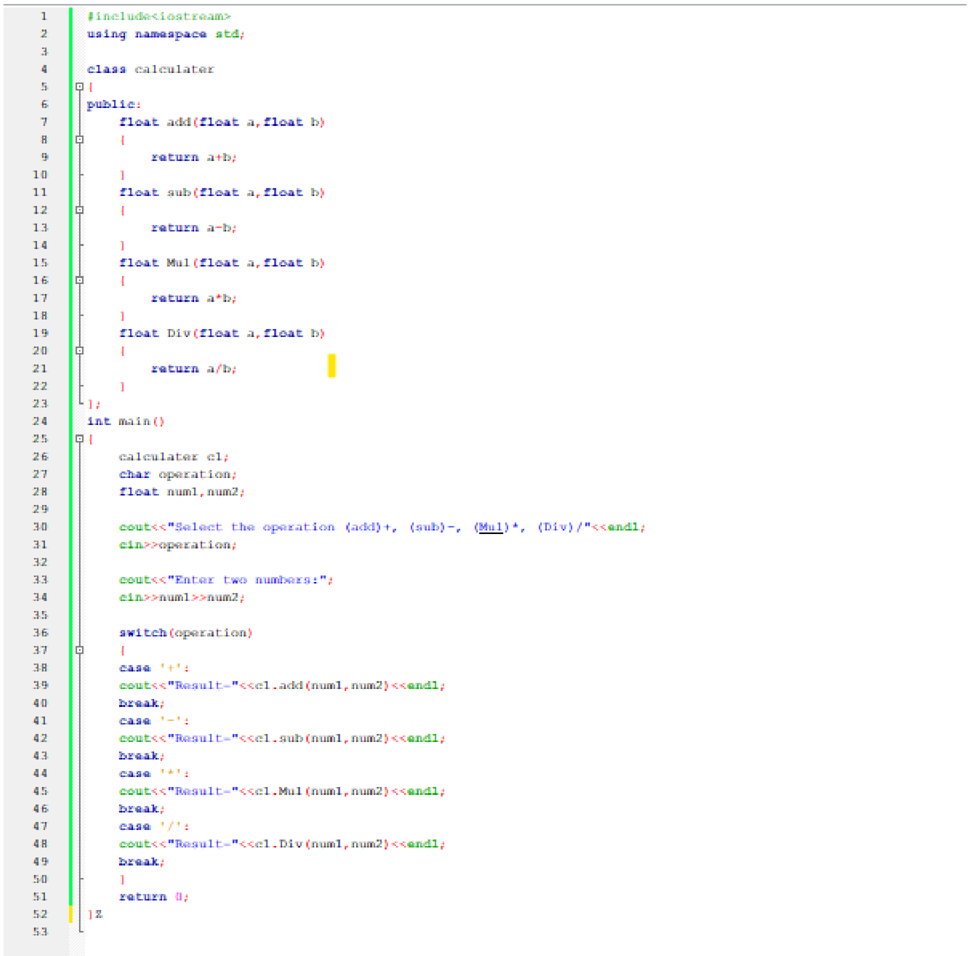
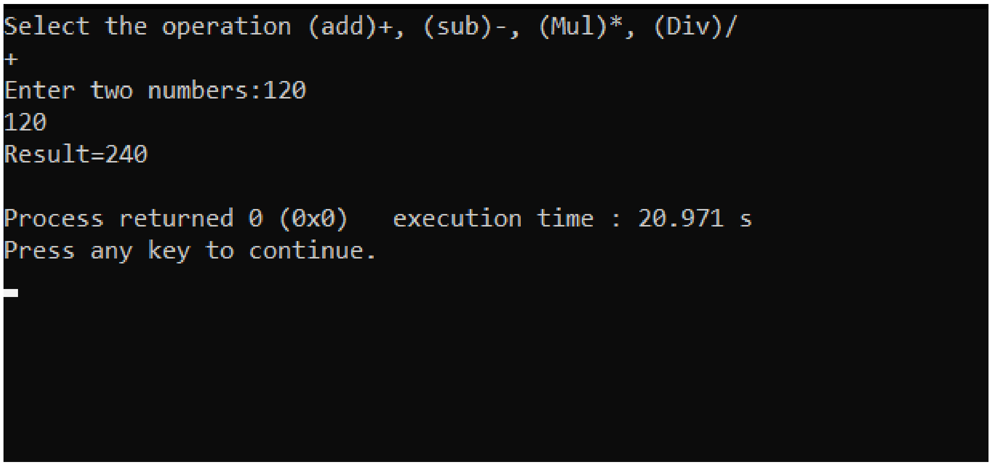
OOP_LAB_REPORT

NAME	AHMAD
RegNO	005
PROGRAM	BS(CS)
SWCTION	B

LAB TASK:-1

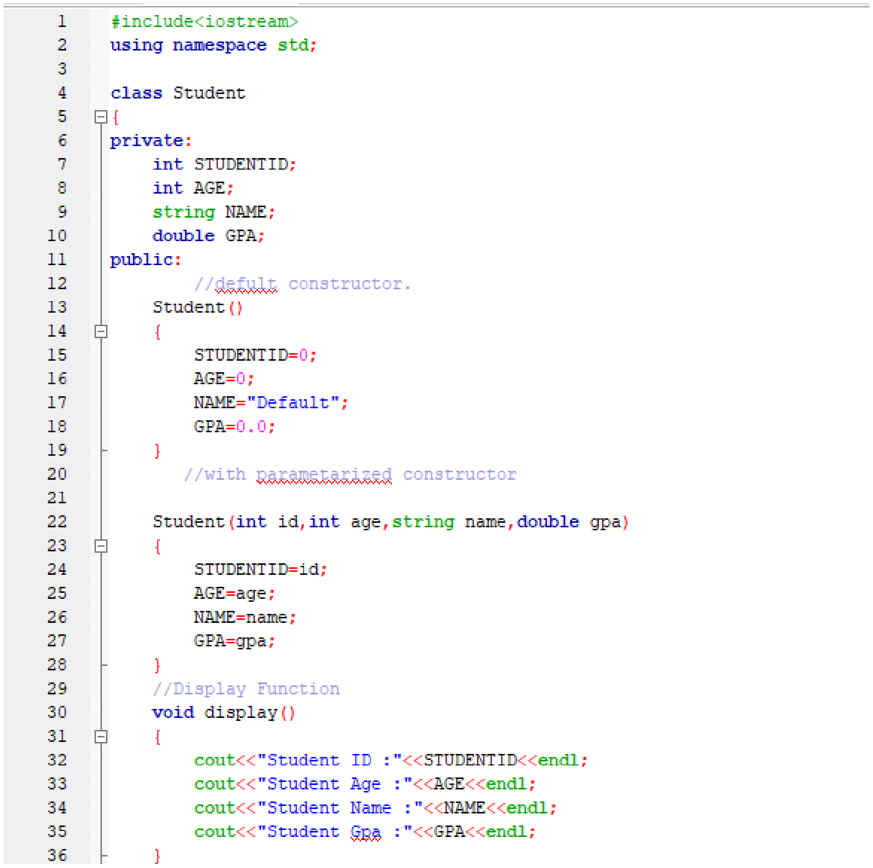


```cpp
#include<iostream>
using namespace std;

class calculater
{
public:
    float add(float a,float b)
    {
        return a+b;
    }
    float sub(float a,float b)
    {
        return a-b;
    }
    float Mul(float a,float b)
    {
        return a*b;
    }
    float Div(float a,float b)
    {
        return a/b;
    }
};
int main()
{
    calculater c1;
    char operation;
    float num1,num2;

    cout<<"Select the operation (add)+, (sub)-, (Mul)*, (Div)/"<<endl;
    cin>>operation;

    cout<<"Enter two numbers:";
    cin>>num1>>num2;

    switch(operation)
    {
    case '+':
    cout<<"Result="<<c1.add(num1,num2)<<endl;
    break;
    case '-':
    cout<<"Result="<<c1.sub(num1,num2)<<endl;
    break;
    case '*':
    cout<<"Result="<<c1.Mul(num1,num2)<<endl;
    break;
    case '/':
    cout<<"Result="<<c1.Div(num1,num2)<<endl;
    break;
    }
    return 0;
}Z
```

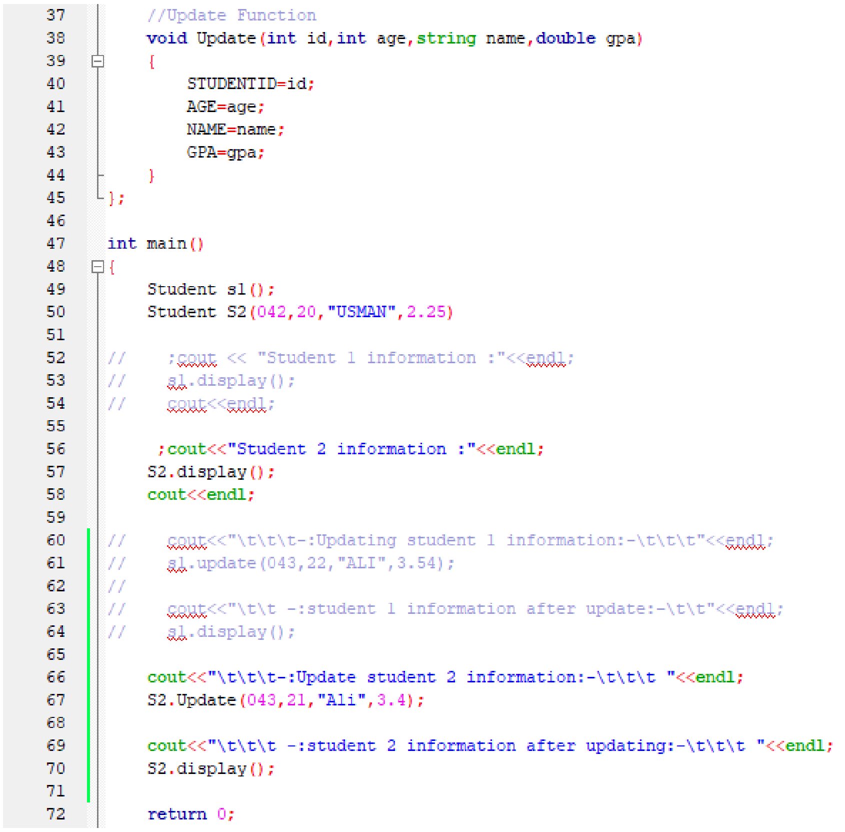


```
Select the operation (add)+, (sub)-, (Mul)*, (Div)/
+
Enter two numbers:120
120
Result=240

Process returned 0 (0x0)   execution time : 20.971 s
Press any key to continue.
```

