do-while loop without increment:" << endl;
    do {
        cout << "i = " << i << endl;
        i++;
    } while (i < 10);

    return 0;
}
```

OUTPUT:



```
(base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIG
&& "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/"Q3
Standard for loop:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

For loop without initialization:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

For loop without increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

For loop without initialization or increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
```



```
Standard while loop:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

While loop without increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

Standard do-while loop:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

Do-while loop without increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
(base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```

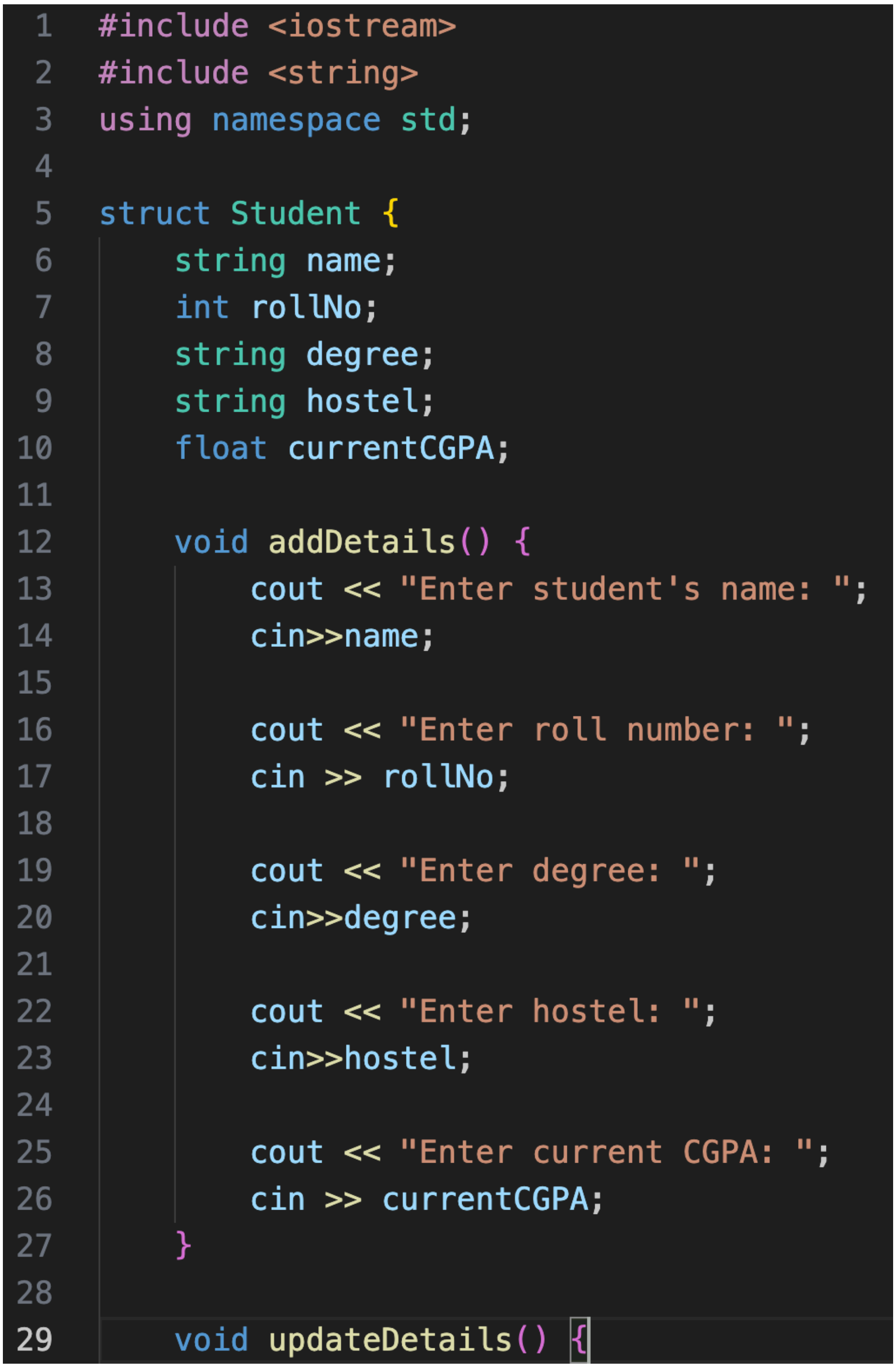
4. Create a structure in C++ containing the details of Students as details below and a main function to execute the structure.

Data Members(properties):

