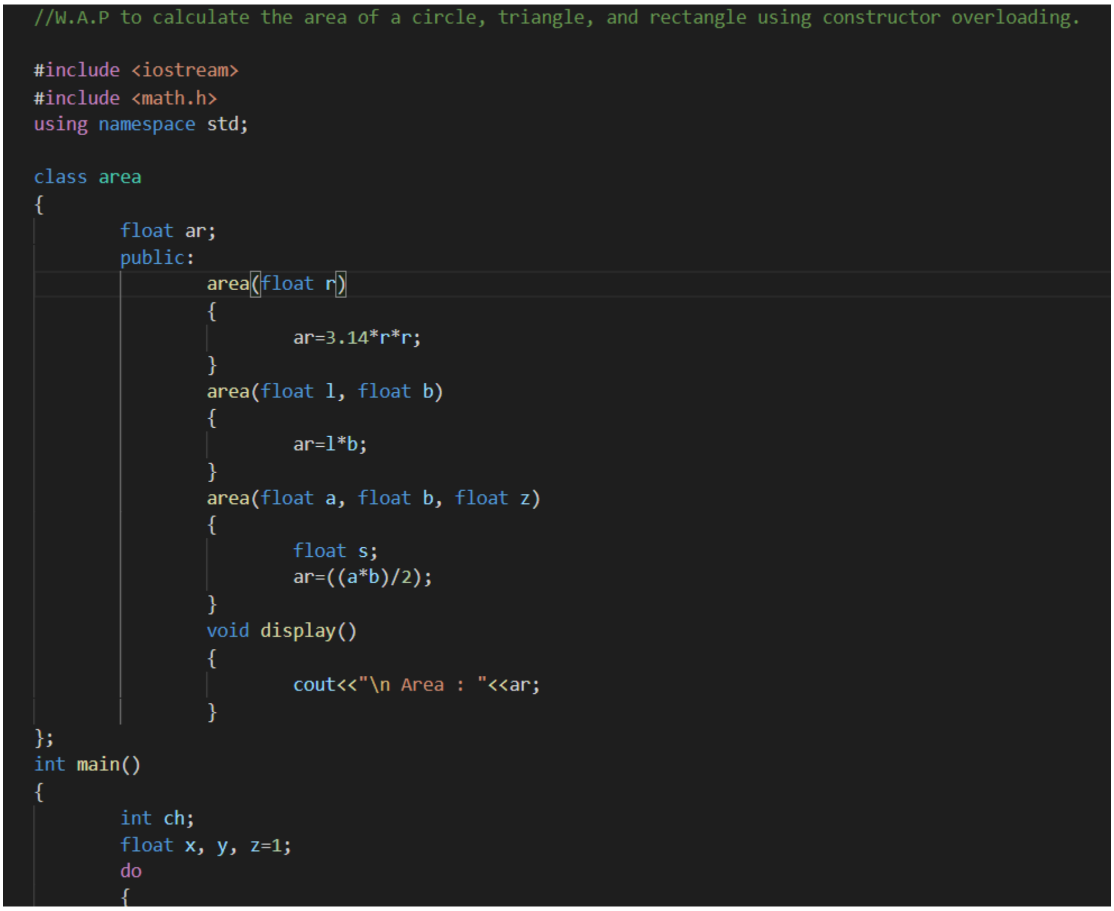
Agnim Gupta

2028083

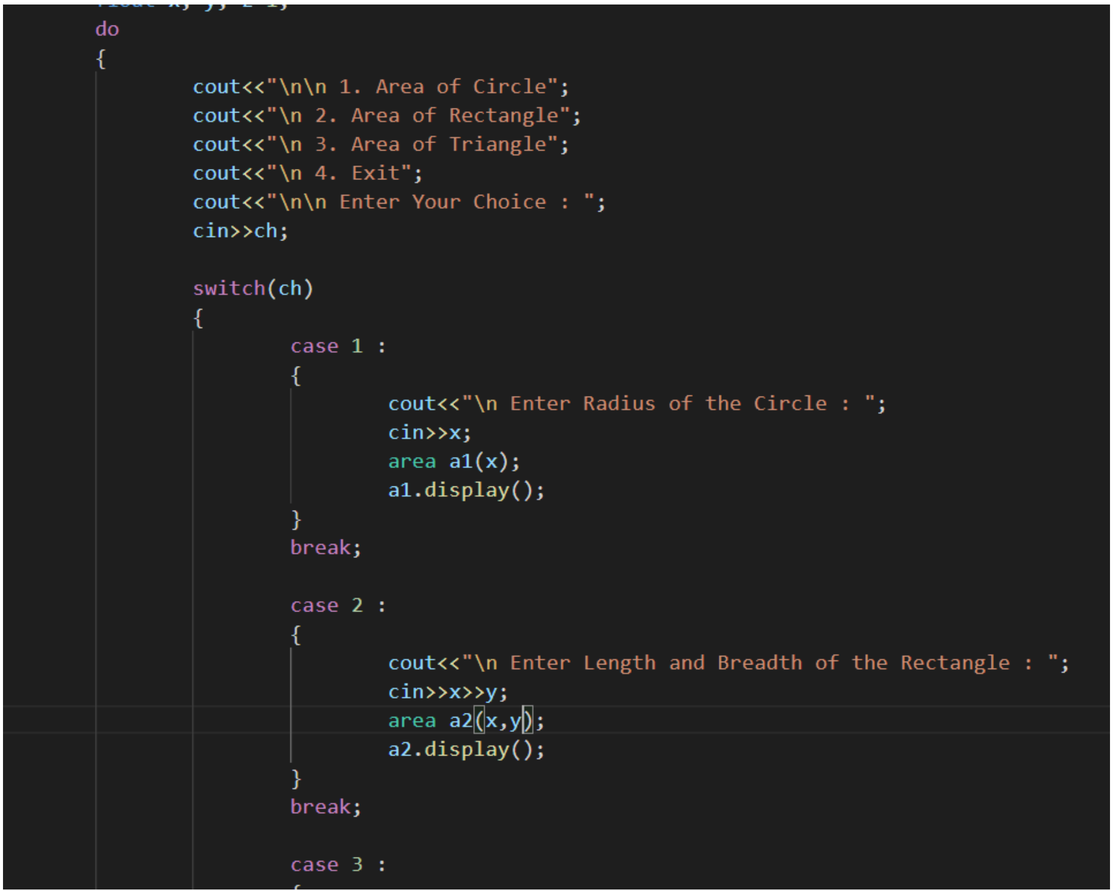
A23

CSSE

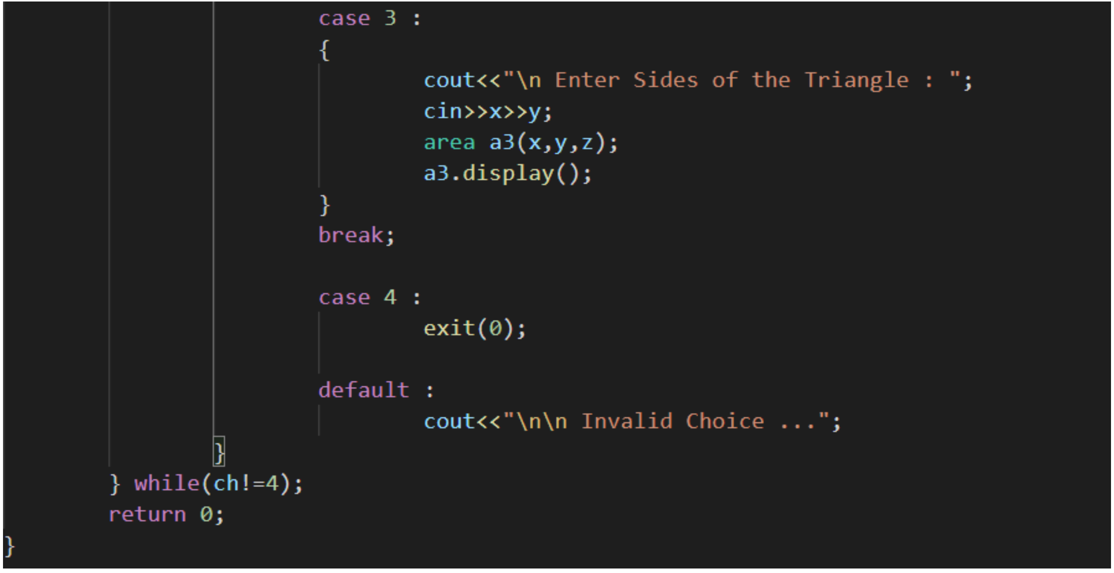
Question 1



```
//W.A.P to calculate the area of a circle, triangle, and rectangle using constructor overloading.

#include <iostream>
#include <math.h>
using namespace std;

class area
{
        float ar;
        public:
                area(float r)
                {
                        ar=3.14*r*r;
                }
                area(float l, float b)
                {
                        ar=l*b;
                }
                area(float a, float b, float z)
                {
                        float s;
                        ar=((a*b)/2);
                }
                void display()
                {
                        cout<<"\n Area : "<<ar;
                }
};