LAB TASK:-2

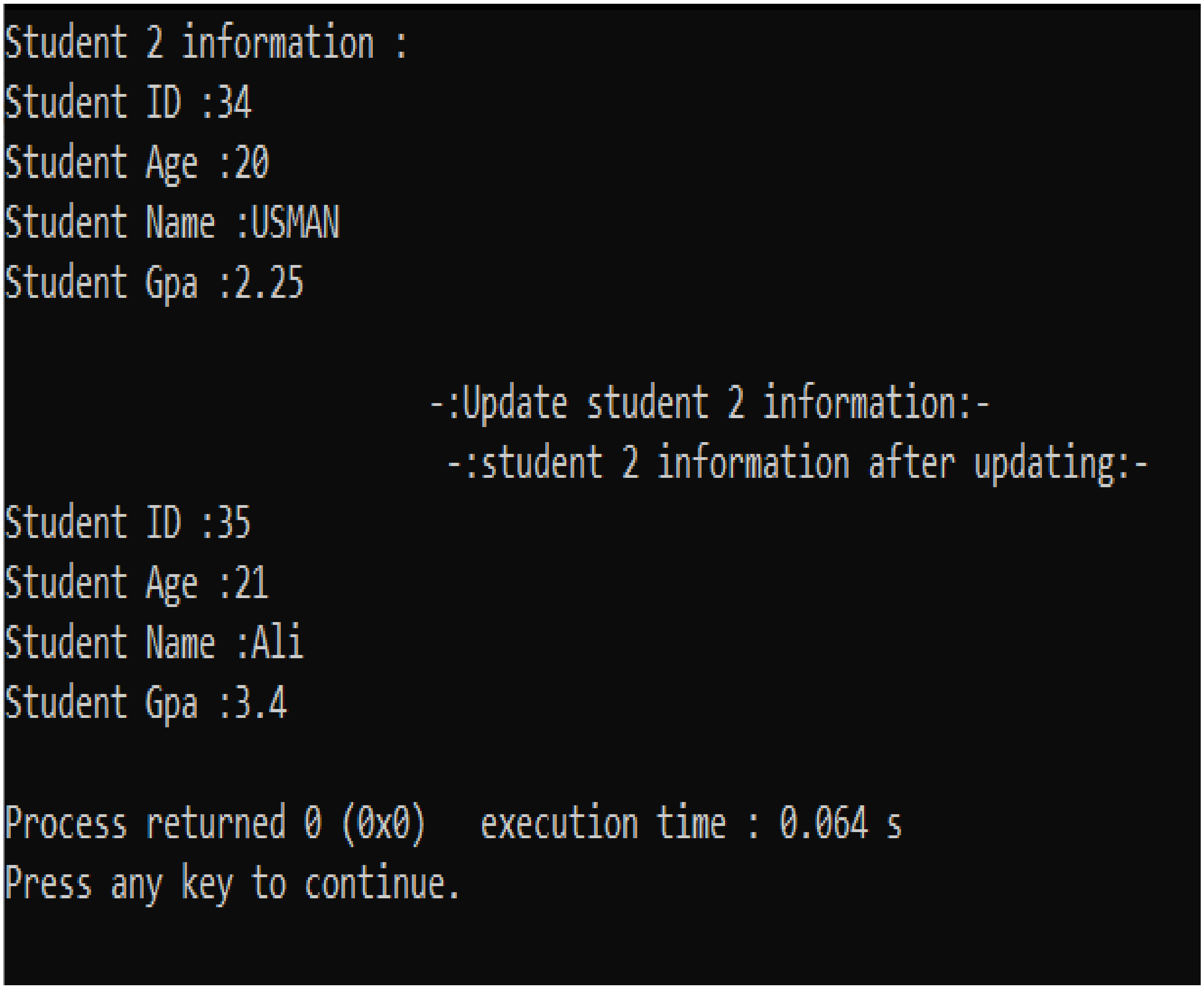


```
#include<iostream>
using namespace std;

class Student
{
private:
    int STUDENTID;
    int AGE;
    string NAME;
    double GPA;
public:
        //defult constructor.
    Student()
    {
        STUDENTID=0;
        AGE=0;
        NAME="Default";
        GPA=0.0;
    }
       //with parametarized constructor

    Student(int id,int age,string name,double gpa)
    {
        STUDENTID=id;
        AGE=age;
        NAME=name;
        GPA=gpa;
    }
    //Display Function
    void display()
    {
        cout<<"Student ID :"<<STUDENTID<<endl;
        cout<<"Student Age :"<<AGE<<endl;
        cout<<"Student Name :"<<NAME<<endl;
        cout<<"Student Gpa :"<<GPA<<endl;
    }
```



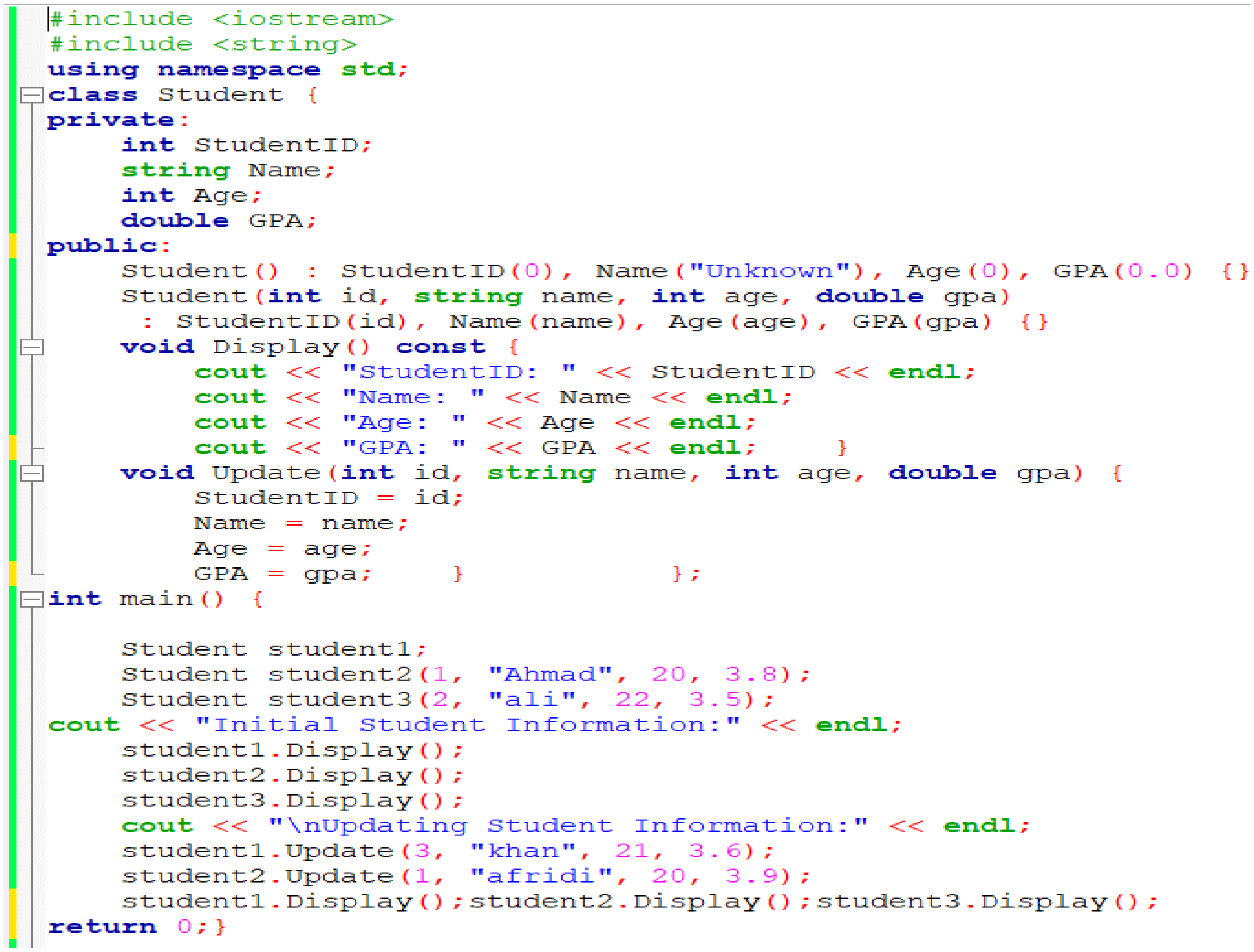
```
    //Update Function
    void Update(int id,int age,string name,double gpa)
    {
        STUDENTID=id;
        AGE=age;
        NAME=name;
        GPA=gpa;
    }
};

int main()
{
    Student s1();
    Student S2(042,20,"USMAN",2.25)

//    ;cout << "Student 1 information :"<<endl;
//    s1.display();
//    cout<<endl;

     ;cout<<"Student 2 information :"<<endl;
    S2.display();
    cout<<endl;

//    cout<<"\t\t\t-:Updating student 1 information:-\t\t\t"<<endl;
//    s1.update(043,22,"ALI",3.54);
//
//    cout<<"\t\t -:student 1 information after update:-\t\t"<<endl;
//    s1.display();

    cout<<"\t\t\t-:Update student 2 information:-\t\t\t "<<endl;
    S2.Update(043,21,"Ali",3.4);

    cout<<"\t\t\t -:student 2 information after updating:-\t\t\t "<<endl;
    S2.display();

    return 0;
```

