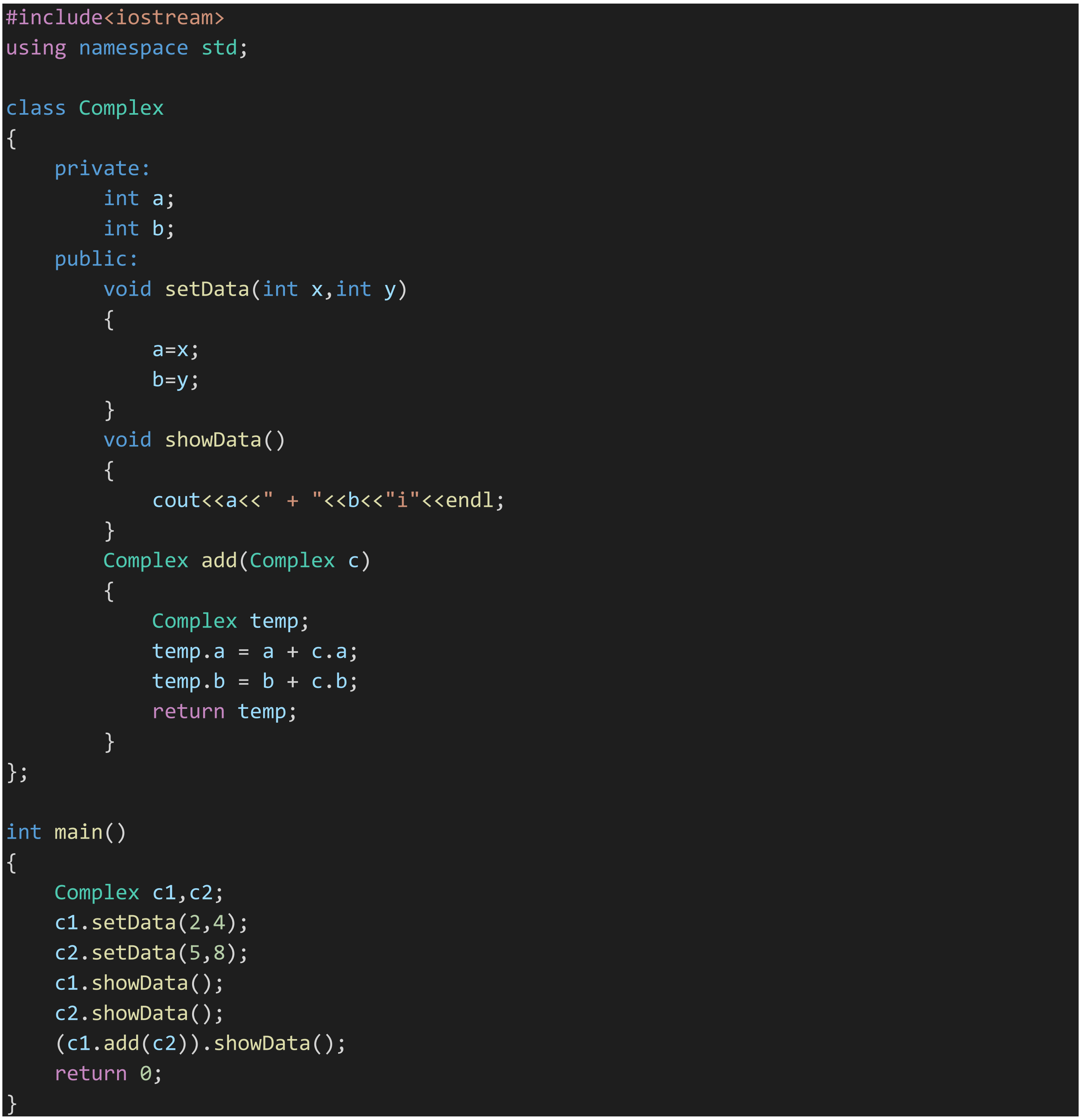
Assignment – 26

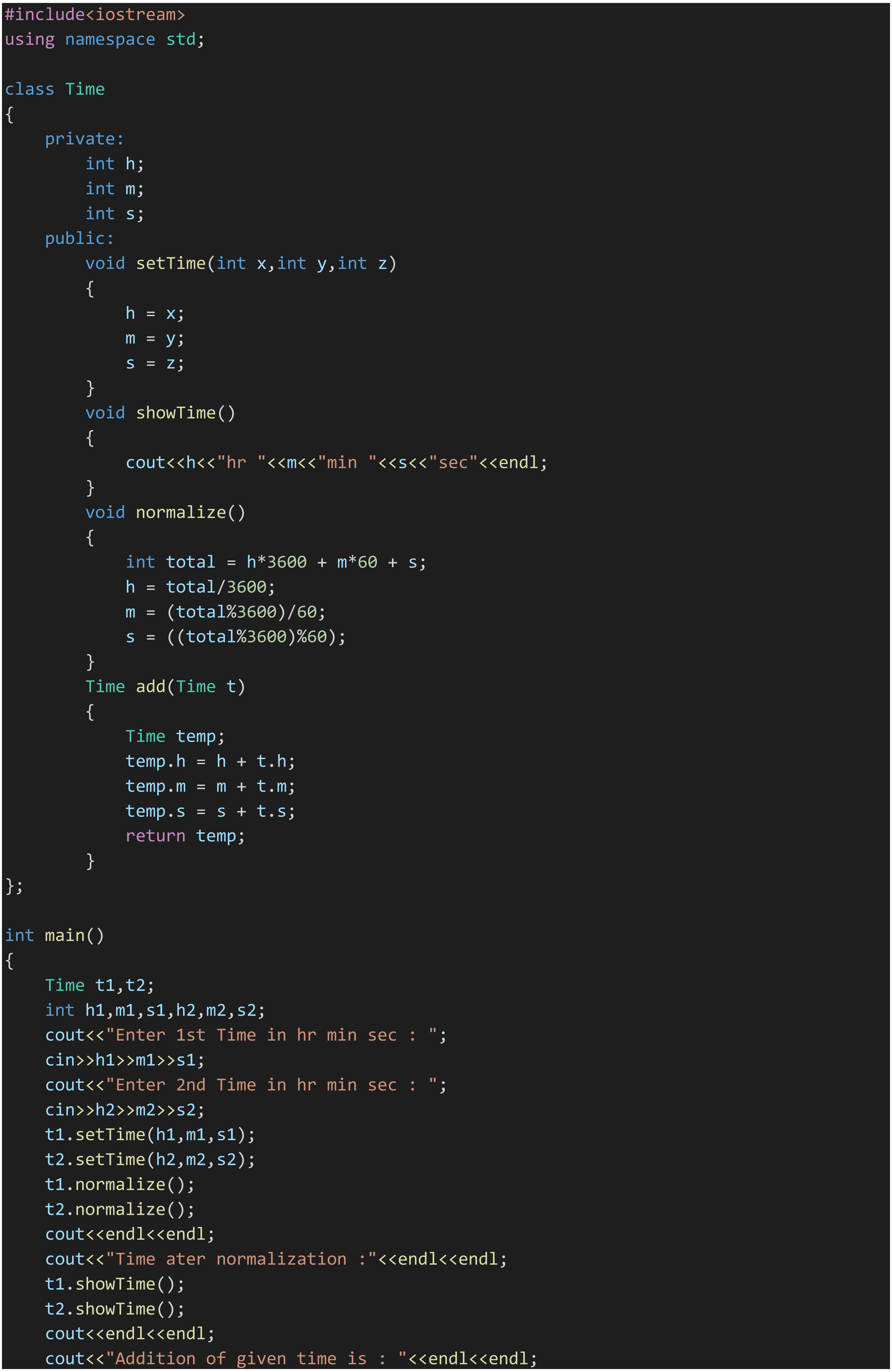
1. Define a class Complex to represent a complex number with instance variables a and b to store real and imaginary parts. Also define following member functions
   a. void setData(int,int)
   b. void showData()
   c. Complex add(Complex)



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

class Complex
{
    private:
        int a;
        int b;
    public:
        void setData(int x,int y)
        {
            a=x;
            b=y;
        }
        void showData()
        {
            cout<<a<<" + "<<b<<"i"<<endl;
        }
        Complex add(Complex c)
        {
            Complex temp;
            temp.a = a + c.a;
            temp.b = b + c.b;
            return temp;
        }
};

int main()
{
    Complex c1,c2;
    c1.setData(2,4);
    c2.setData(5,8);
    c1.showData();
    c2.showData();
    (c1.add(c2)).showData();
    return 0;
}
```

2. Define a class Time to represent a time with instance variables h,m and s to store hour, minute and second. Also define following member functions
   a. void setTime(int,int,int)
   b. void showTime()
   c. void normalize()

d. Time add(Time)