int main()
{
        int ch;
        float x, y, z=1;
        do
        {
```

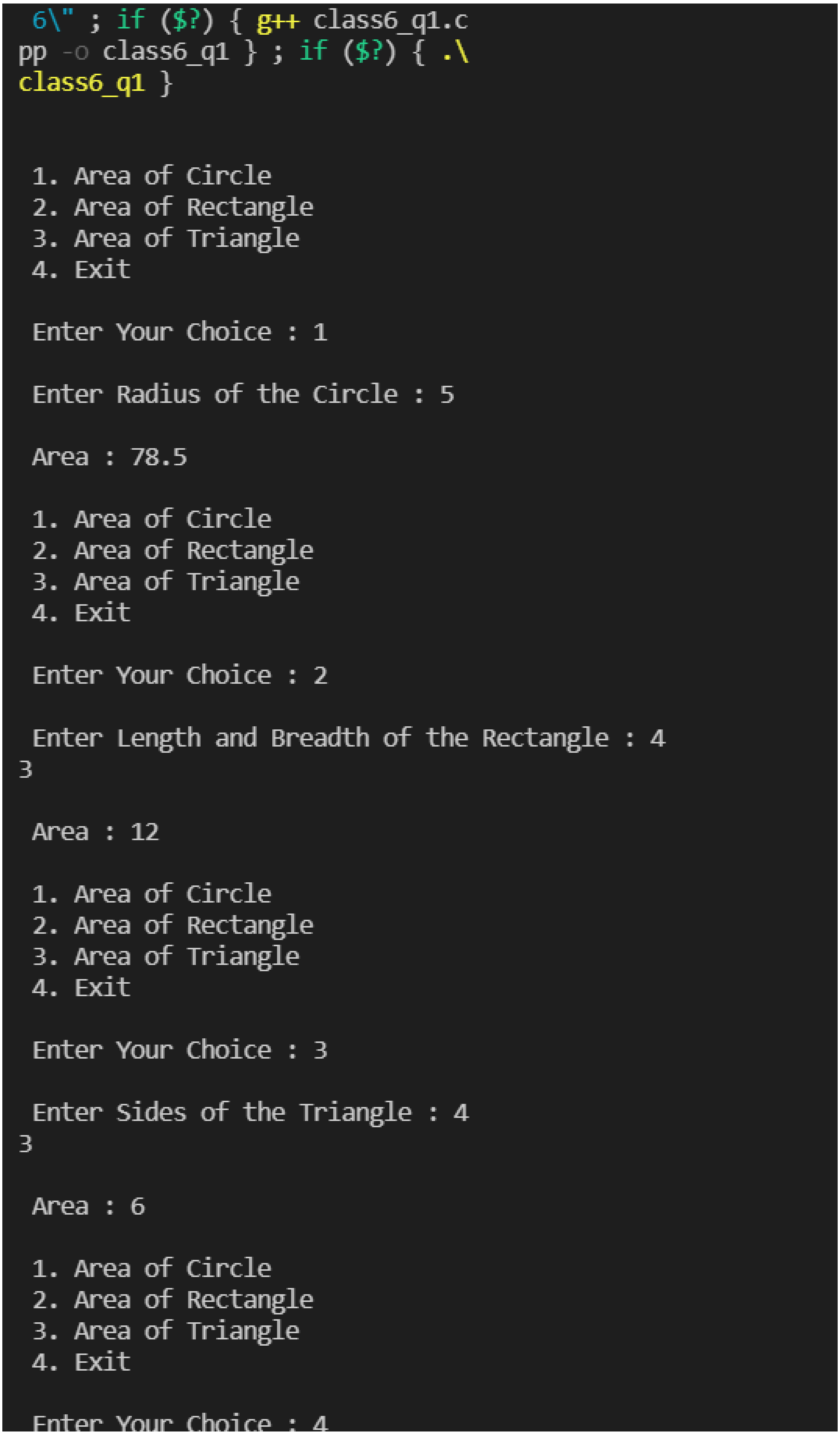


```
    float x, y, z=1;
    do
    {
        cout<<"\n\n 1. Area of Circle";
        cout<<"\n 2. Area of Rectangle";
        cout<<"\n 3. Area of Triangle";
        cout<<"\n 4. Exit";
        cout<<"\n\n Enter Your Choice : ";
        cin>>ch;

        switch(ch)
        {
            case 1 :
            {
                cout<<"\n Enter Radius of the Circle : ";
                cin>>x;
                area a1(x);
                a1.display();
            }
            break;

            case 2 :
            {
                cout<<"\n Enter Length and Breadth of the Rectangle : ";
                cin>>x>>y;
                area a2(x,y);
                a2.display();
            }
            break;

            case 3 :
```