2 result



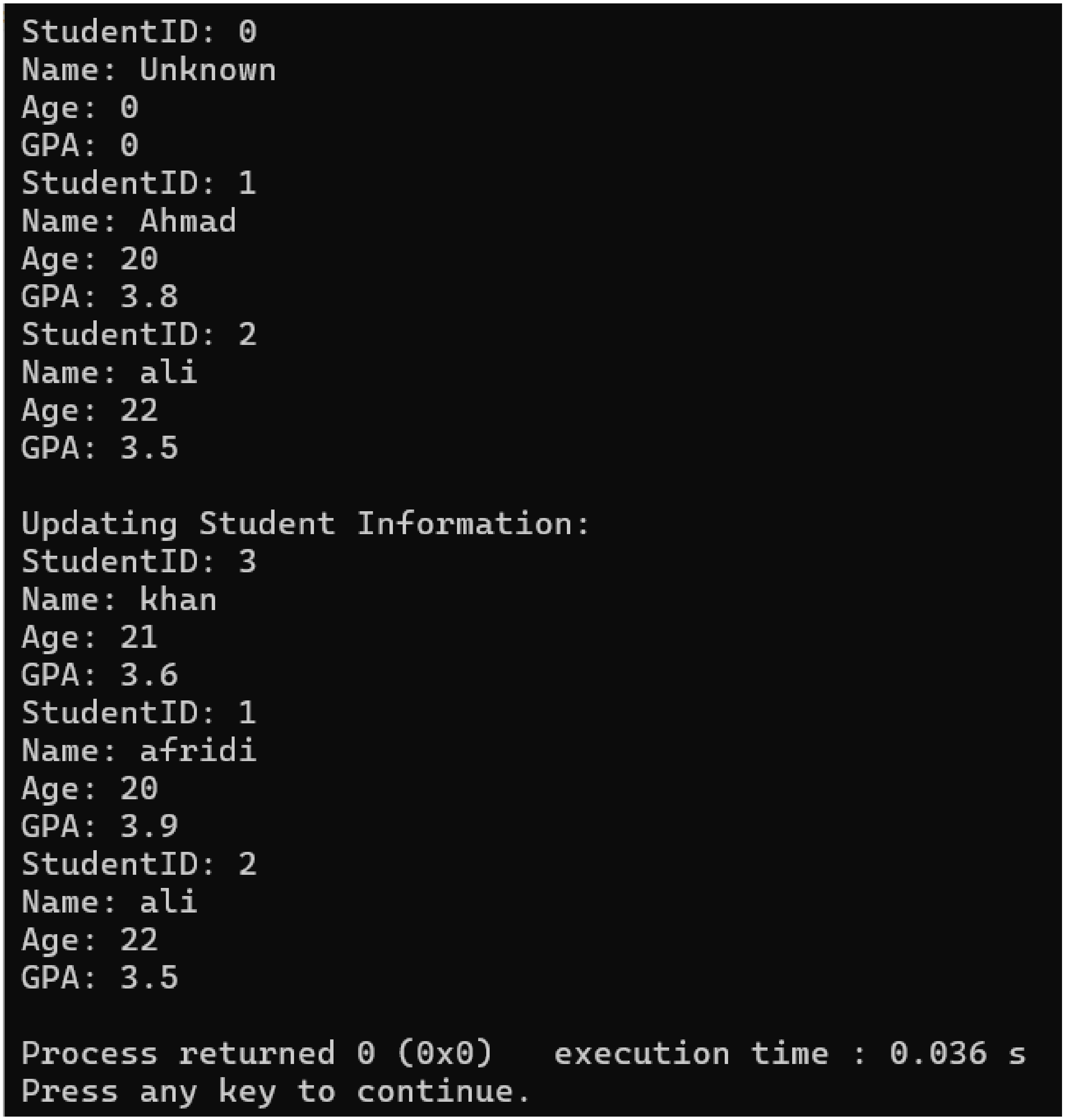
```
Student 2 information :
Student ID :34
Student Age :20
Student Name :USMAN
Student Gpa :2.25

                        -:Update student 2 information:-
                         -:student 2 information after updating:-
Student ID :35
Student Age :21
Student Name :Ali
Student Gpa :3.4

Process returned 0 (0x0)   execution time : 0.064 s
Press any key to continue.
```

LAB TASK:-3



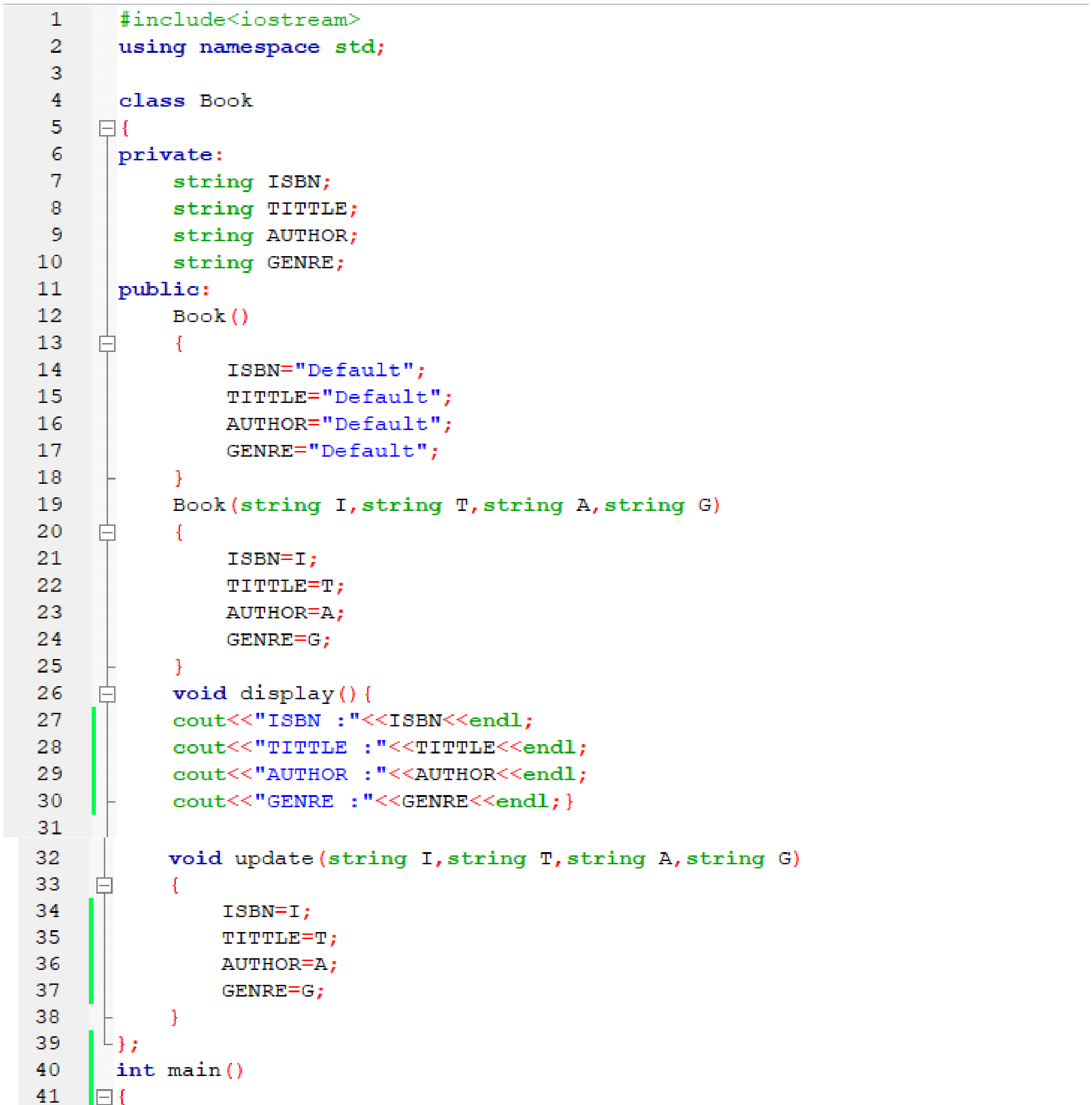
```cpp
#include <iostream>
#include <string>
using namespace std;
class Student {
private:
    int StudentID;
    string Name;
    int Age;
    double GPA;
public:
    Student() : StudentID(0), Name("Unknown"), Age(0), GPA(0.0) {}
    Student(int id, string name, int age, double gpa)
     : StudentID(id), Name(name), Age(age), GPA(gpa) {}
    void Display() const {
        cout << "StudentID: " << StudentID << endl;
        cout << "Name: " << Name << endl;
        cout << "Age: " << Age << endl;
        cout << "GPA: " << GPA << endl;     }
    void Update(int id, string name, int age, double gpa) {
        StudentID = id;
        Name = name;
        Age = age;
        GPA = gpa;     }            };
int main() {

    Student student1;
    Student student2(1, "Ahmad", 20, 3.8);
    Student student3(2, "ali", 22, 3.5);
cout << "Initial Student Information:" << endl;
    student1.Display();
    student2.Display();
    student3.Display();
    cout << "\nUpdating Student Information:" << endl;
    student1.Update(3, "khan", 21, 3.6);
    student2.Update(1, "afridi", 20, 3.9);
    student1.Display();student2.Display();student3.Display();
return 0;}
```