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

class Time
{
    private:
        int h;
        int m;
        int s;
    public:
        void setTime(int x,int y,int z)
        {
            h = x;
            m = y;
            s = z;
        }
        void showTime()
        {
            cout<<h<<"hr "<<m<<"min "<<s<<"sec"<<endl;
        }
        void normalize()
        {
            int total = h*3600 + m*60 + s;
            h = total/3600;
            m = (total%3600)/60;
            s = ((total%3600)%60);
        }
        Time add(Time t)
        {
            Time temp;
            temp.h = h + t.h;
            temp.m = m + t.m;
            temp.s = s + t.s;
            return temp;
        }
};

int main()
{
    Time t1,t2;
    int h1,m1,s1,h2,m2,s2;
    cout<<"Enter 1st Time in hr min sec : ";
    cin>>h1>>m1>>s1;
    cout<<"Enter 2nd Time in hr min sec : ";
    cin>>h2>>m2>>s2;
    t1.setTime(h1,m1,s1);
    t2.setTime(h2,m2,s2);
    t1.normalize();
    t2.normalize();
    cout<<endl<<endl;
    cout<<"Time ater normalization :"<<endl<<endl;
    t1.showTime();
    t2.showTime();
    cout<<endl<<endl;
    cout<<"Addition of given time is : "<<endl<<endl;
    (t1.add(t2)).showTime();
    cout<<endl;
    return 0;
}
```