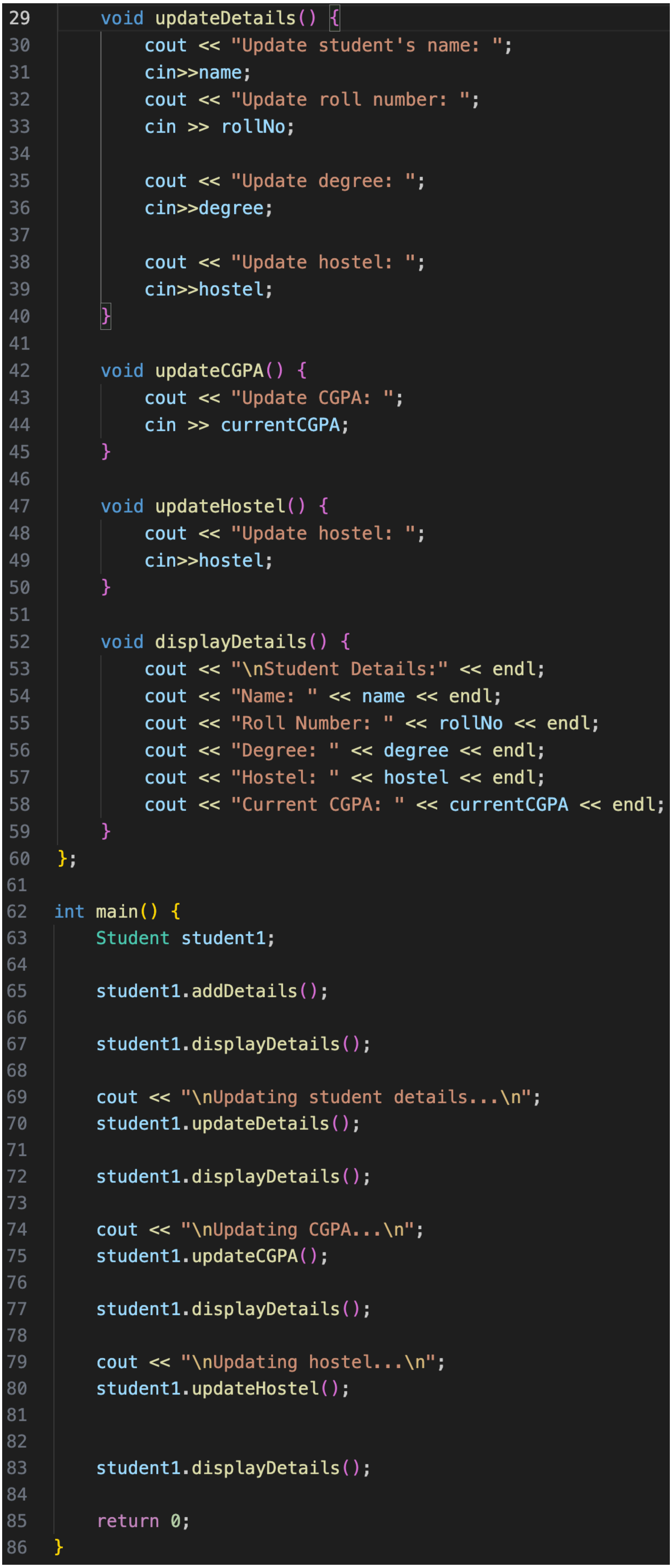
Name
Roll No
Degree
Hostel
CurrentCGPA

Member Function(behavior):

addDetails();
updateDetails();
updateCGPA();
updateHostel();
displaydetails();

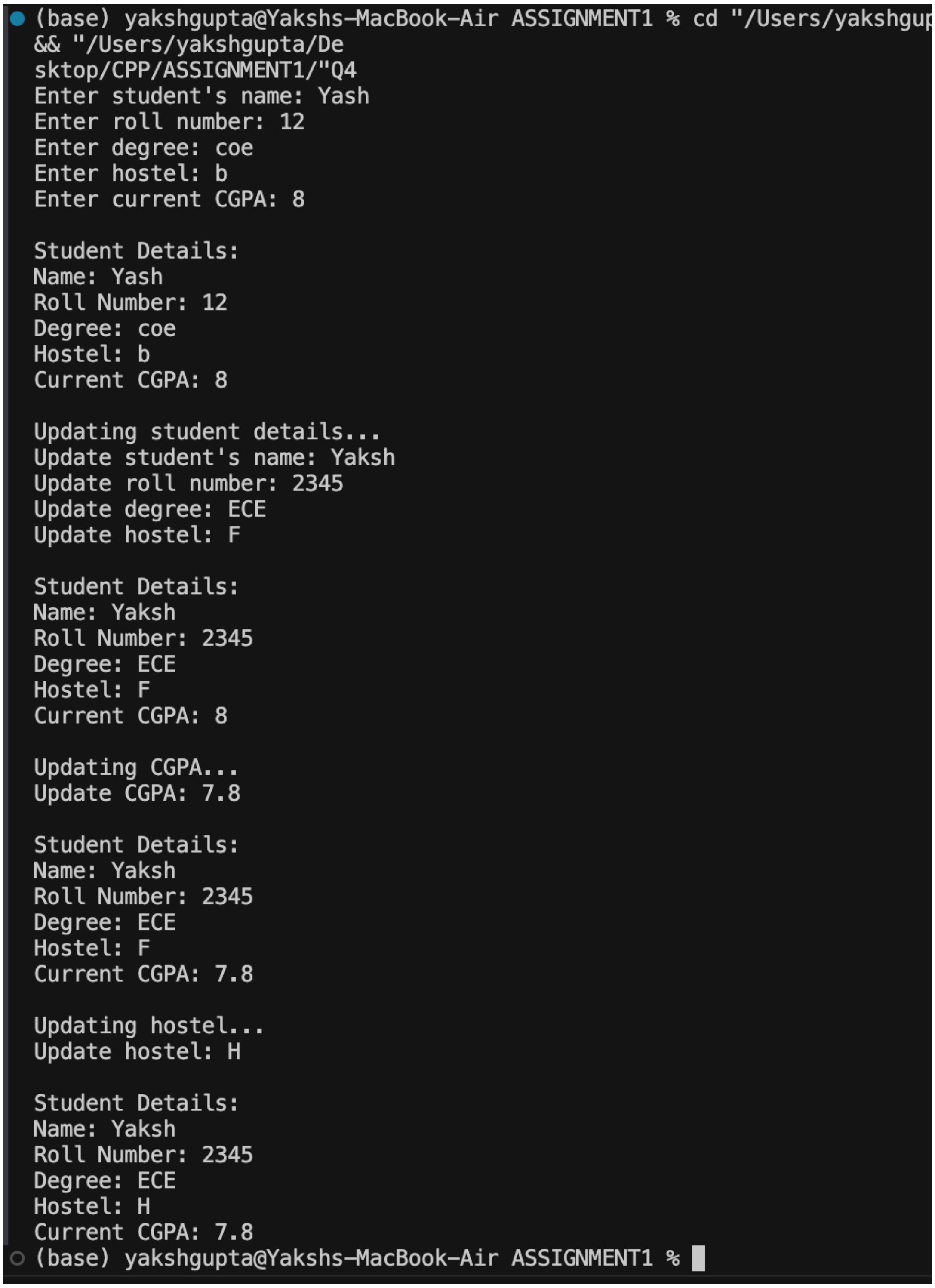


```cpp
#include <iostream>
#include <string>
using namespace std;

struct Student {
    string name;
    int rollNo;
    string degree;
    string hostel;
    float currentCGPA;

    void addDetails() {
        cout << "Enter student's name: ";
        cin>>name;

        cout << "Enter roll number: ";
        cin >> rollNo;

        cout << "Enter degree: ";
        cin>>degree;

        cout << "Enter hostel: ";
        cin>>hostel;

        cout << "Enter current CGPA: ";
        cin >> currentCGPA;
    }

    void updateDetails() {
```