3 result



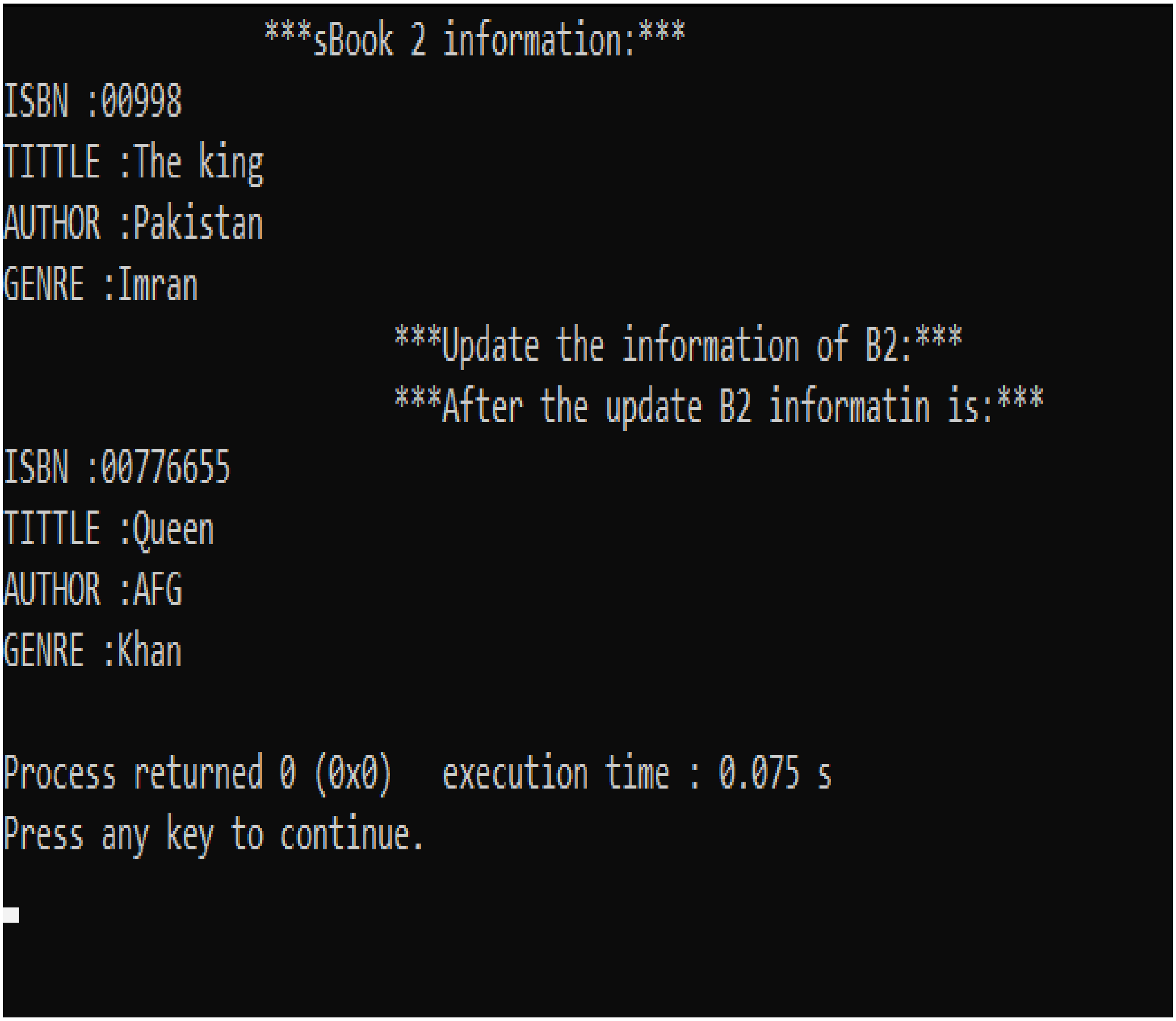
```
StudentID: 0
Name: Unknown
Age: 0
GPA: 0
StudentID: 1
Name: Ahmad
Age: 20
GPA: 3.8
StudentID: 2
Name: ali
Age: 22
GPA: 3.5

Updating Student Information:
StudentID: 3
Name: khan
Age: 21
GPA: 3.6
StudentID: 1
Name: afridi
Age: 20
GPA: 3.9
StudentID: 2
Name: ali
Age: 22
GPA: 3.5

Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```

LAB TASK:-4



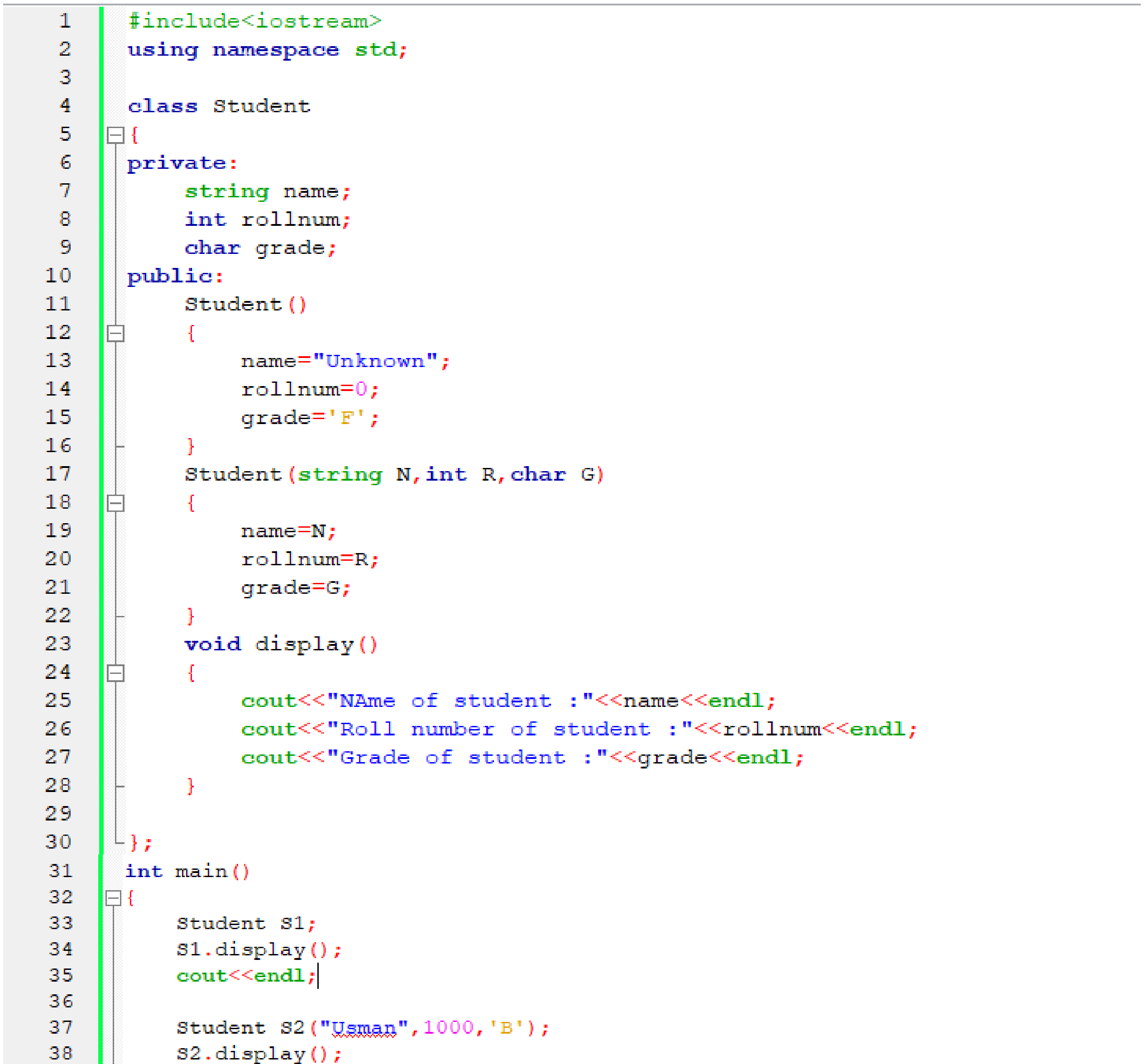
```
#include<iostream>
using namespace std;

class Book
{
private:
    string ISBN;
    string TITTLE;
    string AUTHOR;
    string GENRE;
public:
    Book()
    {
        ISBN="Default";
        TITTLE="Default";
        AUTHOR="Default";
        GENRE="Default";
    }
    Book(string I,string T,string A,string G)
    {
        ISBN=I;
        TITTLE=T;
        AUTHOR=A;
        GENRE=G;
    }
    void display(){
    cout<<"ISBN :"<<ISBN<<endl;
    cout<<"TITTLE :"<<TITTLE<<endl;
    cout<<"AUTHOR :"<<AUTHOR<<endl;
    cout<<"GENRE :"<<GENRE<<endl;}

    void update(string I,string T,string A,string G)
    {
        ISBN=I;
        TITTLE=T;
        AUTHOR=A;
        GENRE=G;
    }
};
int main()
{
```

4 result



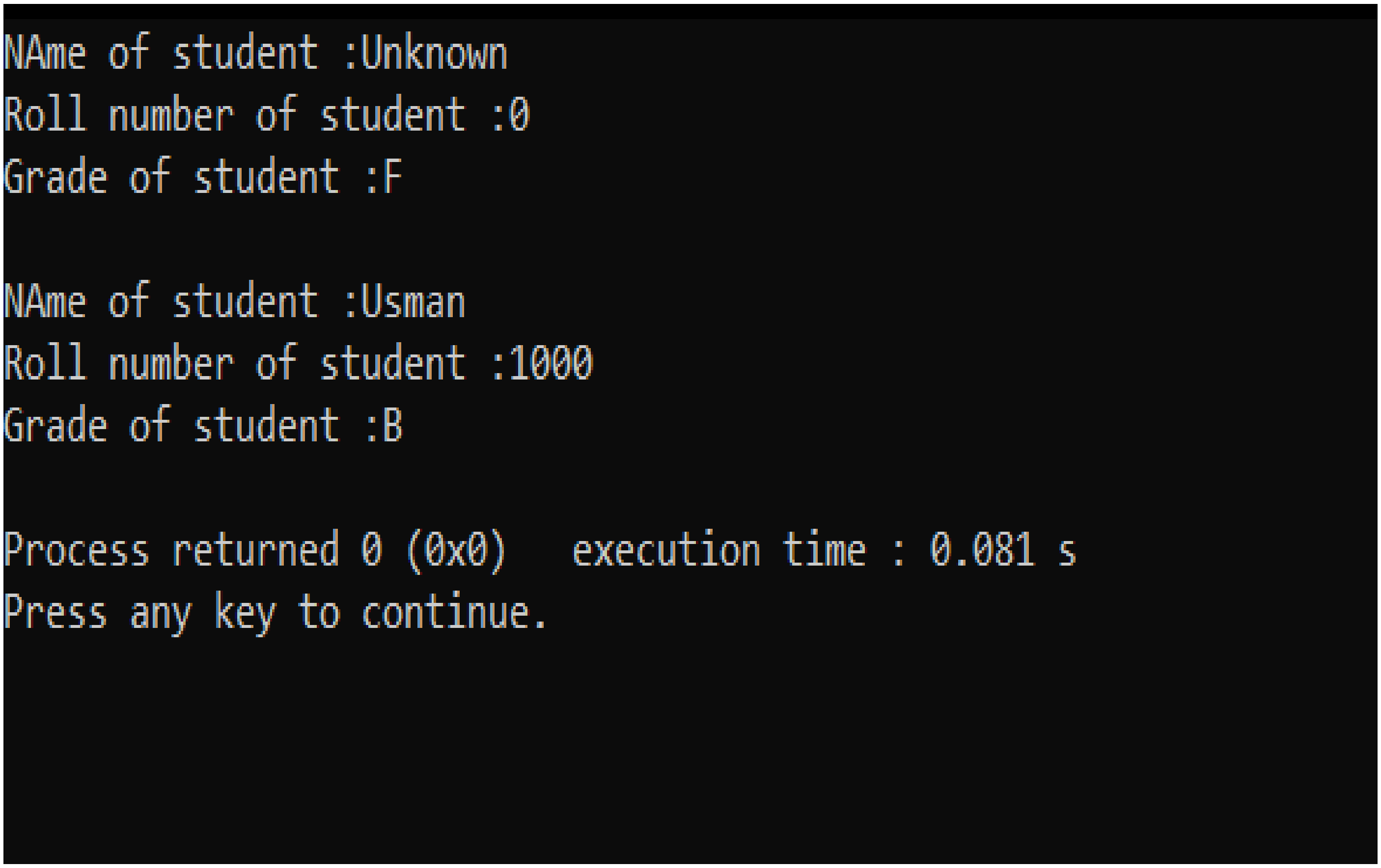
```
                ***sBook 2 information:***
ISBN :00998
TITTLE :The king
AUTHOR :Pakistan
GENRE :Imran
                        ***Update the information of B2:***
                        ***After the update B2 informatin is:***
ISBN :00776655
TITTLE :Queen
AUTHOR :AFG
GENRE :Khan