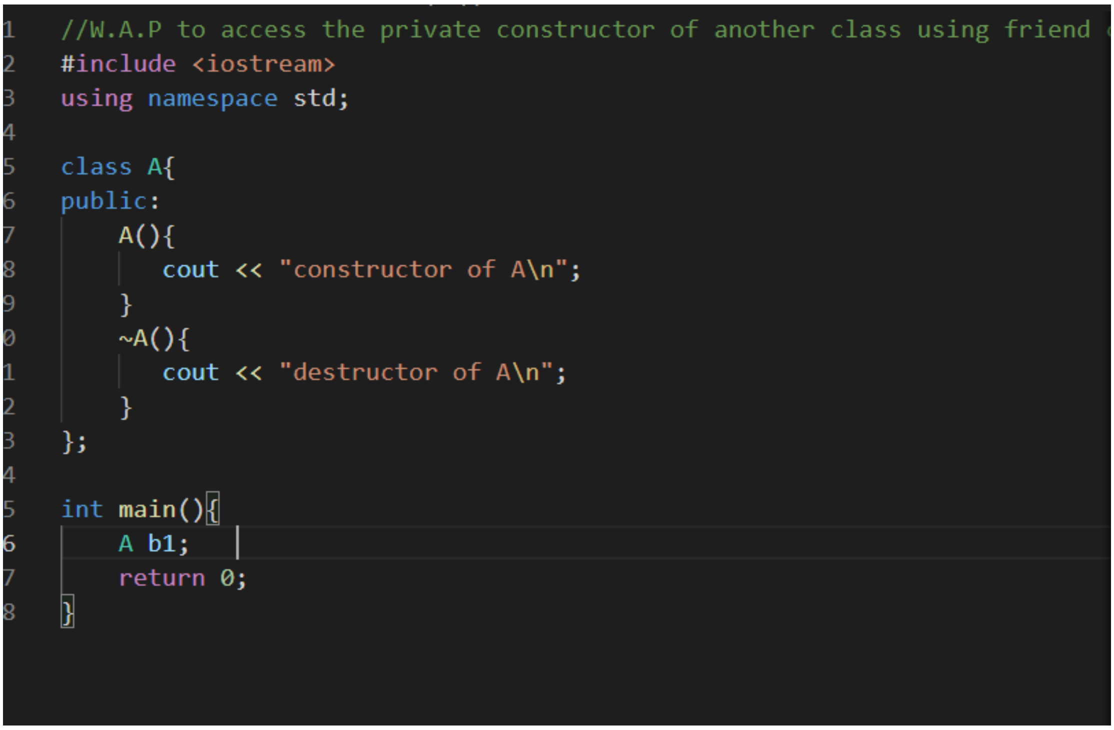
```
            case 3 :
            {
                cout<<"\n Enter Sides of the Triangle : ";
                cin>>x>>y;
                area a3(x,y,z);
                a3.display();
            }
            break;

            case 4 :
                exit(0);

            default :
                cout<<"\n\n Invalid Choice ...";
        }
    } while(ch!=4);
    return 0;
}
```

Output



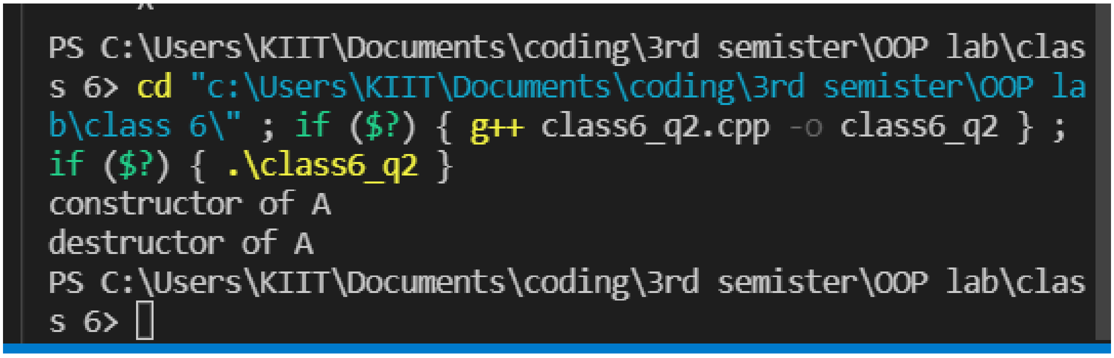
```
 6\" ; if ($?) { g++ class6_q1.c
pp -o class6_q1 } ; if ($?) { .\
class6_q1 }


 1. Area of Circle
 2. Area of Rectangle
 3. Area of Triangle
 4. Exit

 Enter Your Choice : 1

 Enter Radius of the Circle : 5

 Area : 78.5

 1. Area of Circle
 2. Area of Rectangle
 3. Area of Triangle
 4. Exit

 Enter Your Choice : 2

 Enter Length and Breadth of the Rectangle : 4
3

 Area : 12

 1. Area of Circle
 2. Area of Rectangle
 3. Area of Triangle
 4. Exit

 Enter Your Choice : 3

 Enter Sides of the Triangle : 4
3

 Area : 6

 1. Area of Circle
 2. Area of Rectangle
 3. Area of Triangle
 4. Exit

 Enter Your Choice : 4
```

Question 2



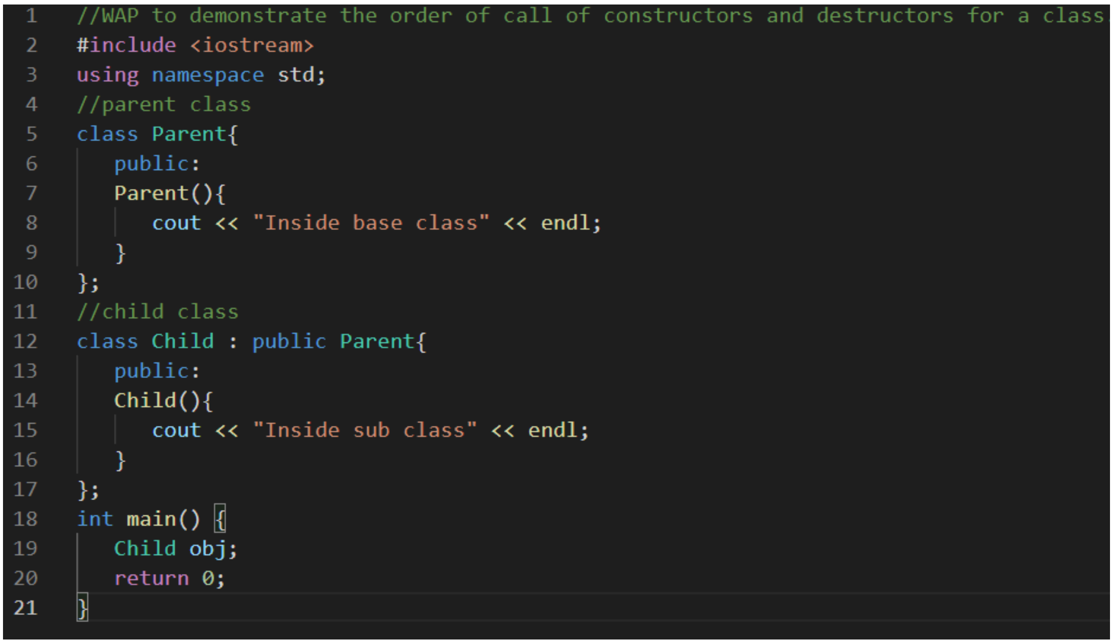
```cpp
//W.A.P to access the private constructor of another class using friend
#include <iostream>
using namespace std;

class A{
public:
    A(){
        cout << "constructor of A\n";
    }
    ~A(){
        cout << "destructor of A\n";
    }
};

int main(){
    A b1;
    return 0;
}
```