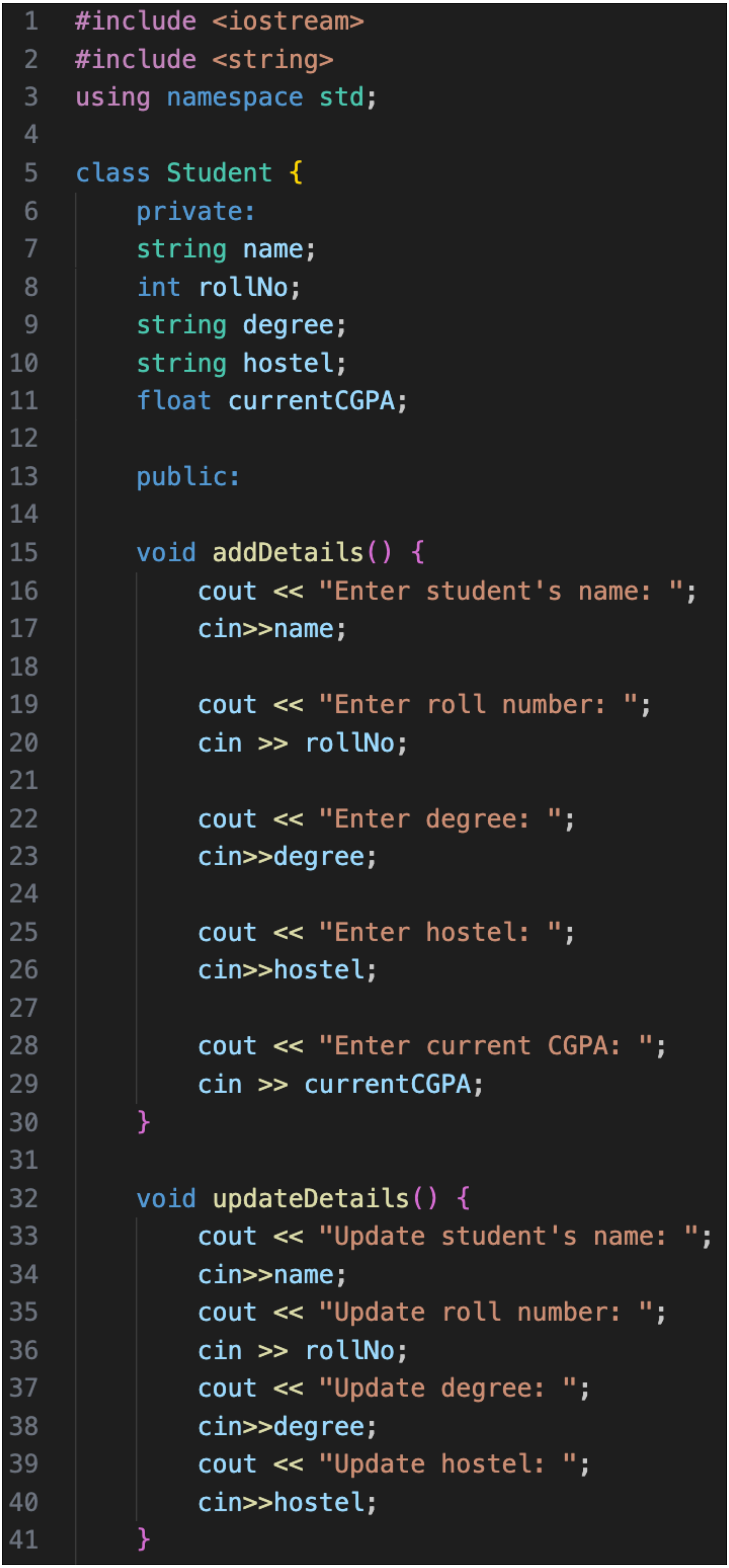
```
    void updateDetails() {
        cout << "Update student's name: ";
        cin>>name;
        cout << "Update roll number: ";
        cin >> rollNo;

        cout << "Update degree: ";
        cin>>degree;

        cout << "Update hostel: ";
        cin>>hostel;
    }

    void updateCGPA() {
        cout << "Update CGPA: ";
        cin >> currentCGPA;
    }

    void updateHostel() {
        cout << "Update hostel: ";
        cin>>hostel;
    }

    void displayDetails() {
        cout << "\nStudent Details:" << endl;
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNo << endl;
        cout << "Degree: " << degree << endl;
        cout << "Hostel: " << hostel << endl;
        cout << "Current CGPA: " << currentCGPA << endl;
    }
};

int main() {
    Student student1;

    student1.addDetails();

    student1.displayDetails();

    cout << "\nUpdating student details...\n";
    student1.updateDetails();

    student1.displayDetails();

    cout << "\nUpdating CGPA...\n";
    student1.updateCGPA();

    student1.displayDetails();

    cout << "\nUpdating hostel...\n";
    student1.updateHostel();


    student1.displayDetails();

    return 0;
}
```