Process returned 0 (0x0)   execution time : 0.075 s
Press any key to continue.
```

LAB TASK:-5



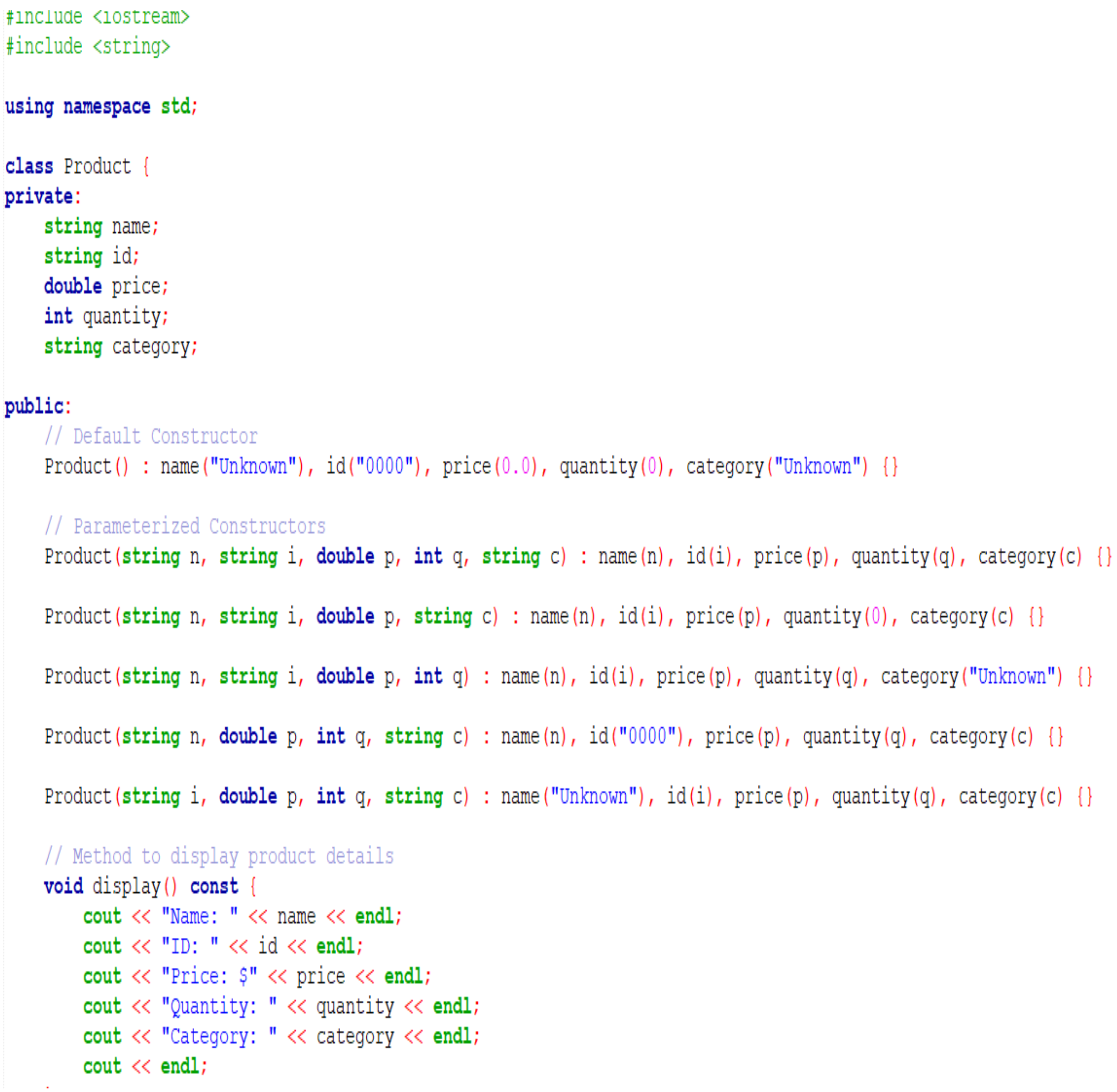
```
#include<iostream>
using namespace std;

class Student
{
private:
    string name;
    int rollnum;
    char grade;
public:
    Student()
    {
        name="Unknown";
        rollnum=0;
        grade='F';
    }
    Student(string N,int R,char G)
    {
        name=N;
        rollnum=R;
        grade=G;
    }
    void display()
    {
        cout<<"NAme of student :"<<name<<endl;
        cout<<"Roll number of student :"<<rollnum<<endl;
        cout<<"Grade of student :"<<grade<<endl;
    }

};
int main()
{
    Student S1;
    S1.display();
    cout<<endl;

    Student S2("Usman",1000,'B');
    S2.display();
```

5 result

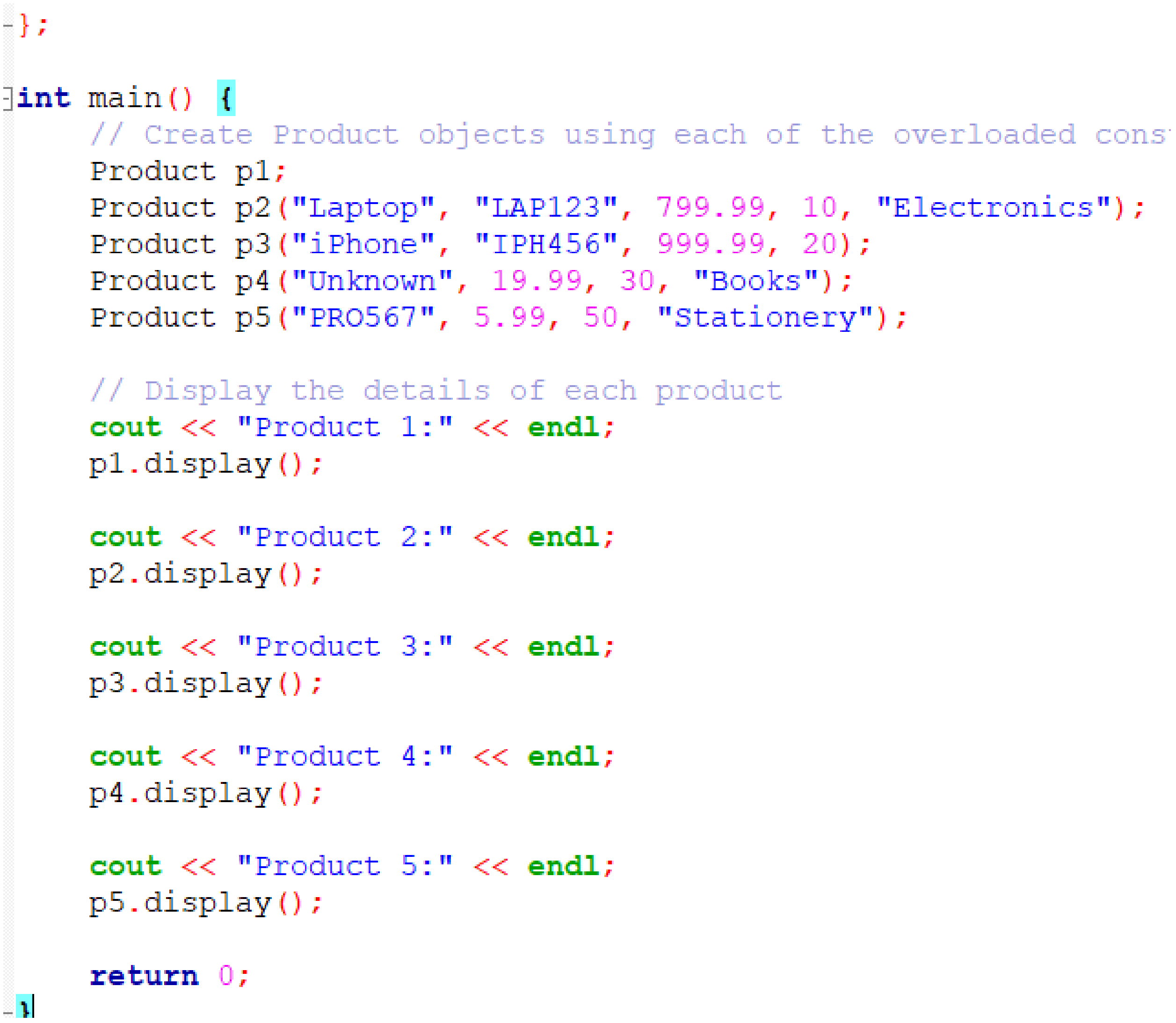


```
NAme of student :Unknown
Roll number of student :0
Grade of student :F

NAme of student :Usman
Roll number of student :1000
Grade of student :B

Process returned 0 (0x0)   execution time : 0.081 s
Press any key to continue.
```

LAB TASK:-6

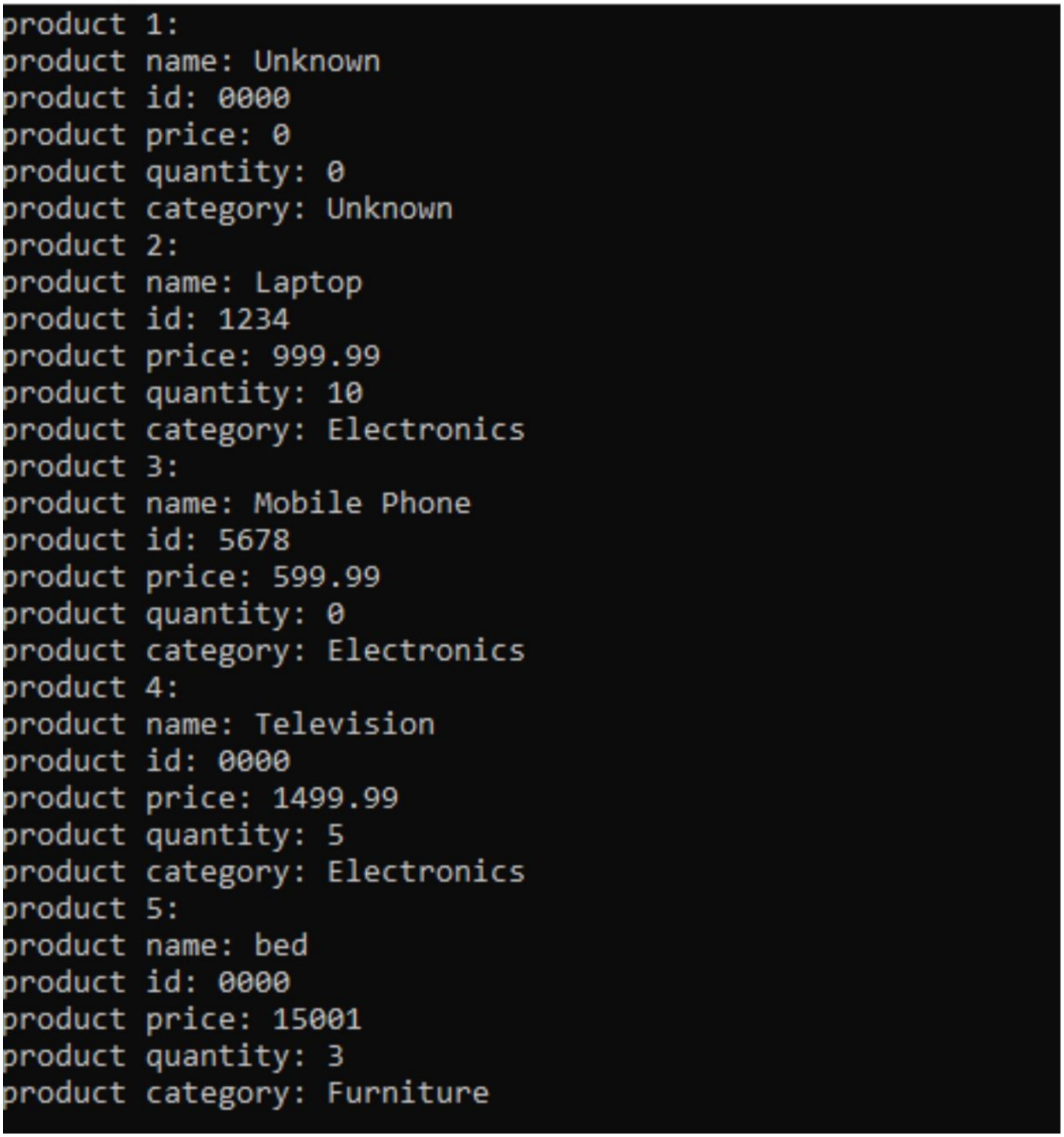