Output



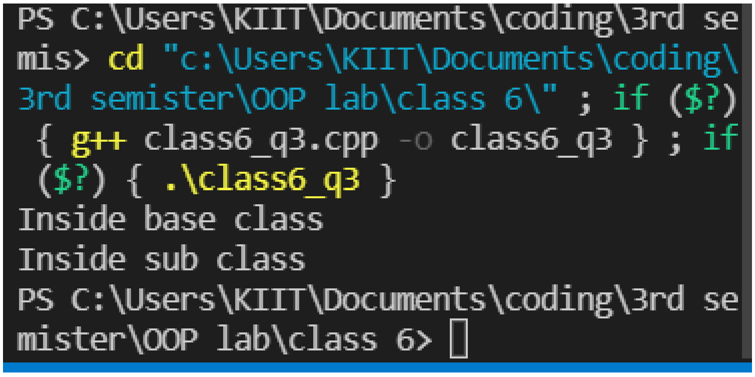
```
PS C:\Users\KIIT\Documents\coding\3rd semister\OOP lab\clas
s 6> cd "c:\Users\KIIT\Documents\coding\3rd semister\OOP la
b\class 6\" ; if ($?) { g++ class6_q2.cpp -o class6_q2 } ;
if ($?) { .\class6_q2 }
constructor of A
destructor of A
PS C:\Users\KIIT\Documents\coding\3rd semister\OOP lab\clas
s 6> 
```

Question 3



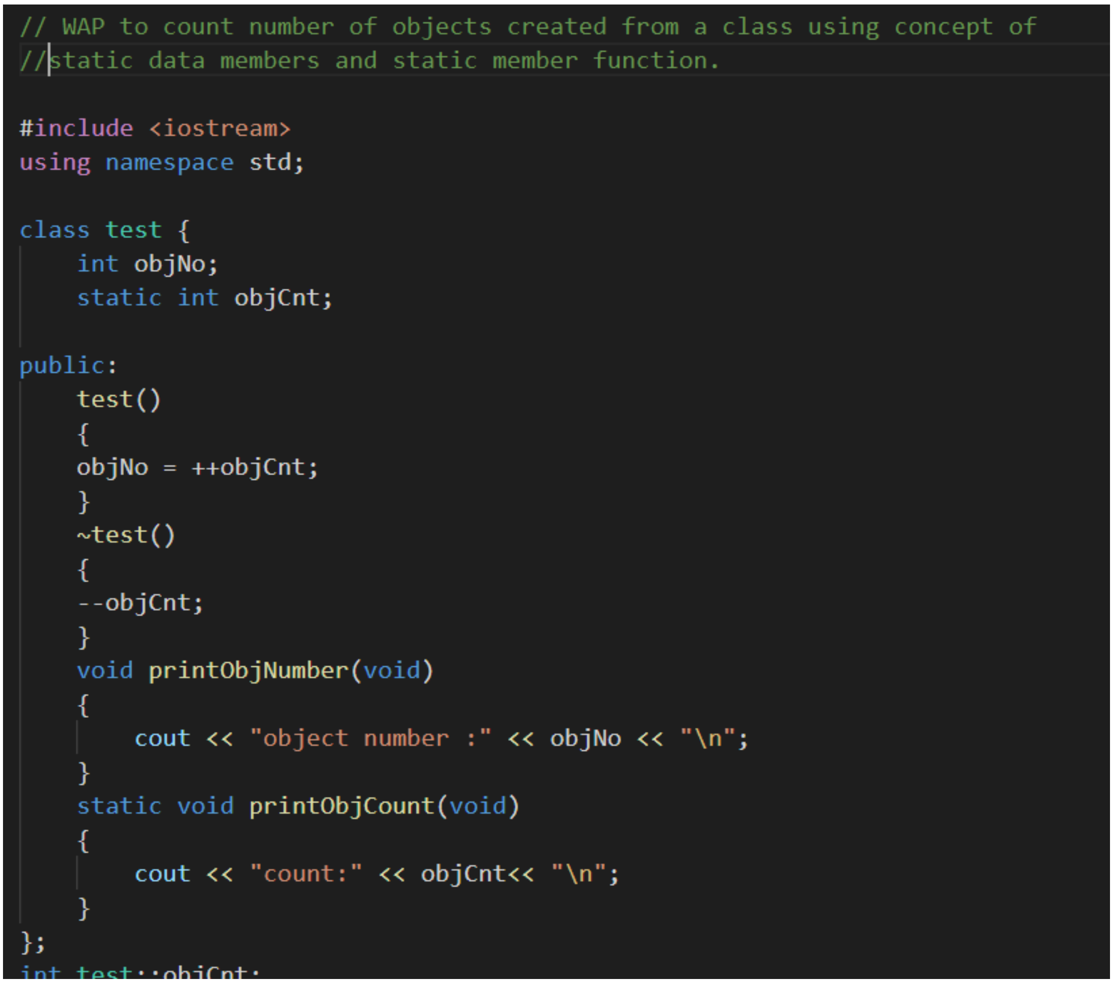
```cpp
//WAP to demonstrate the order of call of constructors and destructors for a class
#include <iostream>
using namespace std;
//parent class
class Parent{
    public:
    Parent(){
        cout << "Inside base class" << endl;
    }
};
//child class
class Child : public Parent{
    public:
    Child(){
        cout << "Inside sub class" << endl;
    }
};
int main() {
    Child obj;
    return 0;
}
```

Output

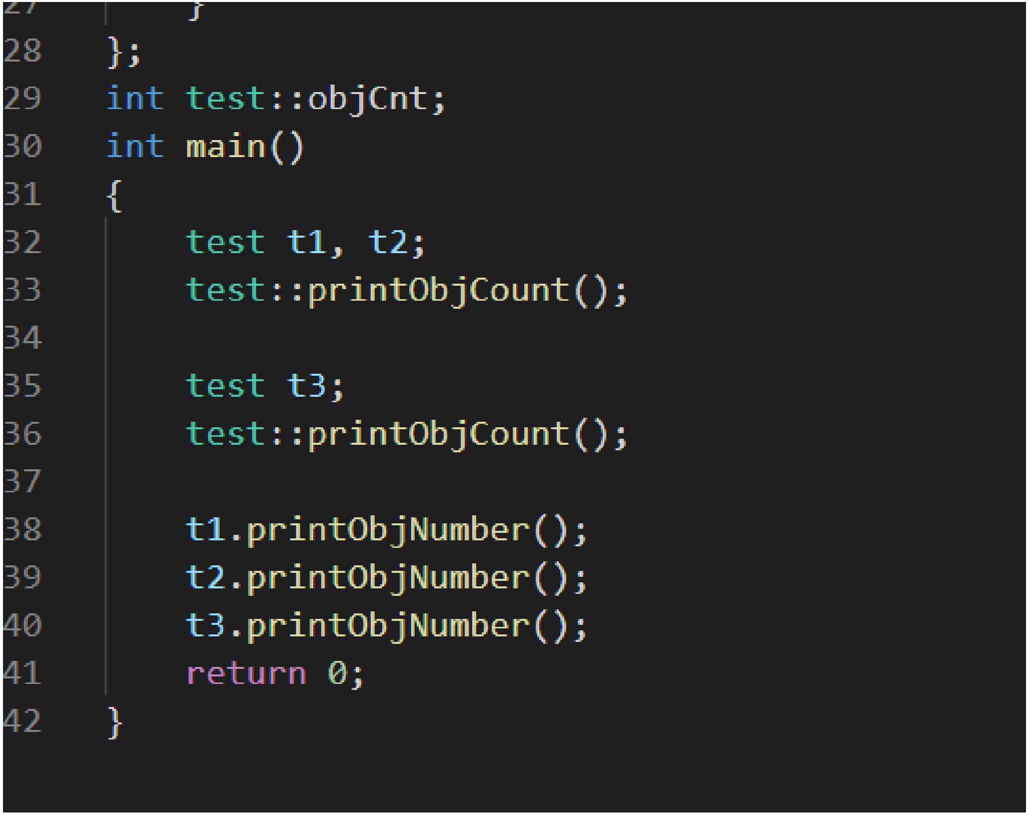


```
PS C:\Users\KIIT\Documents\coding\3rd se
mis> cd "c:\Users\KIIT\Documents\coding\
3rd semister\OOP lab\class 6\" ; if ($?)
 { g++ class6_q3.cpp -o class6_q3 } ; if
 ($?) { .\class6_q3 }
Inside base class
Inside sub class
PS C:\Users\KIIT\Documents\coding\3rd se
mister\OOP lab\class 6>
```