3. Define a class Cube and calculate Volume of Cube and initialise it using constructor.

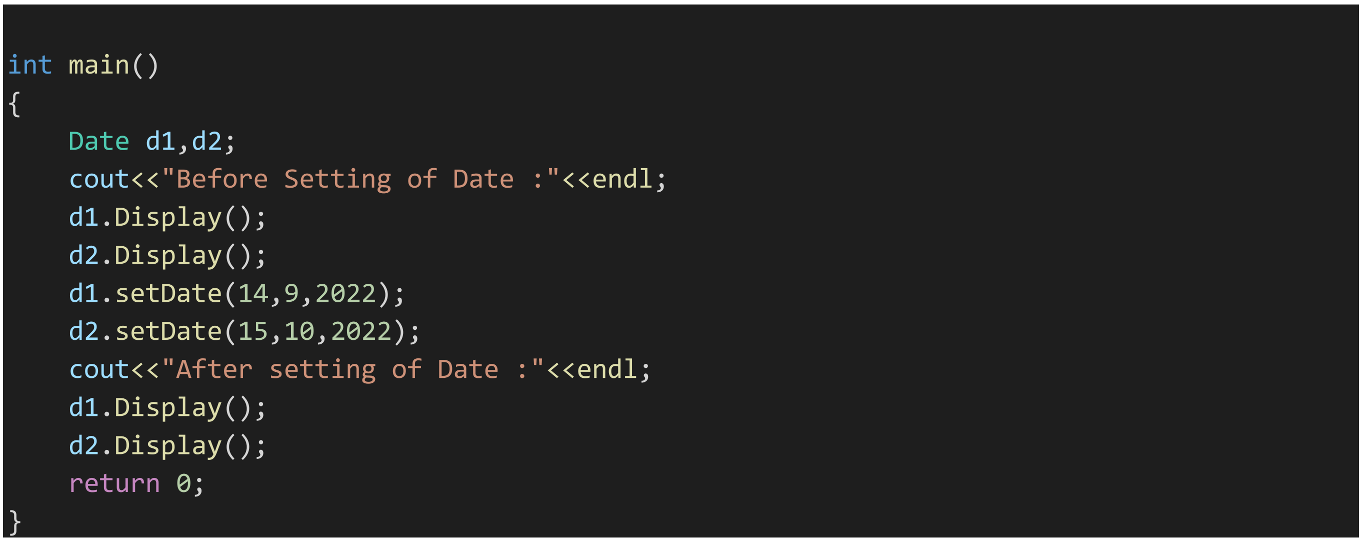
```
#include<iostream>
using namespace std;

class Cube
{
    private:
        float a;
    public:
        Cube(int x)
        {
            a = x;
        }
        void volume()
        {
            cout<<"Volume of Cube is : "<<a*a*a<<endl;
        }
};

int main()
{
    Cube c1(5),c2(10);
    c1.volume();
    c2.volume();
    return 0;
}
```

4. Define a class Counter and Write a program to Show Counter using Constructor.

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

class Counter
{
    private:
        int count;
    public:
        Counter()
        {
            count = 0;
        }
        void increment()
        {
            count++;
        }
        void decrement()
        {
            count--;
        }
        void showData()
        {
            cout<<count;
        }
};

int main()
{
    Counter c1;
    cout<<"Initial value : ";
    c1.showData();
    c1.increment();
    cout<<endl;
    cout<<"After 1st increment : ";
    c1.showData();
    cout<<endl;
    c1.increment();
    cout<<"After 2nd increment : ";
    c1.showData();
    cout<<endl;
    c1.decrement();
    cout<<"After 1st decrement : ";
    c1.showData();
    cout<<endl;
    c1.decrement();
    cout<<"After 2nd decrement : ";
    c1.showData();
    return 0;
}
```

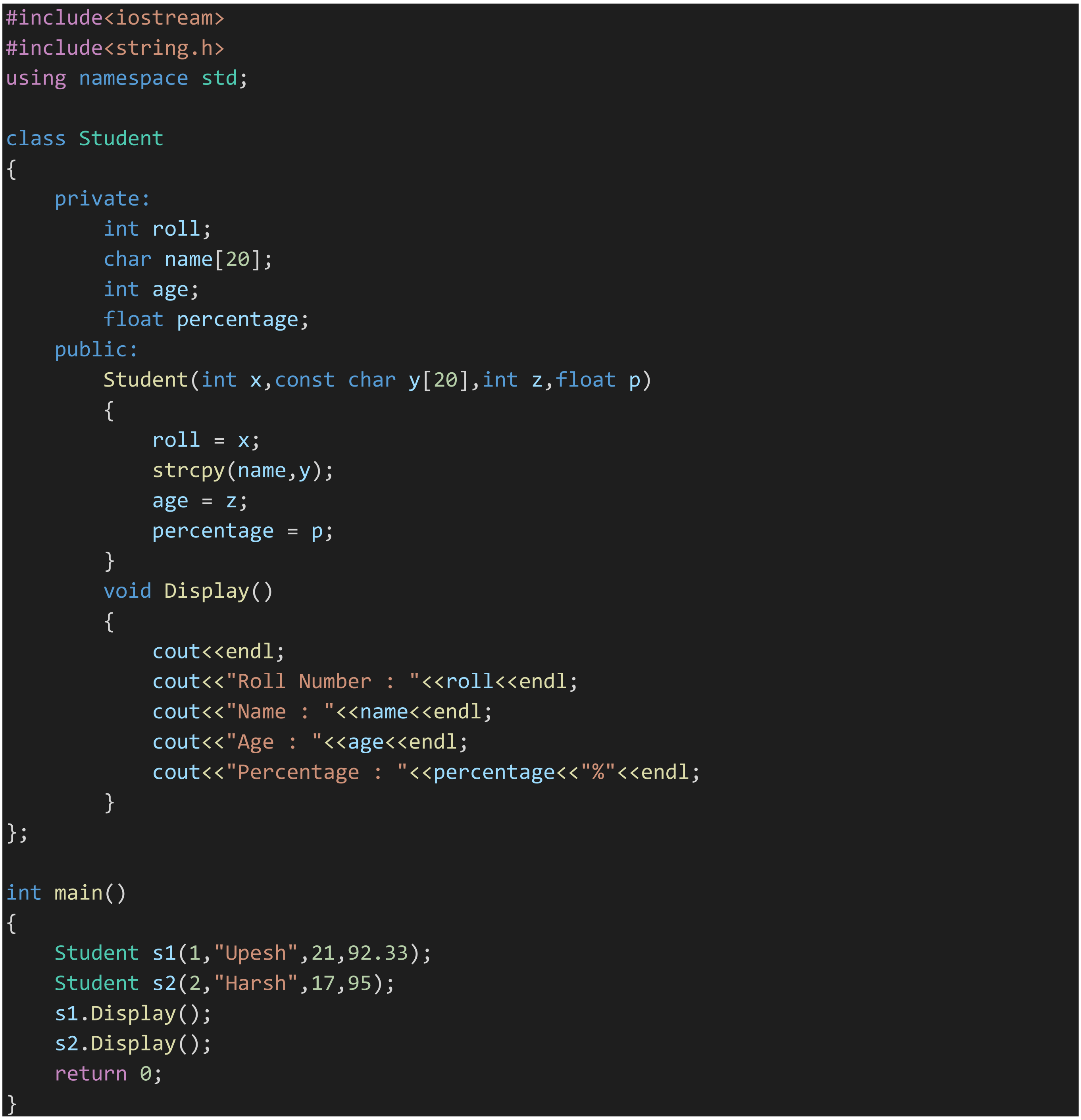
5. Define a class Date and write a program to Display Dateand initialise date object using Constructors.

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

class Date
{
    private:
        int dd;
        int mm;
        int yy;
    public:
        Date()
        {
            dd = 1;
            mm = 1;
            yy = 2001;
        }
        void Display()
        {
            cout<<dd<<"/"<<mm<<"/"<<yy<<endl;
        }
        void setDate(int x,int y,int z)
        {
            dd = x;
            mm = y;
            yy = z;
        }
};
```