OUTPUT:

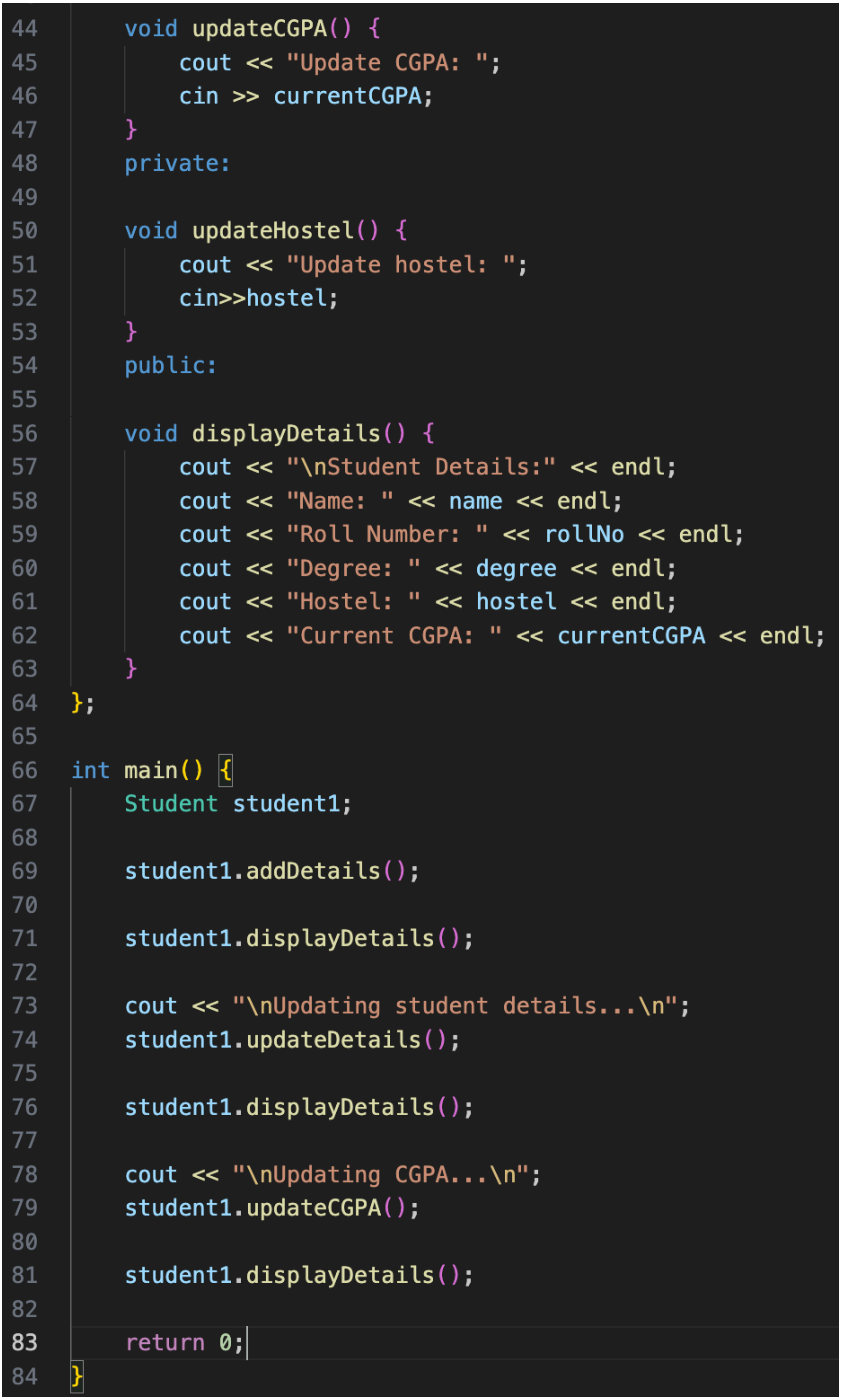


```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgup
  && "/Users/yakshgupta/De
  sktop/CPP/ASSIGNMENT1/"Q4
  Enter student's name: Yash
  Enter roll number: 12
  Enter degree: coe
  Enter hostel: b
  Enter current CGPA: 8

  Student Details:
  Name: Yash
  Roll Number: 12
  Degree: coe
  Hostel: b
  Current CGPA: 8

  Updating student details...
  Update student's name: Yaksh
  Update roll number: 2345
  Update degree: ECE
  Update hostel: F

  Student Details:
  Name: Yaksh
  Roll Number: 2345
  Degree: ECE
  Hostel: F
  Current CGPA: 8

  Updating CGPA...
  Update CGPA: 7.8

  Student Details:
  Name: Yaksh
  Roll Number: 2345
  Degree: ECE
  Hostel: F
  Current CGPA: 7.8

  Updating hostel...
  Update hostel: H

  Student Details:
  Name: Yaksh
  Roll Number: 2345
  Degree: ECE
  Hostel: H
  Current CGPA: 7.8
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % █
```

5. Differentiate between private and public access/scope. Perform the question no. 4 with class instead of structure with having the data members private and some member functions in private scope and some in public scope.