```
#include <iostream>
#include <string>

using namespace std;

class Product {
private:
    string name;
    string id;
    double price;
    int quantity;
    string category;

public:
    // Default Constructor
    Product() : name("Unknown"), id("0000"), price(0.0), quantity(0), category("Unknown") {}

    // Parameterized Constructors
    Product(string n, string i, double p, int q, string c) : name(n), id(i), price(p), quantity(q), category(c) {}

    Product(string n, string i, double p, string c) : name(n), id(i), price(p), quantity(0), category(c) {}

    Product(string n, string i, double p, int q) : name(n), id(i), price(p), quantity(q), category("Unknown") {}

    Product(string n, double p, int q, string c) : name(n), id("0000"), price(p), quantity(q), category(c) {}

    Product(string i, double p, int q, string c) : name("Unknown"), id(i), price(p), quantity(q), category(c) {}

    // Method to display product details
    void display() const {
        cout << "Name: " << name << endl;
        cout << "ID: " << id << endl;
        cout << "Price: $" << price << endl;
        cout << "Quantity: " << quantity << endl;
        cout << "Category: " << category << endl;
        cout << endl;
```



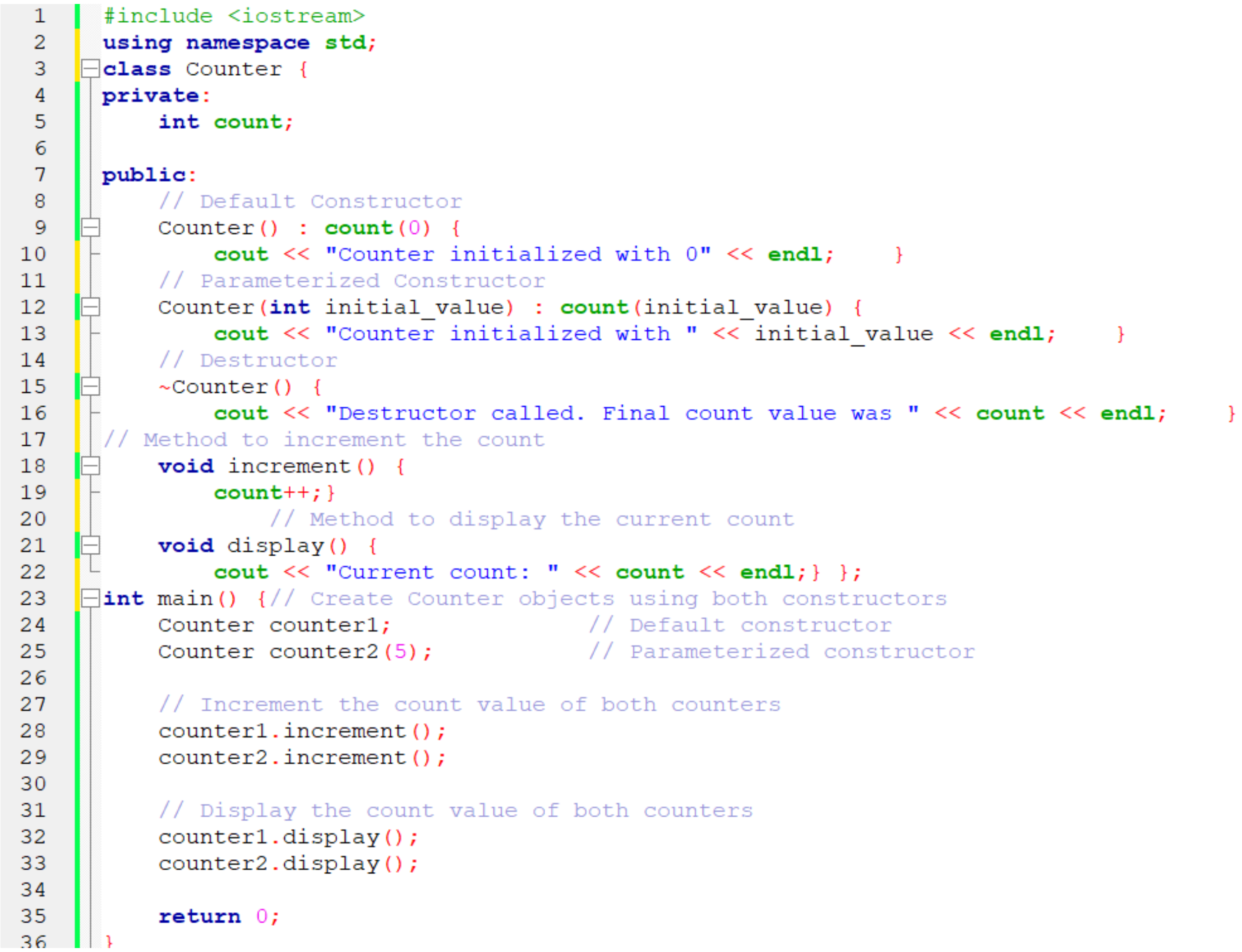
```
};

int main() {
    // Create Product objects using each of the overloaded cons
    Product p1;
    Product p2("Laptop", "LAP123", 799.99, 10, "Electronics");
    Product p3("iPhone", "IPH456", 999.99, 20);
    Product p4("Unknown", 19.99, 30, "Books");
    Product p5("PRO567", 5.99, 50, "Stationery");

    // Display the details of each product
    cout << "Product 1:" << endl;
    p1.display();

    cout << "Product 2:" << endl;
    p2.display();

    cout << "Product 3:" << endl;
    p3.display();

    cout << "Product 4:" << endl;
    p4.display();

    cout << "Product 5:" << endl;
    p5.display();

    return 0;
}
```

6 Result

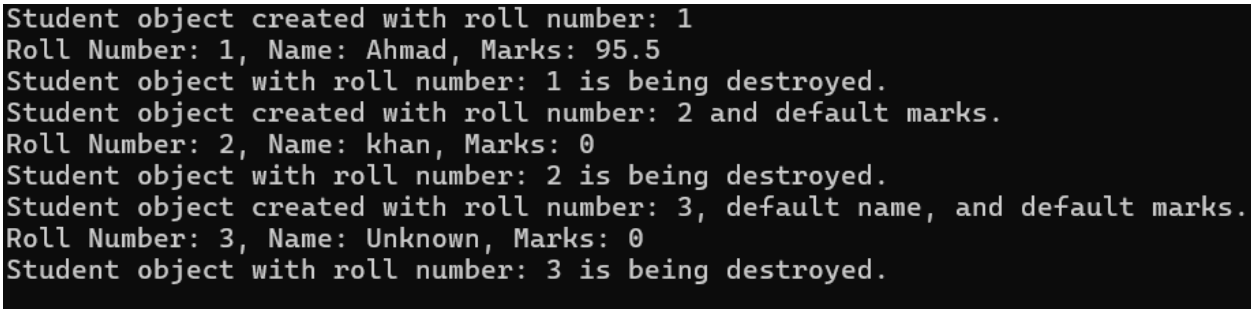


```
product 1:
product name: Unknown
product id: 0000
product price: 0
product quantity: 0
product category: Unknown
product 2:
product name: Laptop
product id: 1234
product price: 999.99
product quantity: 10
product category: Electronics
product 3:
product name: Mobile Phone
product id: 5678
product price: 599.99
product quantity: 0
product category: Electronics
product 4:
product name: Television
product id: 0000
product price: 1499.99
product quantity: 5
product category: Electronics
product 5:
product name: bed
product id: 0000
product price: 15001
product quantity: 3
product category: Furniture
```

Lab Task:-7

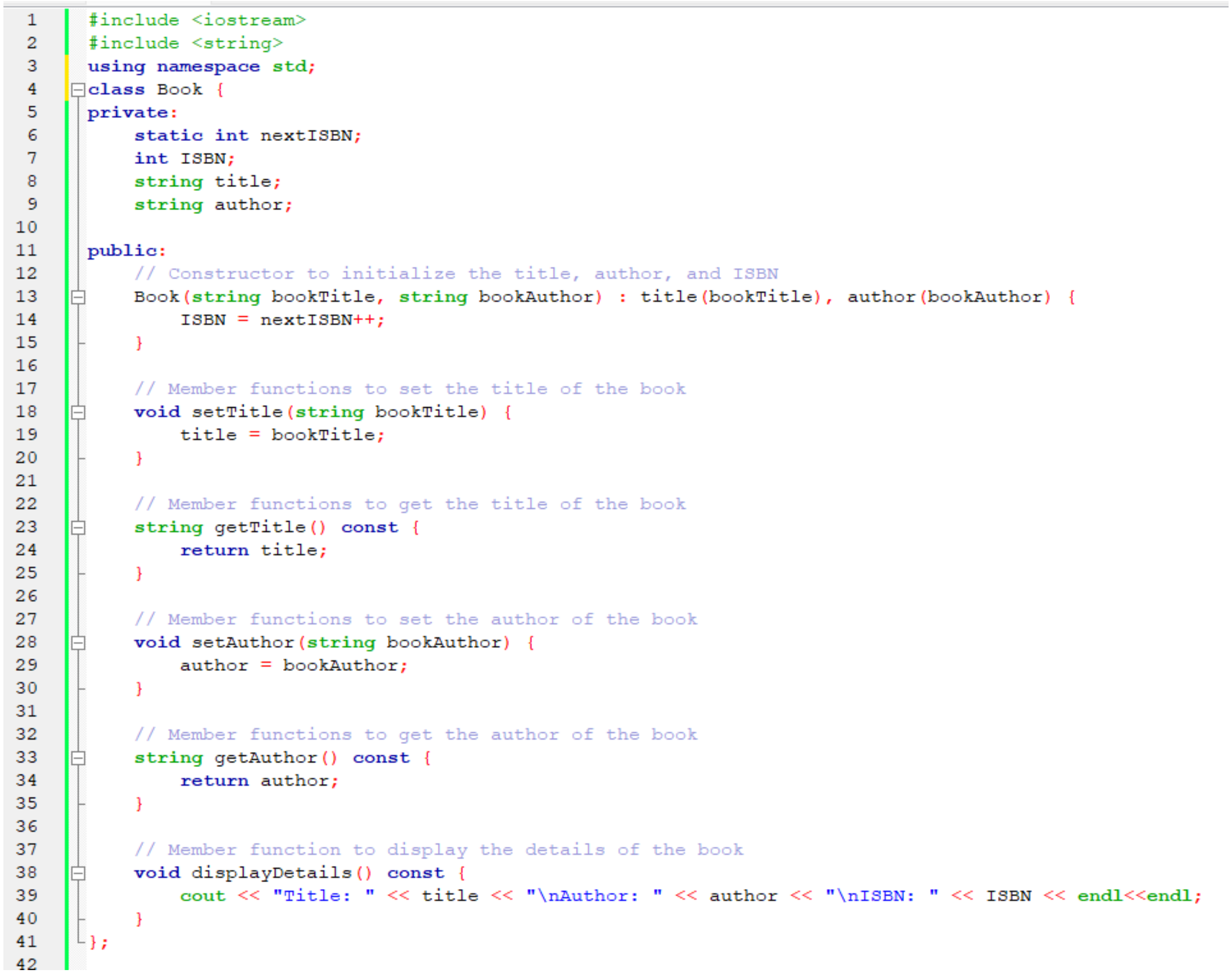