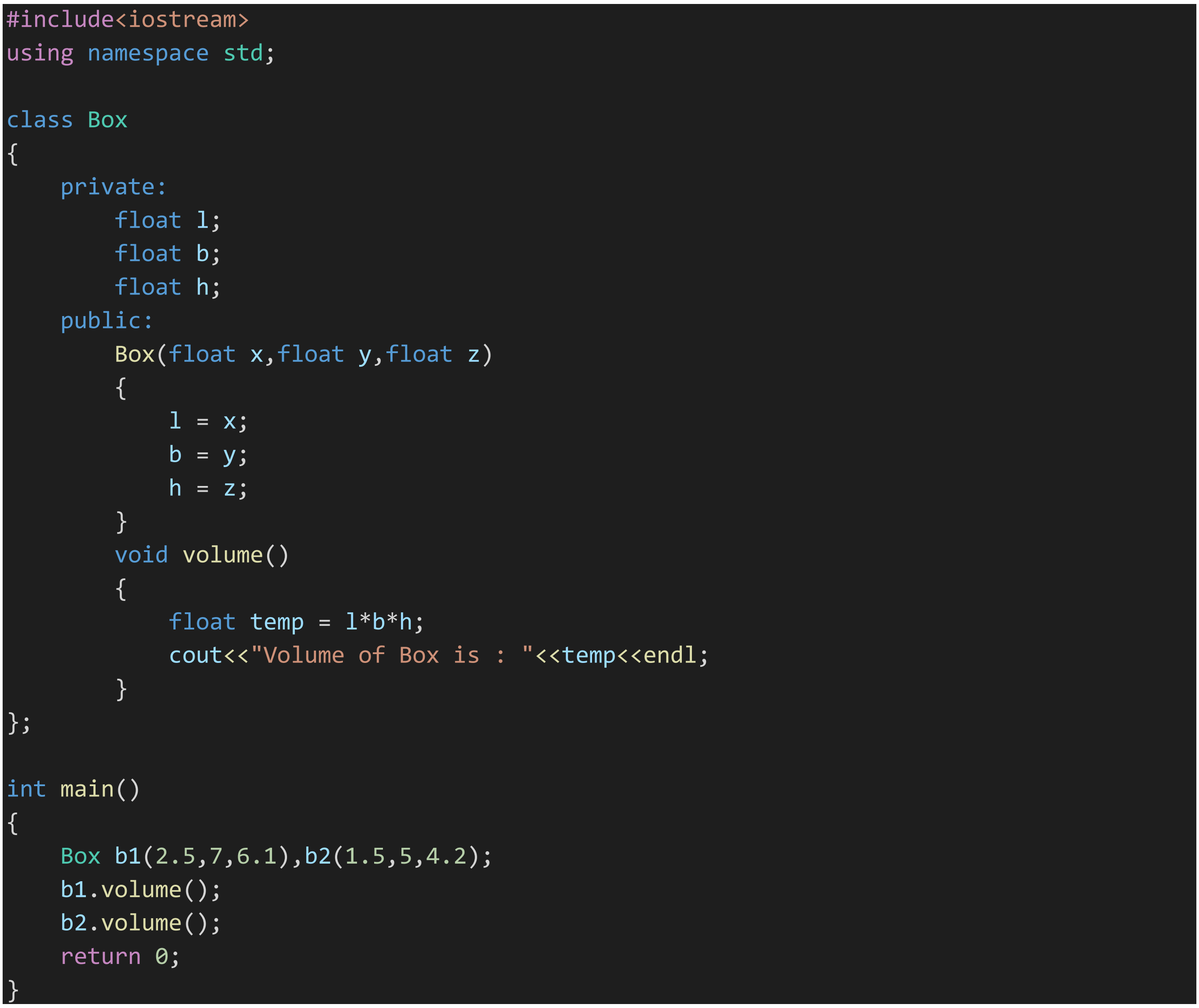
```
int main()
{
    Date d1,d2;
    cout<<"Before Setting of Date :"<<endl;
    d1.Display();
    d2.Display();
    d1.setDate(14,9,2022);
    d2.setDate(15,10,2022);
    cout<<"After setting of Date :"<<endl;
    d1.Display();
    d2.Display();
    return 0;
}
```

6. Define a class student and write a program to enter student details using constructor and define member function to display all the details.