Question 4

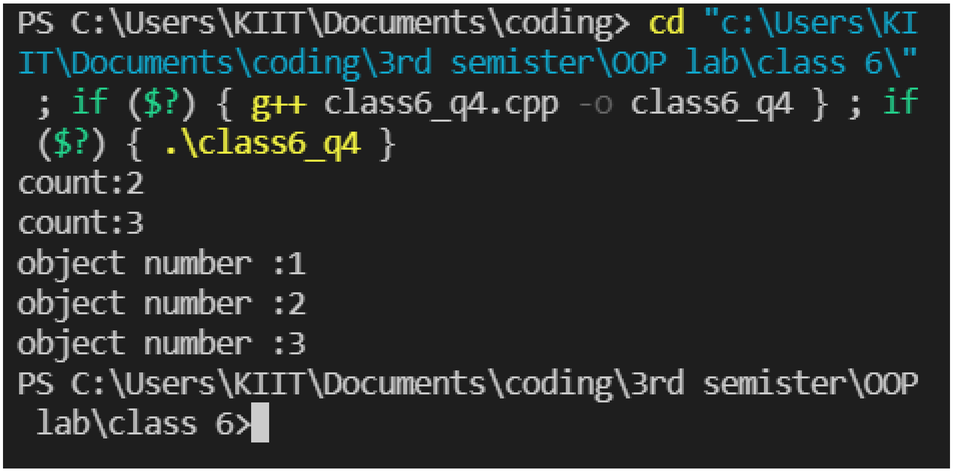


```
// WAP to count number of objects created from a class using concept of
//static data members and static member function.

#include <iostream>
using namespace std;

class test {
    int objNo;
    static int objCnt;

public:
    test()
    {
    objNo = ++objCnt;
    }
    ~test()
    {
    --objCnt;
    }
    void printObjNumber(void)
    {
        cout << "object number :" << objNo << "\n";
    }
    static void printObjCount(void)
    {
        cout << "count:" << objCnt<< "\n";
    }
};
int test::objCnt;
```



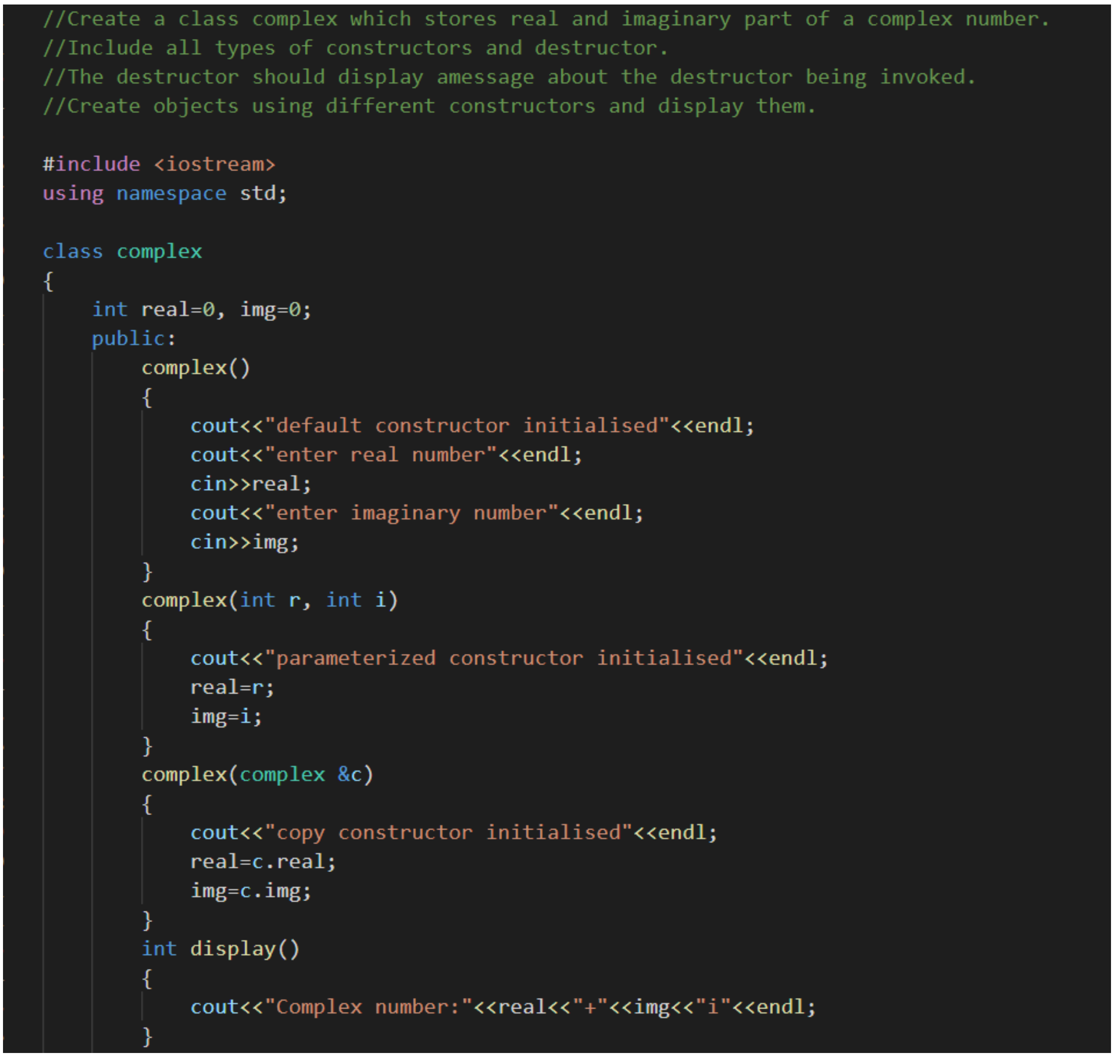
```
    }
};
int test::objCnt;
int main()
{
    test t1, t2;
    test::printObjCount();

    test t3;
    test::printObjCount();

    t1.printObjNumber();
    t2.printObjNumber();
    t3.printObjNumber();
    return 0;
}
```