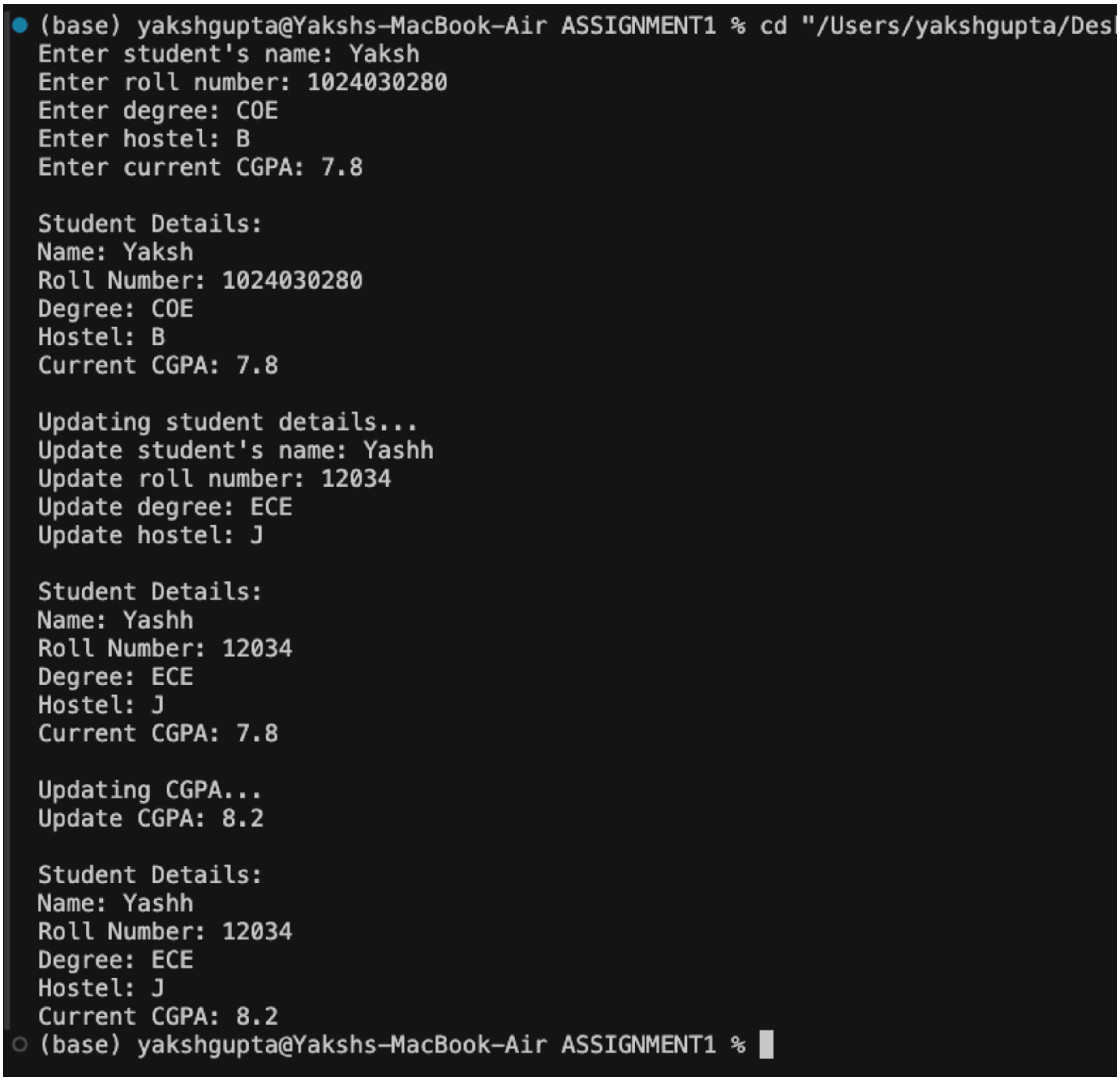


```
#include <iostream>
#include <string>
using namespace std;

class Student {
    private:
    string name;
    int rollNo;
    string degree;
    string hostel;
    float currentCGPA;

    public:

    void addDetails() {
        cout << "Enter student's name: ";
        cin>>name;

        cout << "Enter roll number: ";
        cin >> rollNo;

        cout << "Enter degree: ";
        cin>>degree;

        cout << "Enter hostel: ";
        cin>>hostel;

        cout << "Enter current CGPA: ";
        cin >> currentCGPA;
    }

    void updateDetails() {
        cout << "Update student's name: ";
        cin>>name;
        cout << "Update roll number: ";
        cin >> rollNo;
        cout << "Update degree: ";
        cin>>degree;
        cout << "Update hostel: ";
        cin>>hostel;
    }
```