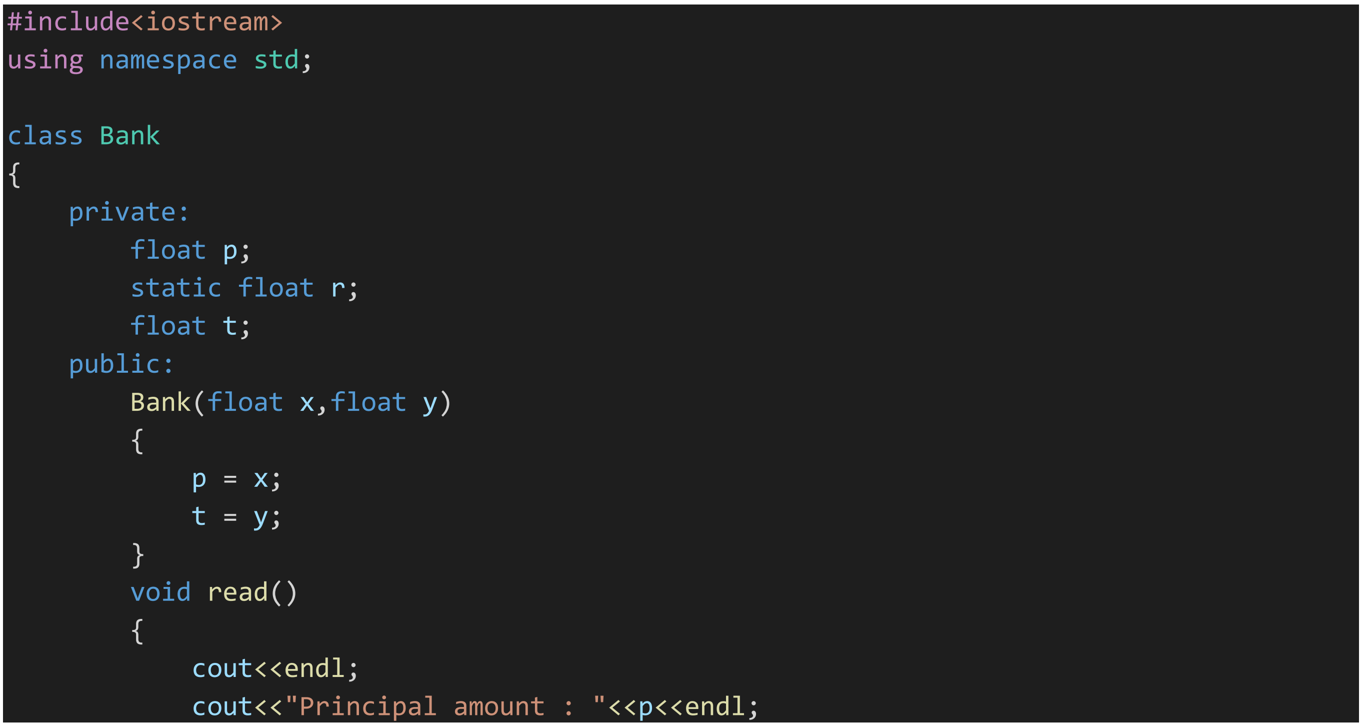
```
#include<iostream>
#include<string.h>
using namespace std;

class Student
{
    private:
        int roll;
        char name[20];
        int age;
        float percentage;
    public:
        Student(int x,const char y[20],int z,float p)
        {
            roll = x;
            strcpy(name,y);
            age = z;
            percentage = p;
        }
        void Display()
        {
            cout<<endl;
            cout<<"Roll Number : "<<roll<<endl;
            cout<<"Name : "<<name<<endl;
            cout<<"Age : "<<age<<endl;
            cout<<"Percentage : "<<percentage<<"%"<<endl;
        }
};

int main()
{
    Student s1(1,"Upesh",21,92.33);
    Student s2(2,"Harsh",17,95);
    s1.Display();
    s2.Display();
    return 0;
}
```

7. Define a class Box and write a program to enter length, breadth and height and initialise objects using constructor also define member functions to calculate volume of the box.