Output

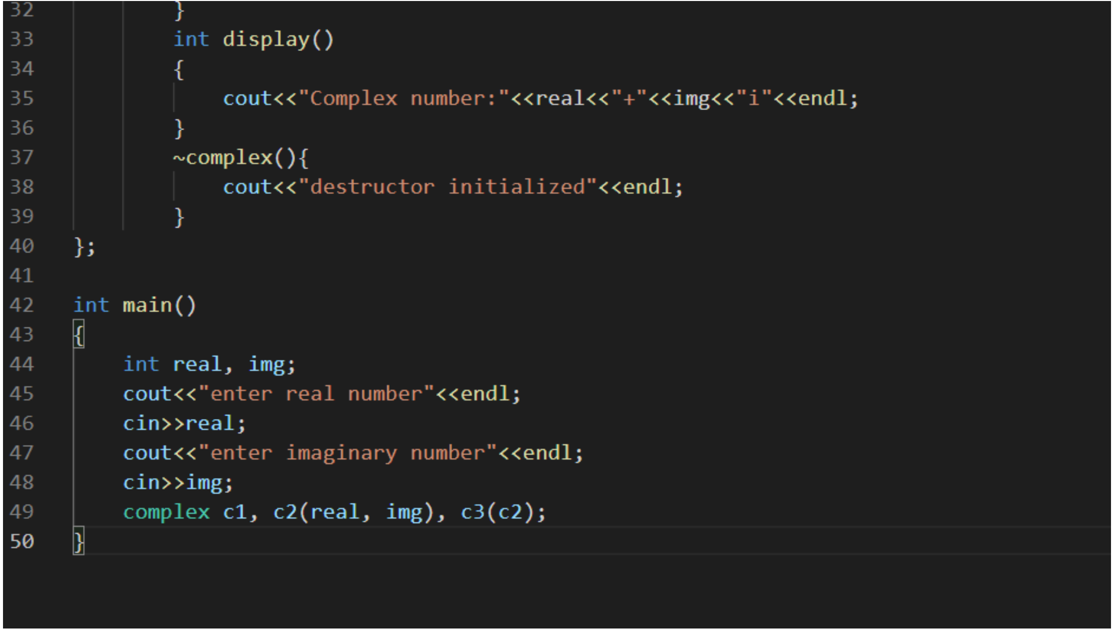


```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KI
IT\Documents\coding\3rd semister\OOP lab\class 6\"
 ; if ($?) { g++ class6_q4.cpp -o class6_q4 } ; if
 ($?) { .\class6_q4 }
count:2
count:3
object number :1
object number :2
object number :3
PS C:\Users\KIIT\Documents\coding\3rd semister\OOP
 lab\class 6>
```

Question 5

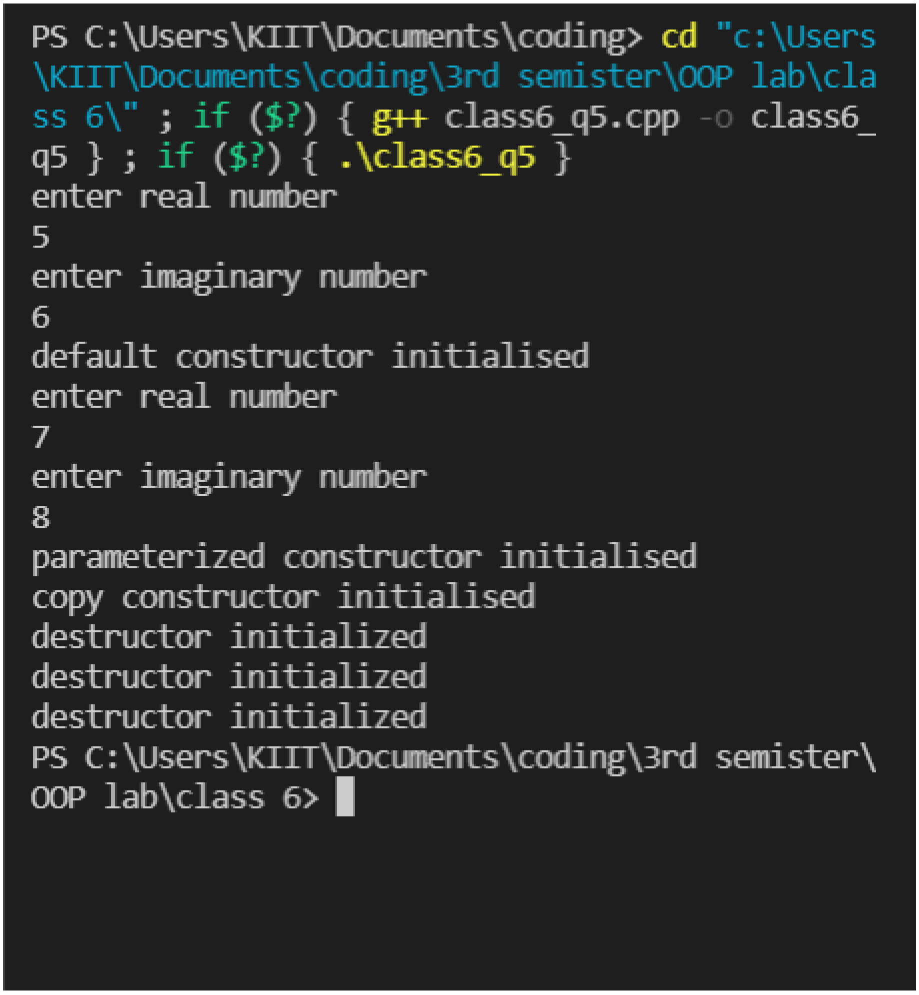


```
//Create a class complex which stores real and imaginary part of a complex number.
//Include all types of constructors and destructor.
//The destructor should display amessage about the destructor being invoked.
//Create objects using different constructors and display them.

#include <iostream>
using namespace std;

class complex
{
    int real=0, img=0;
    public:
        complex()
        {
            cout<<"default constructor initialised"<<endl;
            cout<<"enter real number"<<endl;
            cin>>real;
            cout<<"enter imaginary number"<<endl;
            cin>>img;
        }
        complex(int r, int i)
        {
            cout<<"parameterized constructor initialised"<<endl;
            real=r;
            img=i;
        }
        complex(complex &c)
        {
            cout<<"copy constructor initialised"<<endl;
            real=c.real;
            img=c.img;
        }
        int display()
        {
            cout<<"Complex number:"<<real<<"+"<<img<<"i"<<endl;
        }
```