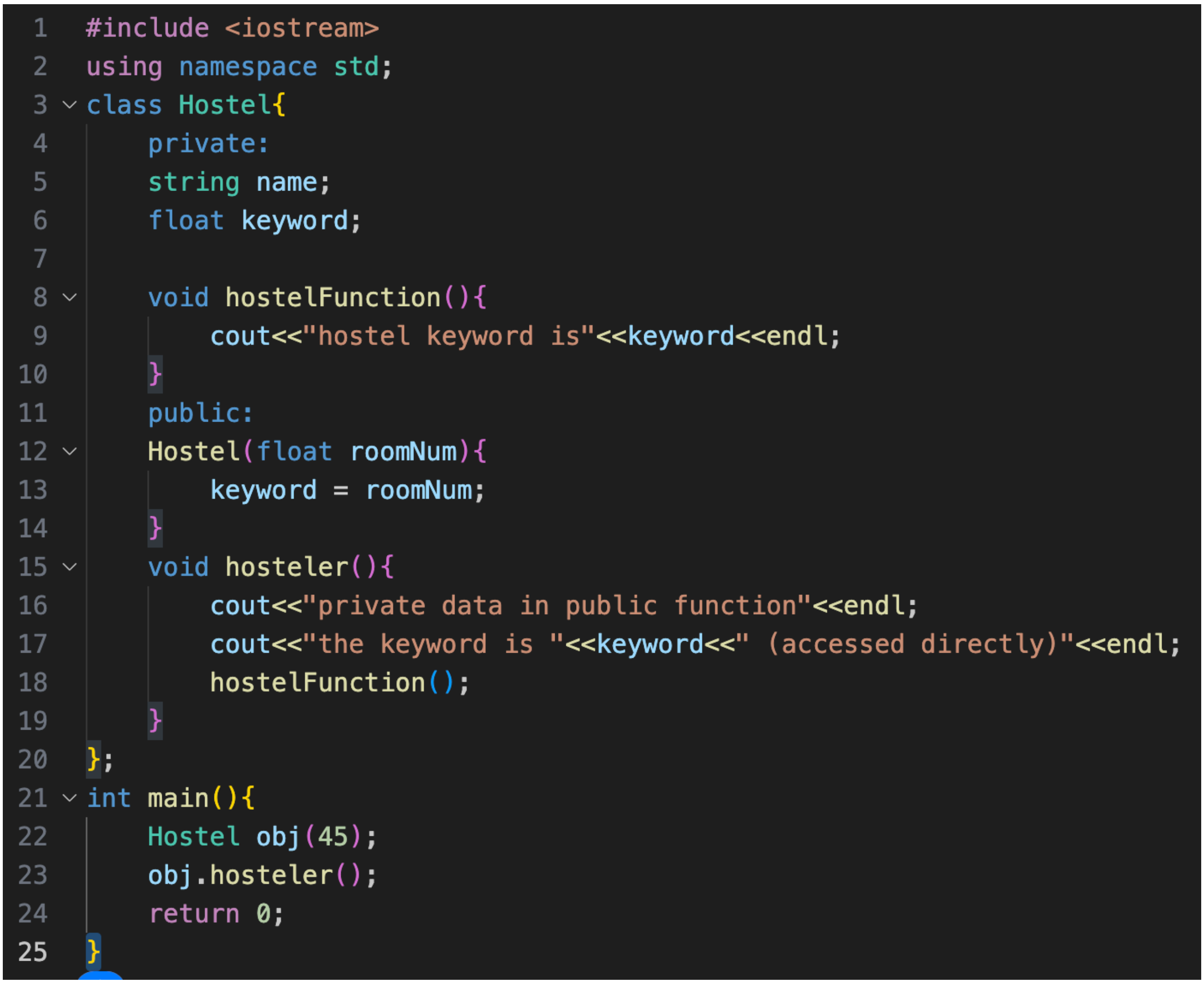
```cpp
    void updateCGPA() {
        cout << "Update CGPA: ";
        cin >> currentCGPA;
    }
    private:

    void updateHostel() {
        cout << "Update hostel: ";
        cin>>hostel;
    }
    public:

    void displayDetails() {
        cout << "\nStudent Details:" << endl;
        cout << "Name: " << name << endl;
        cout << "Roll Number: " << rollNo << endl;
        cout << "Degree: " << degree << endl;
        cout << "Hostel: " << hostel << endl;
        cout << "Current CGPA: " << currentCGPA << endl;
    }
};

int main() {
    Student student1;

    student1.addDetails();

    student1.displayDetails();

    cout << "\nUpdating student details...\n";
    student1.updateDetails();

    student1.displayDetails();

    cout << "\nUpdating CGPA...\n";
    student1.updateCGPA();

    student1.displayDetails();

    return 0;
}
```

OUTPUT:



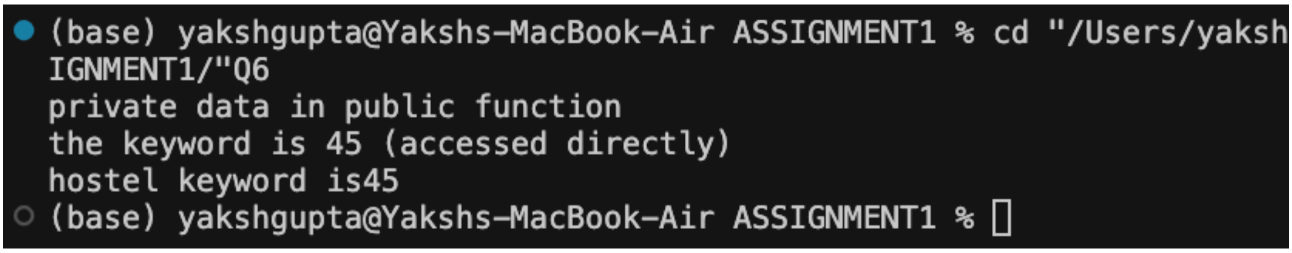
```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Des
  Enter student's name: Yaksh
  Enter roll number: 1024030280
  Enter degree: COE
  Enter hostel: B
  Enter current CGPA: 7.8

  Student Details:
  Name: Yaksh
  Roll Number: 1024030280
  Degree: COE
  Hostel: B
  Current CGPA: 7.8

  Updating student details...
  Update student's name: Yashh
  Update roll number: 12034
  Update degree: ECE
  Update hostel: J

  Student Details:
  Name: Yashh
  Roll Number: 12034
  Degree: ECE
  Hostel: J
  Current CGPA: 7.8

  Updating CGPA...
  Update CGPA: 8.2

  Student Details:
  Name: Yashh
  Roll Number: 12034
  Degree: ECE
  Hostel: J
  Current CGPA: 8.2
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % █
```