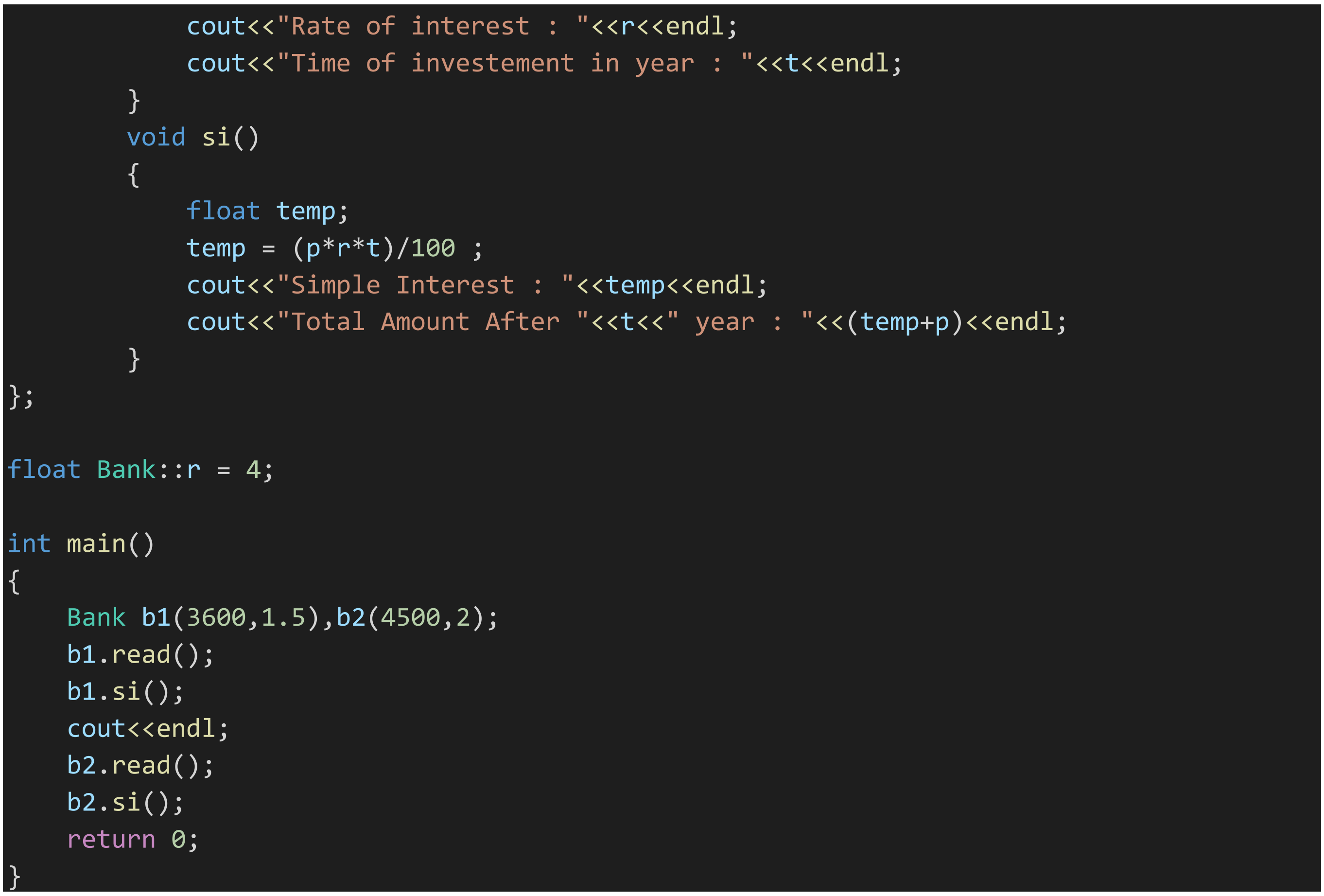


```
#include<iostream>
using namespace std;

class Box
{
    private:
        float l;
        float b;
        float h;
    public:
        Box(float x,float y,float z)
        {
            l = x;
            b = y;
            h = z;
        }
        void volume()
        {
            float temp = l*b*h;
            cout<<"Volume of Box is : "<<temp<<endl;
        }
};

int main()
{
    Box b1(2.5,7,6.1),b2(1.5,5,4.2);
    b1.volume();
    b2.volume();
    return 0;
}
```

8. Define a class Bank and define member functions to read principal , rate of interest and year. Another member functions to calculate simple interest and display it. Initialise all details using constructor.

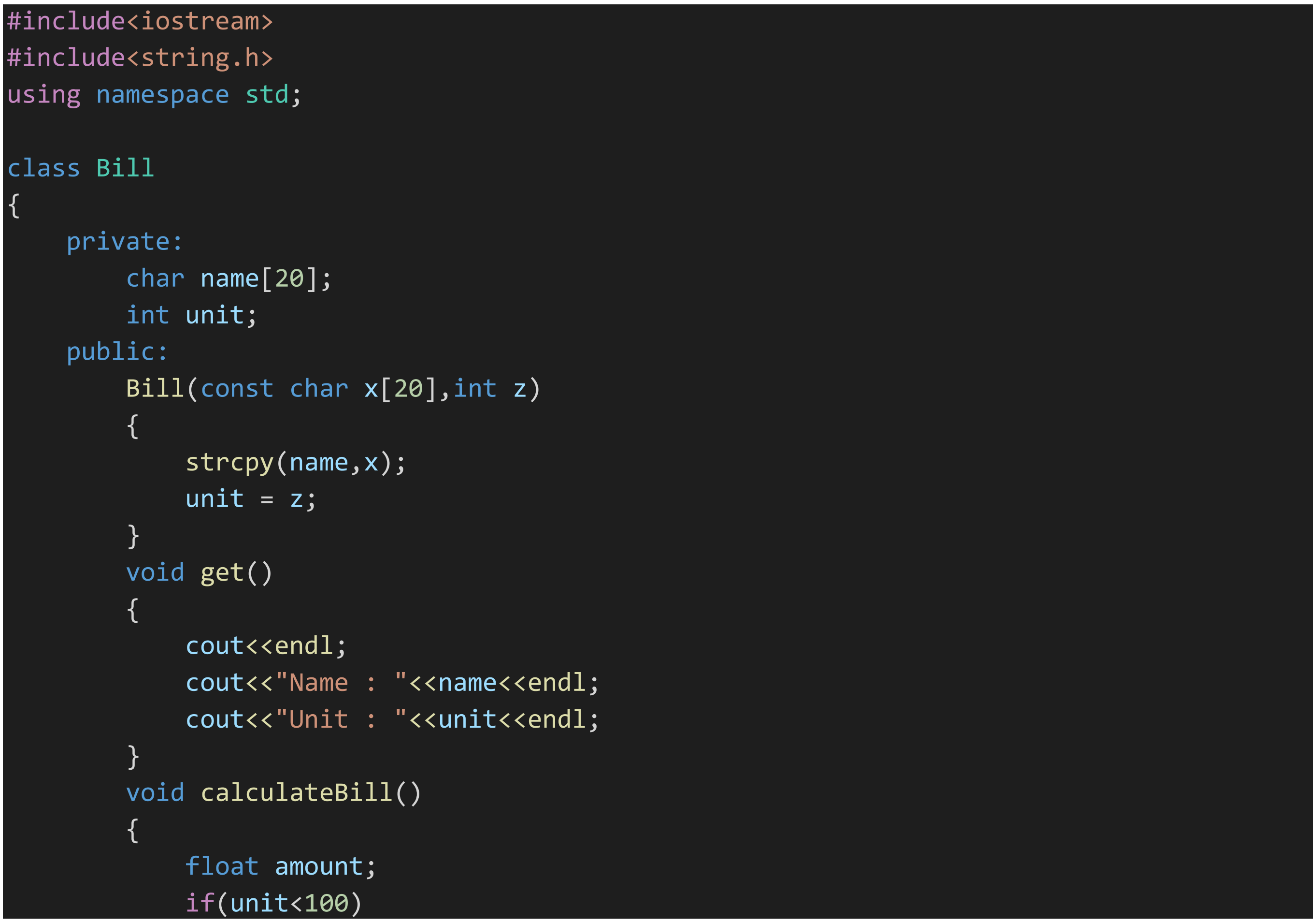


```
#include<iostream>
using namespace std;

class Bank
{
    private:
        float p;
        static float r;
        float t;
    public:
        Bank(float x,float y)
        {
            p = x;
            t = y;
        }
        void read()
        {
            cout<<endl;
            cout<<"Principal amount : "<<p<<endl;
```