```cpp
#include <iostream>
using namespace std;
class Counter {
private:
    int count;

public:
    // Default Constructor
    Counter() : count(0) {
        cout << "Counter initialized with 0" << endl;     }
    // Parameterized Constructor
    Counter(int initial_value) : count(initial_value) {
        cout << "Counter initialized with " << initial_value << endl;     }
    // Destructor
    ~Counter() {
        cout << "Destructor called. Final count value was " << count << endl;     }
// Method to increment the count
    void increment() {
        count++;}
            // Method to display the current count
    void display() {
        cout << "Current count: " << count << endl;} };
int main() {// Create Counter objects using both constructors
    Counter counter1;                  // Default constructor
    Counter counter2(5);               // Parameterized constructor

    // Increment the count value of both counters
    counter1.increment();
    counter2.increment();

    // Display the count value of both counters
    counter1.display();
    counter2.display();

    return 0;
}
```



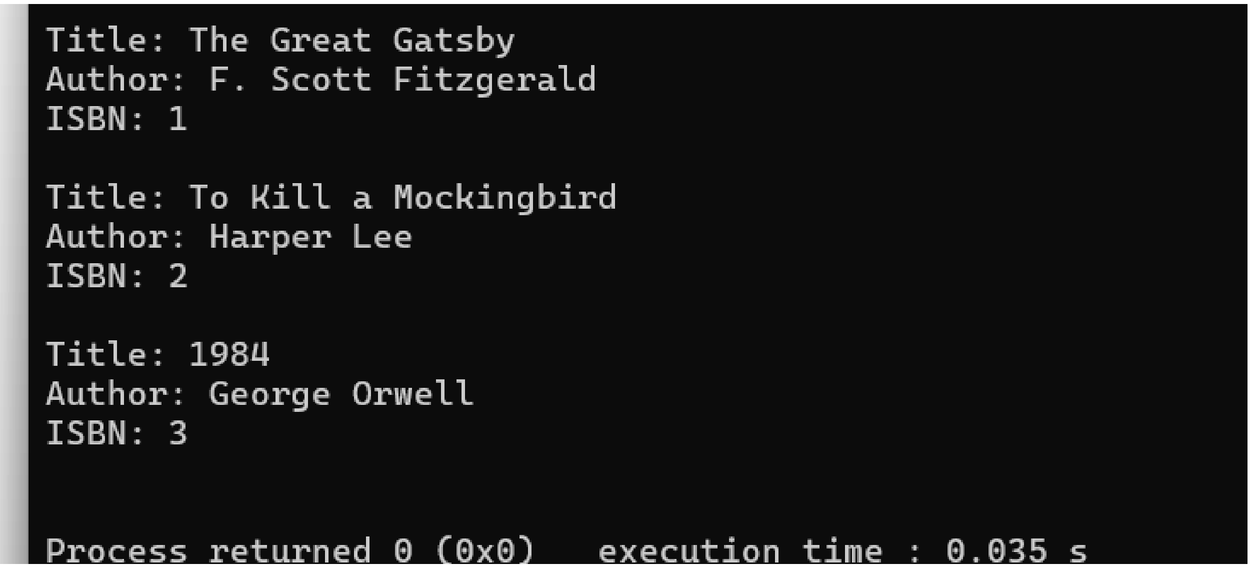
```
Student object created with roll number: 1
Roll Number: 1, Name: Ahmad, Marks: 95.5
Student object with roll number: 1 is being destroyed.
Student object created with roll number: 2 and default marks.
Roll Number: 2, Name: khan, Marks: 0
Student object with roll number: 2 is being destroyed.
Student object created with roll number: 3, default name, and default marks.
Roll Number: 3, Name: Unknown, Marks: 0
Student object with roll number: 3 is being destroyed.
```

Lab Task:-8



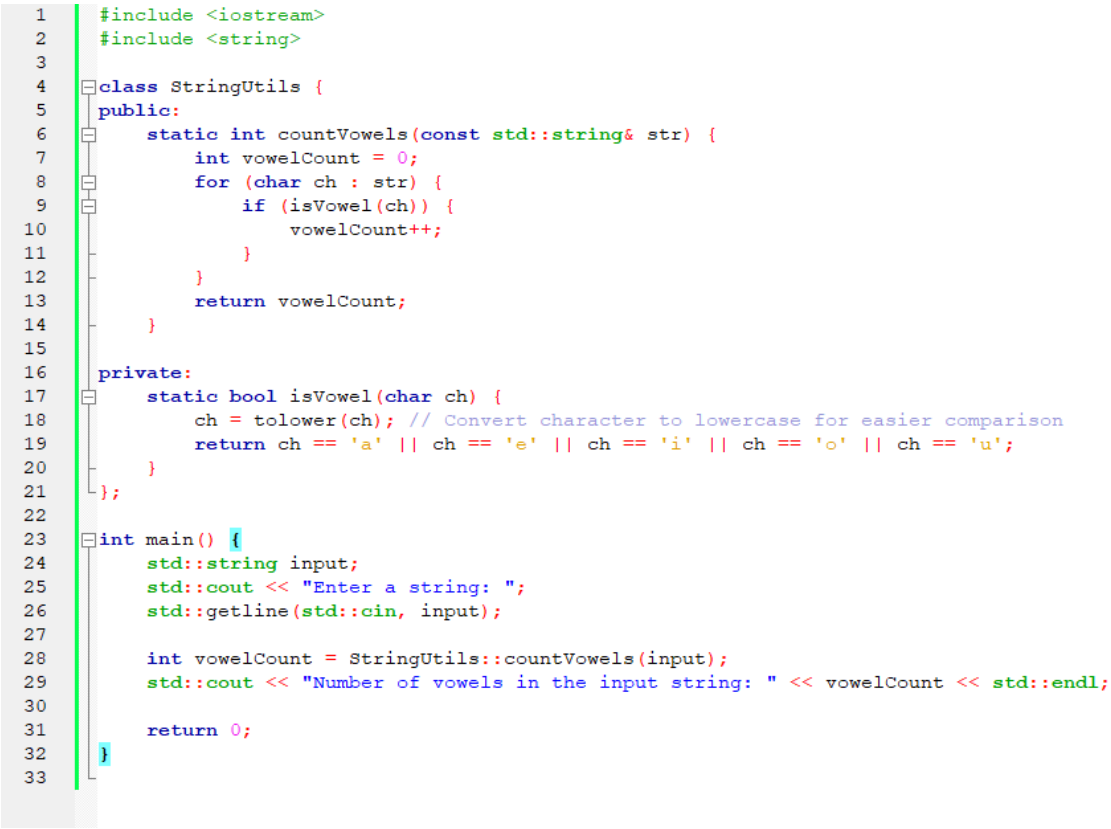
```cpp
#include <iostream>
#include <string>
using namespace std;
class Book {
private:
    static int nextISBN;
    int ISBN;
    string title;
    string author;

public:
    // Constructor to initialize the title, author, and ISBN
    Book(string bookTitle, string bookAuthor) : title(bookTitle), author(bookAuthor) {
        ISBN = nextISBN++;
    }

    // Member functions to set the title of the book
    void setTitle(string bookTitle) {
        title = bookTitle;
    }

    // Member functions to get the title of the book
    string getTitle() const {
        return title;
    }

    // Member functions to set the author of the book
    void setAuthor(string bookAuthor) {
        author = bookAuthor;
    }

    // Member functions to get the author of the book
    string getAuthor() const {
        return author;
    }

    // Member function to display the details of the book
    void displayDetails() const {
        cout << "Title: " << title << "\nAuthor: " << author << "\nISBN: " << ISBN << endl<<endl;
    }
};

```

```cpp
// Initialize static member
int Book::nextISBN = 1;

int main() {
    // Create three book objects
    Book book1("The Great Gatsby", "F. Scott Fitzgerald");
    Book book2("To Kill a Mockingbird", "Harper Lee");
    Book book3("1984", "George Orwell");