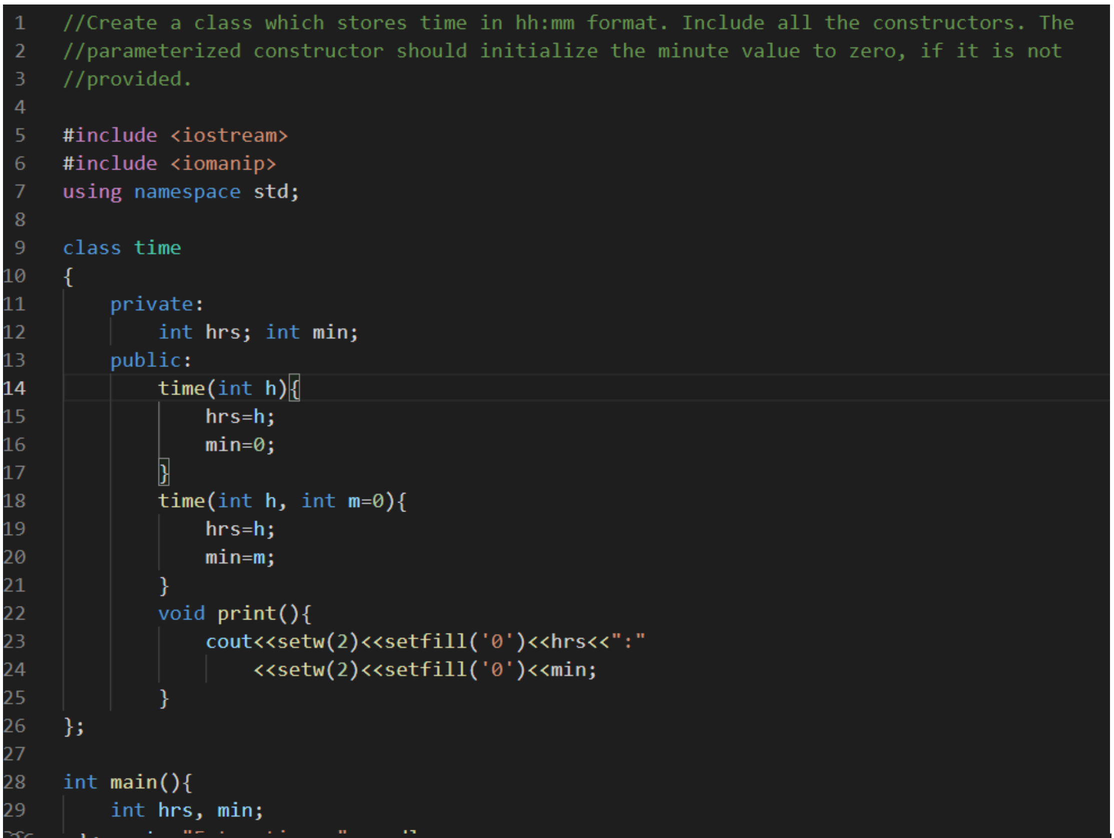
```
        }
        int display()
        {
            cout<<"Complex number:"<<real<<"+"<<img<<"i"<<endl;
        }
        ~complex(){
            cout<<"destructor initialized"<<endl;
        }
};

int main()
{
    int real, img;
    cout<<"enter real number"<<endl;
    cin>>real;
    cout<<"enter imaginary number"<<endl;
    cin>>img;
    complex c1, c2(real, img), c3(c2);
}
```

Output

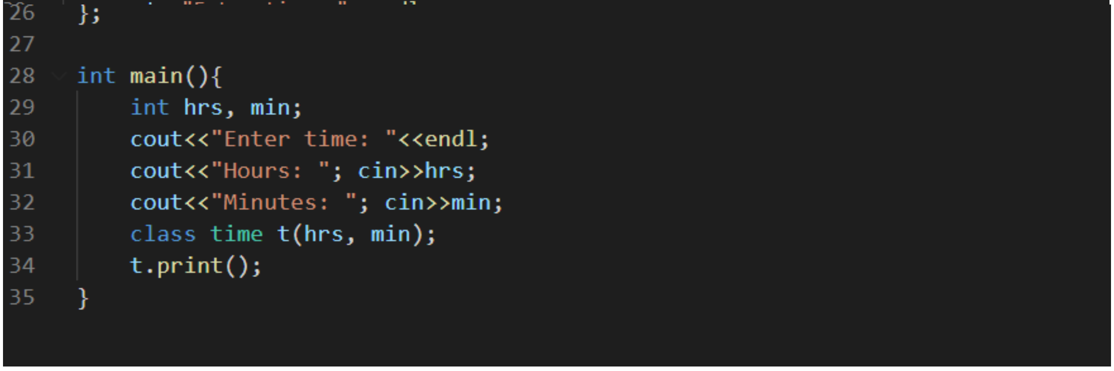


```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users
\KIIT\Documents\coding\3rd semister\OOP lab\cla
ss 6\" ; if ($?) { g++ class6_q5.cpp -o class6_
q5 } ; if ($?) { .\class6_q5 }
enter real number
5
enter imaginary number
6
default constructor initialised
enter real number
7
enter imaginary number
8
parameterized constructor initialised
copy constructor initialised
destructor initialized
destructor initialized
destructor initialized
PS C:\Users\KIIT\Documents\coding\3rd semister\
OOP lab\class 6>
```

Question 6

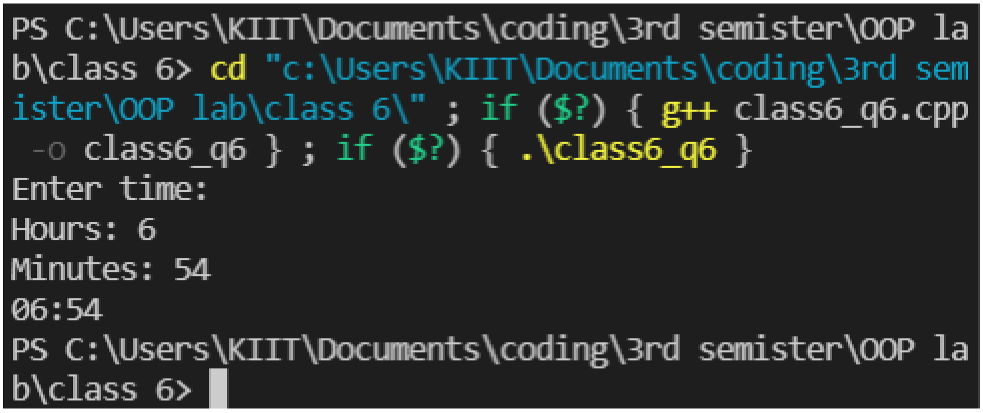


```
//Create a class which stores time in hh:mm format. Include all the constructors. The
//parameterized constructor should initialize the minute value to zero, if it is not
//provided.

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

class time
{
    private:
        int hrs; int min;
    public:
        time(int h){
            hrs=h;
            min=0;
        }
        time(int h, int m=0){
            hrs=h;
            min=m;
        }
        void print(){
            cout<<setw(2)<<setfill('0')<<hrs<<":"
                <<setw(2)<<setfill('0')<<min;
        }
};

int main(){
    int hrs, min;
```