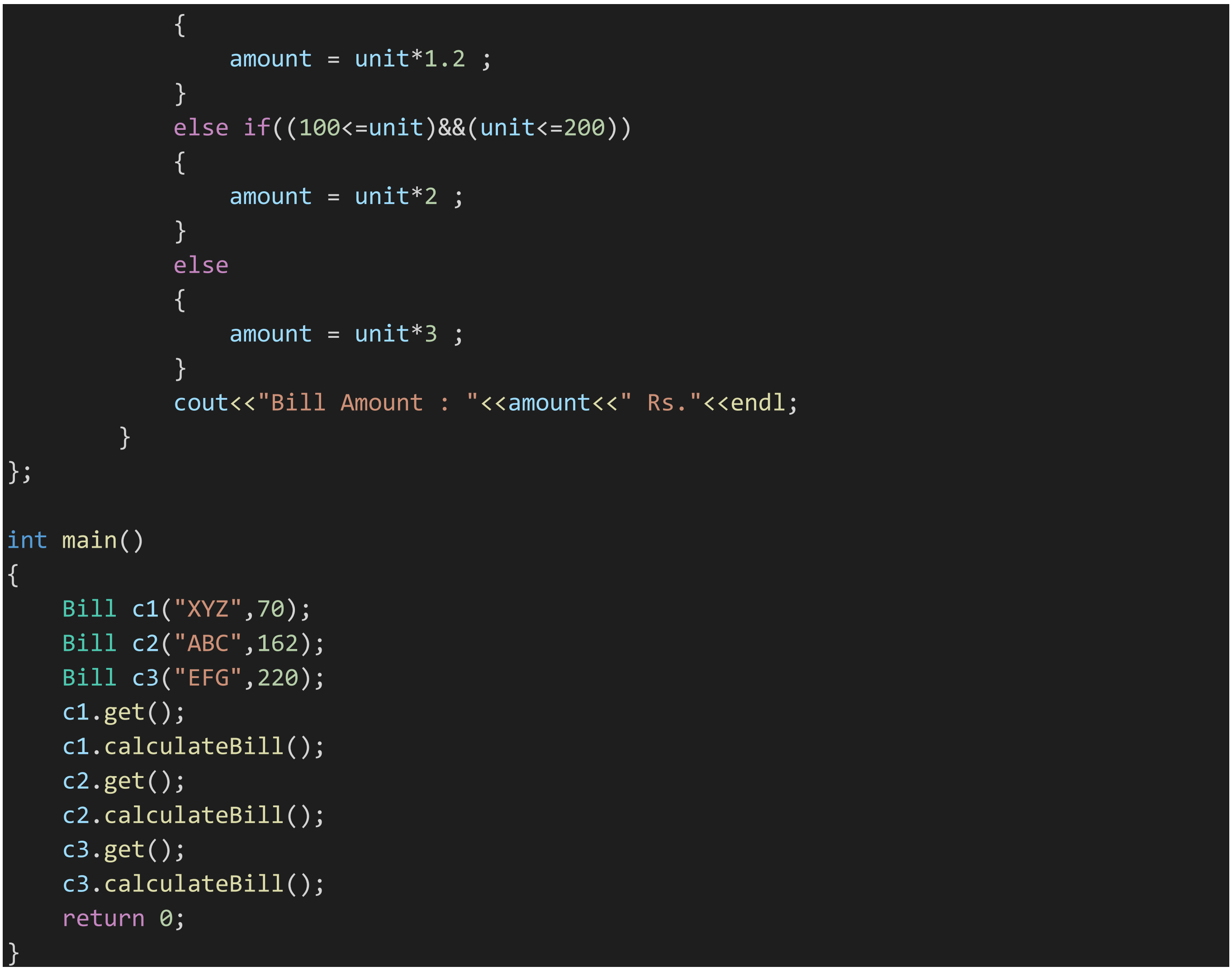


```cpp
            cout<<"Rate of interest : "<<r<<endl;
            cout<<"Time of investement in year : "<<t<<endl;
        }
        void si()
        {
            float temp;
            temp = (p*r*t)/100 ;
            cout<<"Simple Interest : "<<temp<<endl;
            cout<<"Total Amount After "<<t<<" year : "<<(temp+p)<<endl;
        }
};

float Bank::r = 4;

int main()
{
    Bank b1(3600,1.5),b2(4500,2);
    b1.read();
    b1.si();
    cout<<endl;
    b2.read();
    b2.si();
    return 0;
}
```

9. Define a class Bill and define its member function get() to take detail of customer , calculateBill() function to calculate electricity bill using below tariff :
   Upto 100 unit RS. 1.20 per unit
   From 100 to 200 unit RS. 2 per unit
   Above 200 units RS. 3 per unit.