    // Display the details of each book
    book1.displayDetails();
    book2.displayDetails();
    book3.displayDetails();

    return 0;
}
```

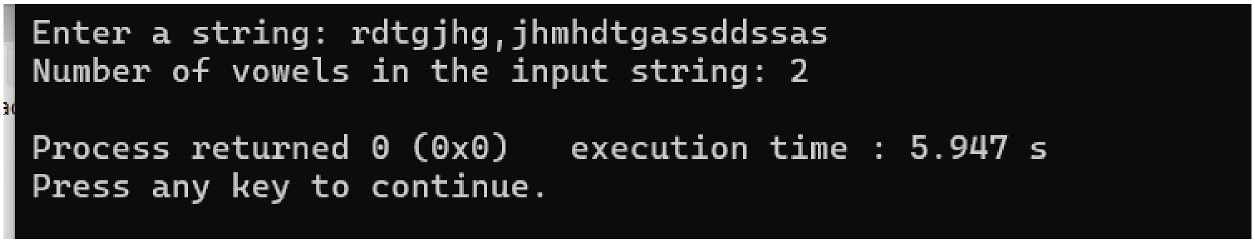


```
Title: The Great Gatsby
Author: F. Scott Fitzgerald
ISBN: 1

Title: To Kill a Mockingbird
Author: Harper Lee
ISBN: 2

Title: 1984
Author: George Orwell
ISBN: 3


Process returned 0 (0x0)   execution time : 0.035 s
```

Lab Task :-9

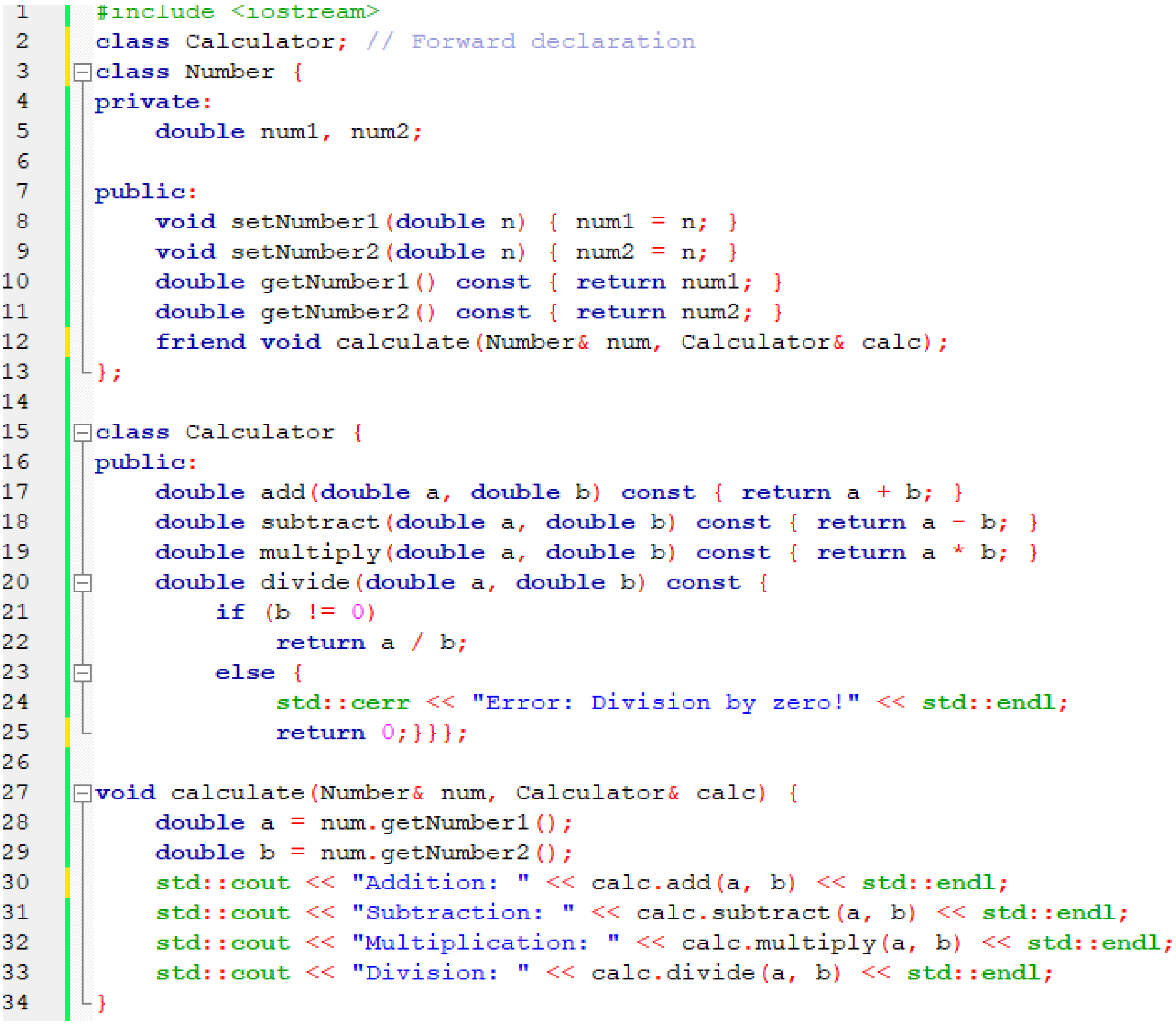


```cpp
#include <iostream>
#include <string>

class StringUtils {
public:
    static int countVowels(const std::string& str) {
        int vowelCount = 0;
        for (char ch : str) {
            if (isVowel(ch)) {
                vowelCount++;
            }
        }
        return vowelCount;
    }

private:
    static bool isVowel(char ch) {
        ch = tolower(ch); // Convert character to lowercase for easier comparison
        return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
    }
};

int main() {
    std::string input;
    std::cout << "Enter a string: ";
    std::getline(std::cin, input);

    int vowelCount = StringUtils::countVowels(input);
    std::cout << "Number of vowels in the input string: " << vowelCount << std::endl;

    return 0;
}
```

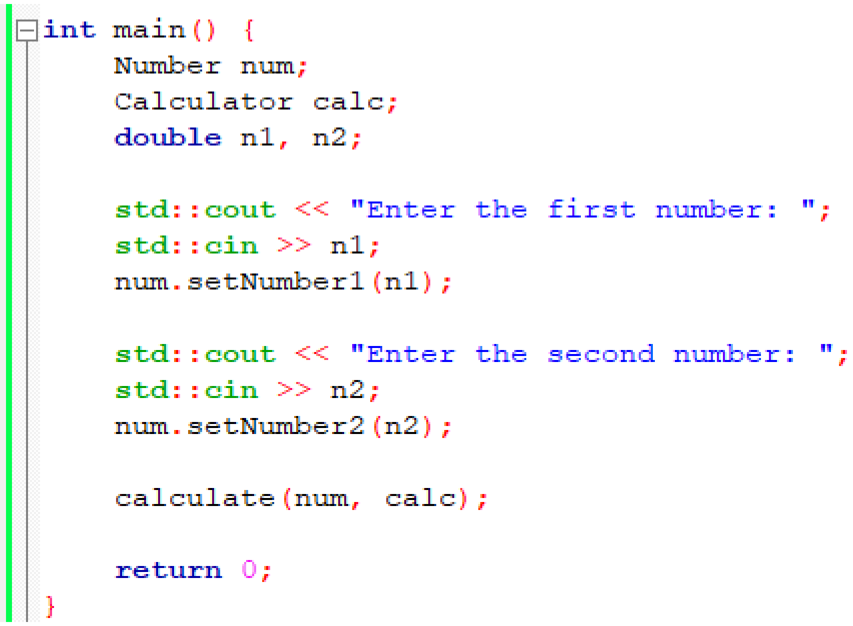


```
Enter a string: rdtgjhg,jhmhdtgassddssas
Number of vowels in the input string: 2

Process returned 0 (0x0)   execution time : 5.947 s
Press any key to continue.
```

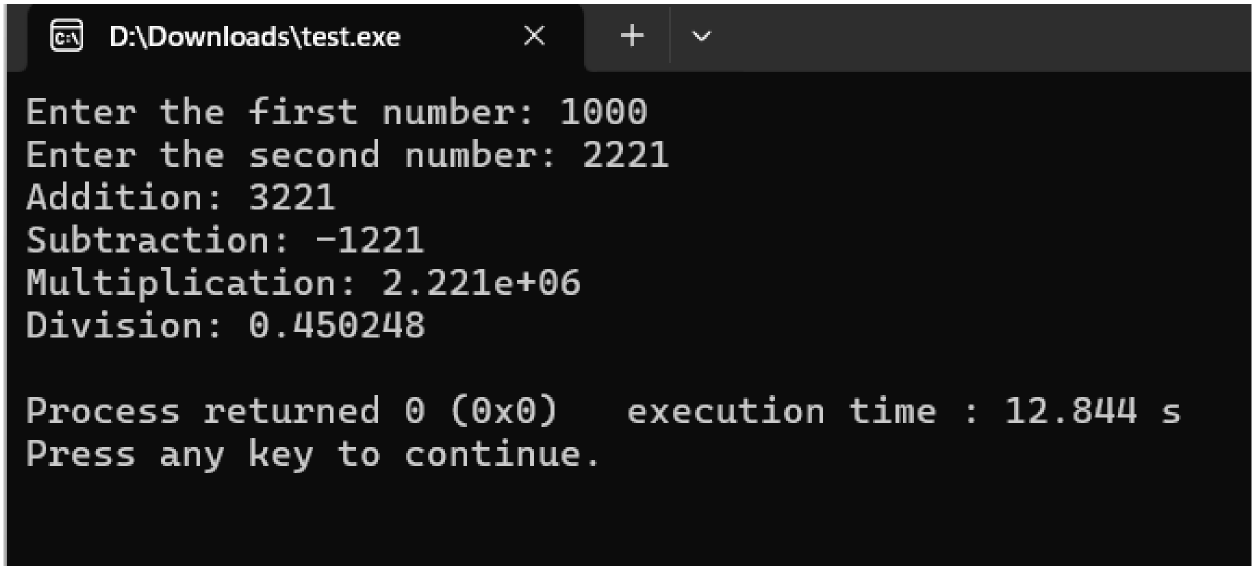
Lab Task :-10



```cpp
#include <iostream>
class Calculator; // Forward declaration
class Number {
private:
    double num1, num2;

public:
    void setNumber1(double n) { num1 = n; }
    void setNumber2(double n) { num2 = n; }
    double getNumber1() const { return num1; }
    double getNumber2() const { return num2; }
    friend void calculate(Number& num, Calculator& calc);
};

class Calculator {
public:
    double add(double a, double b) const { return a + b; }
    double subtract(double a, double b) const { return a - b; }
    double multiply(double a, double b) const { return a * b; }
    double divide(double a, double b) const {
        if (b != 0)
            return a / b;
        else {
            std::cerr << "Error: Division by zero!" << std::endl;
            return 0;}}};

void calculate(Number& num, Calculator& calc) {
    double a = num.getNumber1();
    double b = num.getNumber2();
    std::cout << "Addition: " << calc.add(a, b) << std::endl;
    std::cout << "Subtraction: " << calc.subtract(a, b) << std::endl;
    std::cout << "Multiplication: " << calc.multiply(a, b) << std::endl;
    std::cout << "Division: " << calc.divide(a, b) << std::endl;
}
```



```
int main() {
    Number num;
    Calculator calc;
    double n1, n2;

    std::cout << "Enter the first number: ";
    std::cin >> n1;
    num.setNumber1(n1);

    std::cout << "Enter the second number: ";
    std::cin >> n2;
    num.setNumber2(n2);

    calculate(num, calc);

    return 0;
}
```



```
Enter the first number: 1000
Enter the second number: 2221
Addition: 3221
Subtraction: -1221
Multiplication: 2.221e+06
Division: 0.450248

Process returned 0 (0x0)   execution time : 12.844 s
Press any key to continue.
```