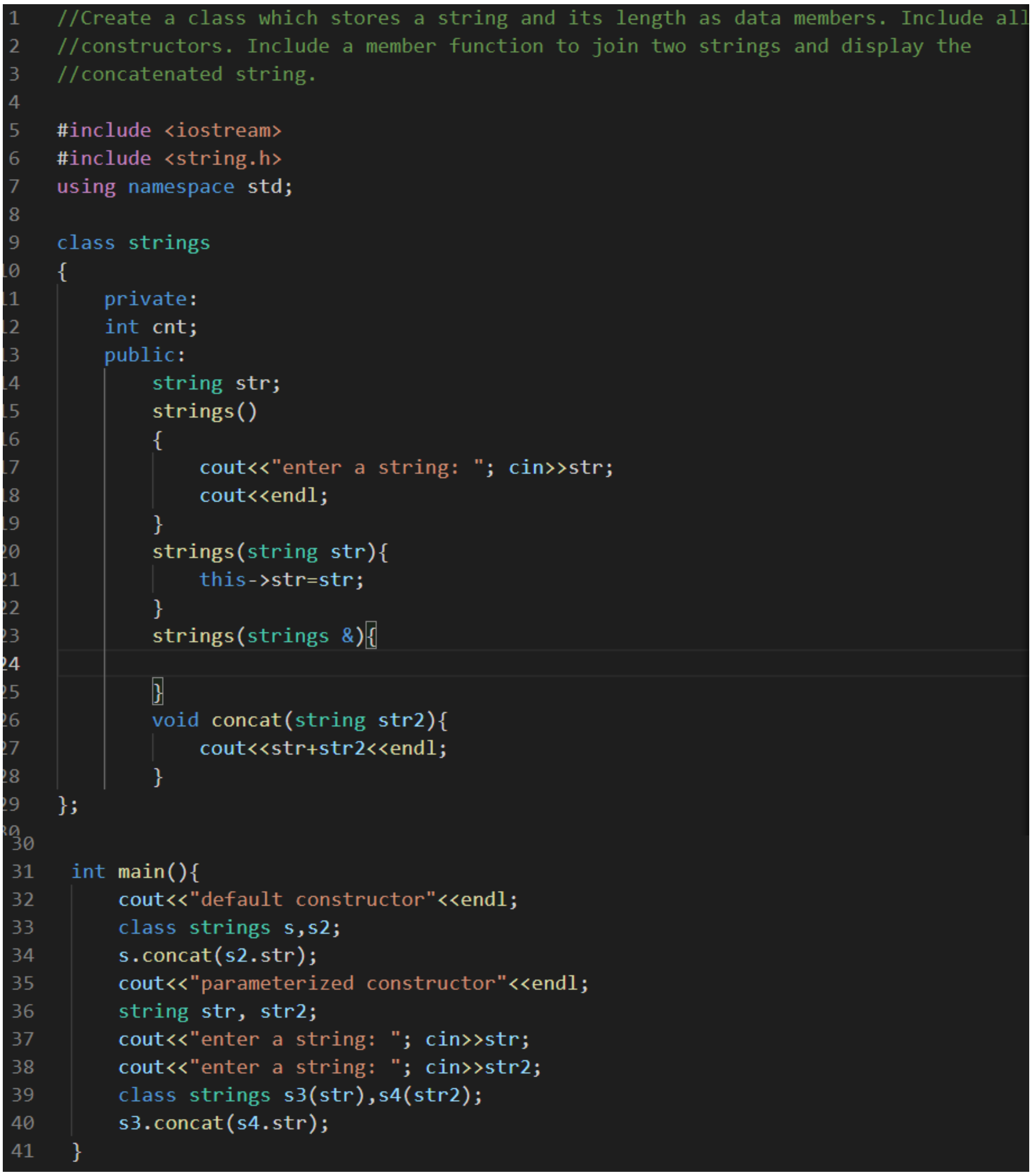
```
};

int main(){
    int hrs, min;
    cout<<"Enter time: "<<endl;
    cout<<"Hours: "; cin>>hrs;
    cout<<"Minutes: "; cin>>min;
    class time t(hrs, min);
    t.print();
}
```

Output



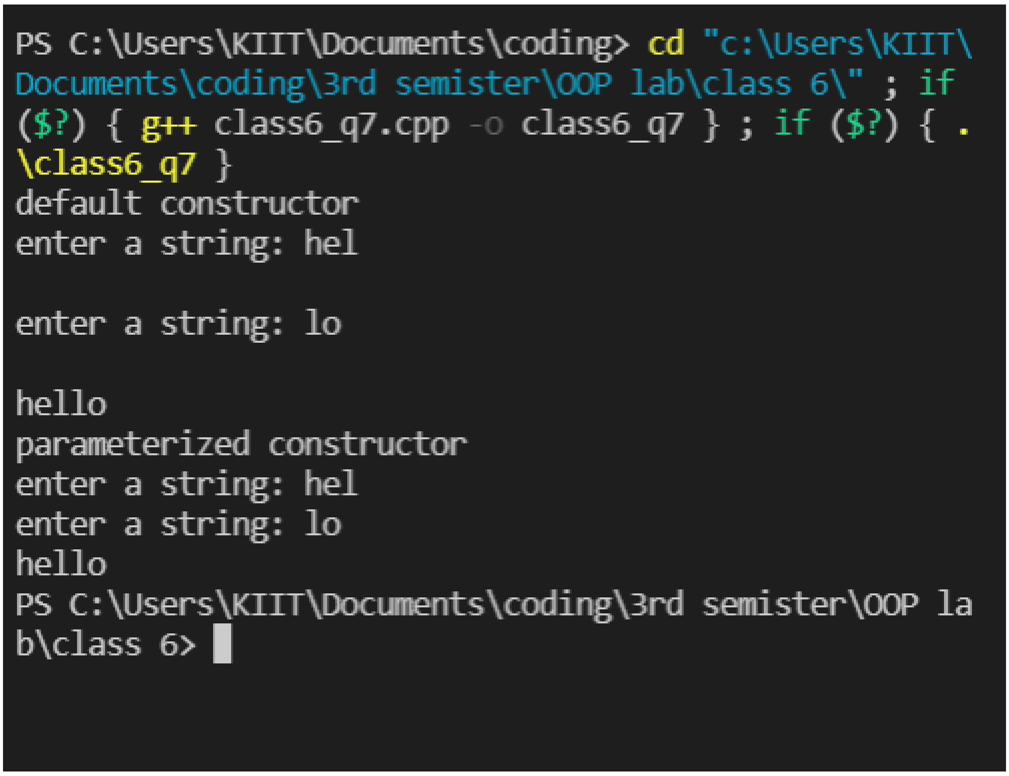
```
PS C:\Users\KIIT\Documents\coding\3rd semister\OOP la
b\class 6> cd "c:\Users\KIIT\Documents\coding\3rd sem
ister\OOP lab\class 6\" ; if ($?) { g++ class6_q6.cpp
 -o class6_q6 } ; if ($?) { .\class6_q6 }
Enter time:
Hours: 6
Minutes: 54
06:54
PS C:\Users\KIIT\Documents\coding\3rd semister\OOP la
b\class 6>
```

Question 7



```cpp
//Create a class which stores a string and its length as data members. Include all
//constructors. Include a member function to join two strings and display the
//concatenated string.

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

class strings
{
    private:
    int cnt;
    public:
        string str;
        strings()
        {
            cout<<"enter a string: "; cin>>str;
            cout<<endl;
        }
        strings(string str){
            this->str=str;
        }
        strings(strings &){

        }
        void concat(string str2){
            cout<<str+str2<<endl;
        }
};

int main(){
    cout<<"default constructor"<<endl;
    class strings s,s2;
    s.concat(s2.str);
    cout<<"parameterized constructor"<<endl;
    string str, str2;
    cout<<"enter a string: "; cin>>str;
    cout<<"enter a string: "; cin>>str2;
    class strings s3(str),s4(str2);
    s3.concat(s4.str);
}
```

Output



```
PS C:\Users\KIIT\Documents\coding> cd "c:\Users\KIIT\
Documents\coding\3rd semister\OOP lab\class 6\" ; if
($?) { g++ class6_q7.cpp -o class6_q7 } ; if ($?) { .
\class6_q7 }
default constructor
enter a string: hel

enter a string: lo

hello
parameterized constructor
enter a string: hel
enter a string: lo
hello
PS C:\Users\KIIT\Documents\coding\3rd semister\OOP la
b\class 6> 
```