6. Create a code snippet that illustrates the following:
    a. Calling of private member functions inside public member function
    b. Access private member functions inside public member function



```cpp
#include <iostream>
using namespace std;
class Hostel{
    private:
    string name;
    float keyword;

    void hostelFunction(){
        cout<<"hostel keyword is"<<keyword<<endl;
    }
    public:
    Hostel(float roomNum){
        keyword = roomNum;
    }
    void hosteler(){
        cout<<"private data in public function"<<endl;
        cout<<"the keyword is "<<keyword<<" (accessed directly)"<<endl;
        hostelFunction();
    }
};
int main(){
    Hostel obj(45);
    obj.hosteler();
    return 0;
}
```

OUTPUT:

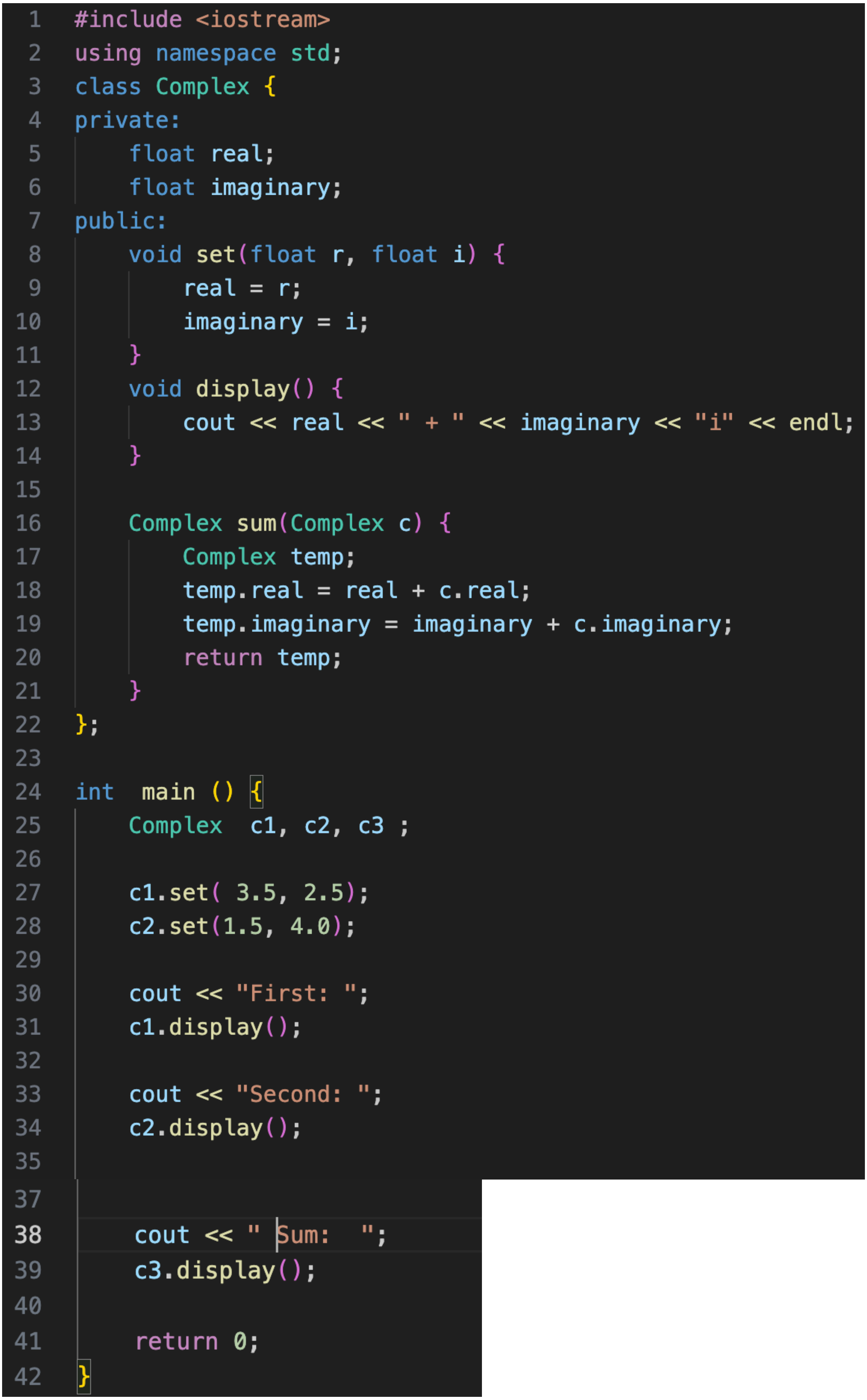


```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yaksh
  IGNMENT1/"Q6
  private data in public function
  the keyword is 45 (accessed directly)
  hostel keyword is45
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % ▯
```