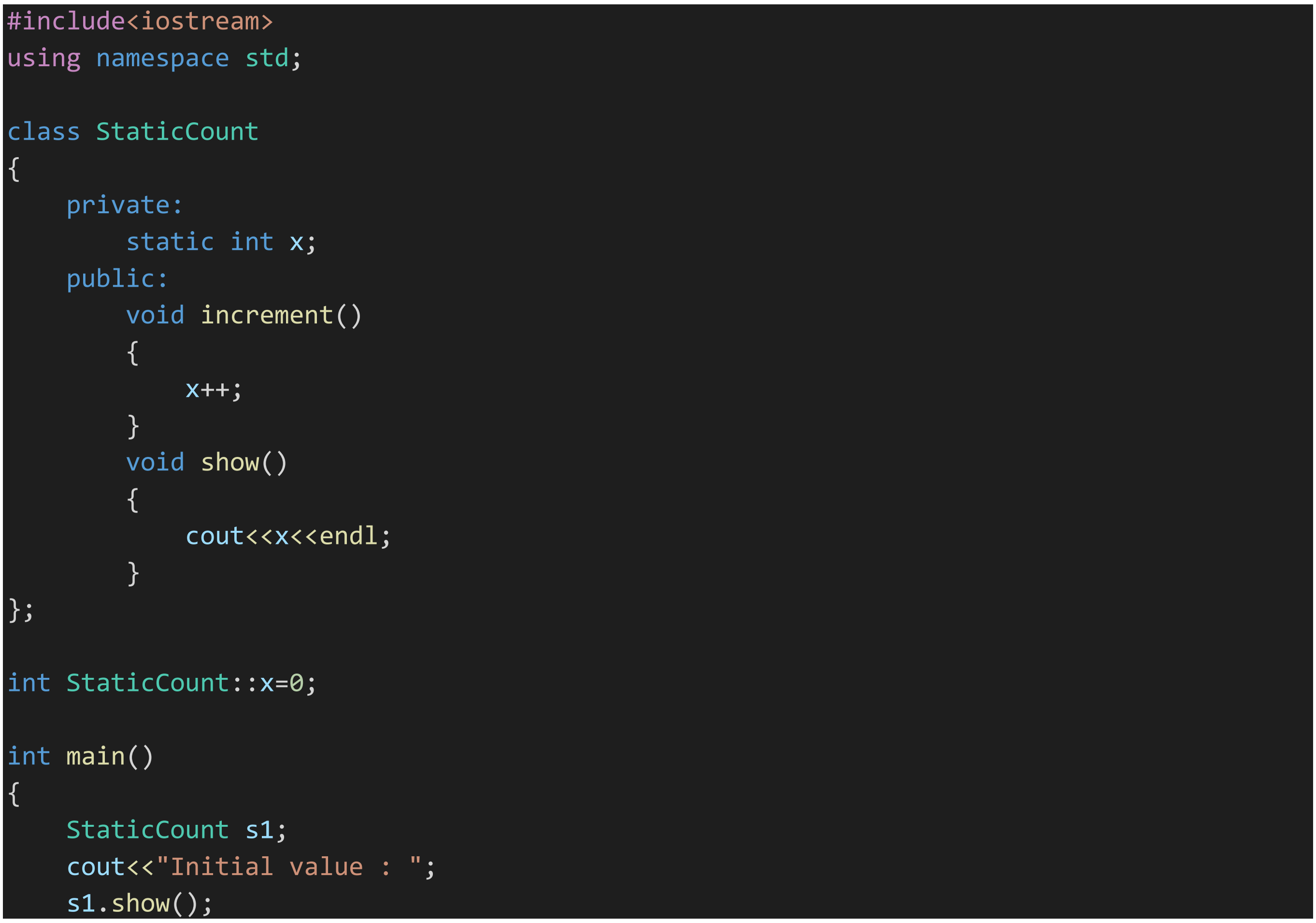


```cpp
#include<iostream>
#include<string.h>
using namespace std;

class Bill
{
    private:
        char name[20];
        int unit;
    public:
        Bill(const char x[20],int z)
        {
            strcpy(name,x);
            unit = z;
        }
        void get()
        {
            cout<<endl;
            cout<<"Name : "<<name<<endl;
            cout<<"Unit : "<<unit<<endl;
        }
        void calculateBill()
        {
            float amount;
            if(unit<100)
```



```
            {
                amount = unit*1.2 ;
            }
            else if((100<=unit)&&(unit<=200))
            {
                amount = unit*2 ;
            }
            else
            {
                amount = unit*3 ;
            }
            cout<<"Bill Amount : "<<amount<<" Rs."<<endl;
        }
};

int main()
{
    Bill c1("XYZ",70);
    Bill c2("ABC",162);
    Bill c3("EFG",220);
    c1.get();
    c1.calculateBill();
    c2.get();
    c2.calculateBill();
    c3.get();
    c3.calculateBill();
    return 0;
}
```

10.Define a class StaticCount and create a static variable. Increment this variable in a function and call this 3 times and display the result.



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

class StaticCount
{
    private:
        static int x;
    public:
        void increment()
        {
            x++;
        }
        void show()
        {
            cout<<x<<endl;
        }
};

int StaticCount::x=0;

int main()
{
    StaticCount s1;
    cout<<"Initial value : ";
    s1.show();
    s1.increment();
    cout<<"Value After 1st increment : ";
    s1.show();
    s1.increment();
    cout<<"Value After 2nd increment : ";
    s1.show();
    s1.increment();
    cout<<"Value After 3rd increment : ";
    s1.show();
    return 0;
}
```