7. Define a class named ***Complex*** with properties (real and imaginary) and methods as per following details.

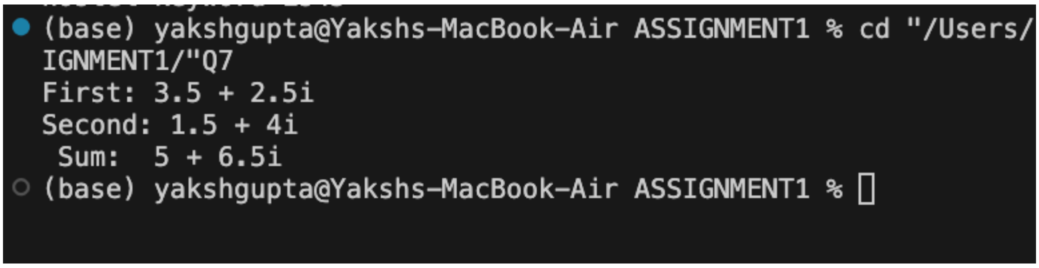
    void set () to initialize object values.
    void display () to display complex number.
    Complex sum (Complex) or ***void sum (Complex)*** to add two complex numbers (objects of Complex class) and ***return complex_number*** (object of Complex class) as result.

   Properties (real and imaginary) of the code should have private access modifier and member functions should have public access modifier in C++ class.



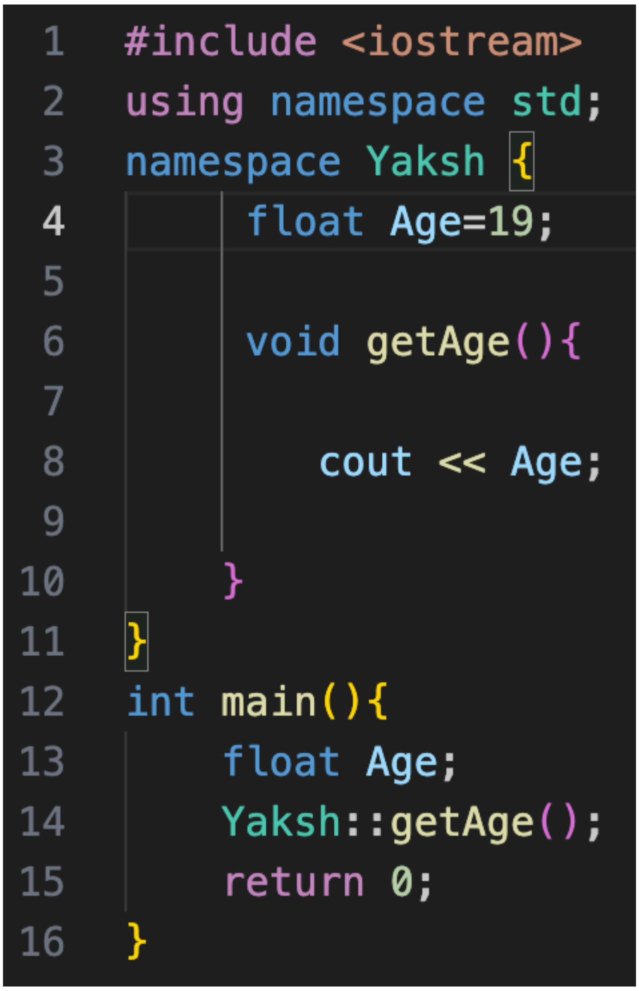
```cpp
#include <iostream>
using namespace std;
class Complex {
private:
    float real;
    float imaginary;
public:
    void set(float r, float i) {
        real = r;
        imaginary = i;
    }
    void display() {
        cout << real << " + " << imaginary << "i" << endl;
    }

    Complex sum(Complex c) {
        Complex temp;
        temp.real = real + c.real;
        temp.imaginary = imaginary + c.imaginary;
        return temp;
    }
};

int  main () {
    Complex  c1, c2, c3 ;

    c1.set( 3.5, 2.5);
    c2.set(1.5, 4.0);

    cout << "First: ";
    c1.display();

    cout << "Second: ";
    c2.display();


    cout << " Sum:   ";
    c3.display();

    return 0;
}
```

OUTPUT:

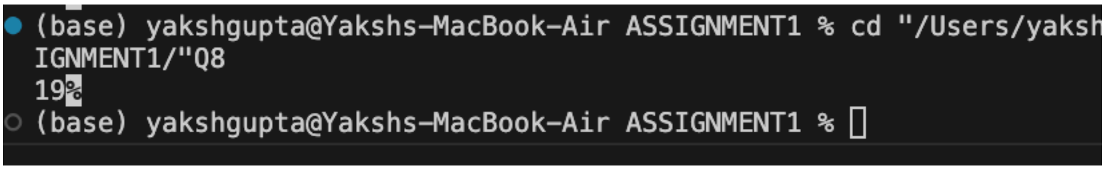


```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/
  IGNMENT1/"Q7
  First: 3.5 + 2.5i
  Second: 1.5 + 4i
   Sum:   5 + 6.5i
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % ▯
```

8. Implement *namespace* in a program to illustrate the use of same name variables and functions in different sections/libraries of the code.



```cpp
#include <iostream>
using namespace std;
namespace Yaksh {
        float Age=19;

        void getAge(){

            cout << Age;

        }
}
int main(){
      float Age;
      Yaksh::getAge();
      return 0;
}
```



```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yaksh
  IGNMENT1/"Q8
  19%
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % ▯
```