ASSIGNMENT CONSTRUCTOR & FRIEND FUNCTION

1.Answer the questions (i) and (iii) after going through the following class:

```
class Seminar
{
  int time;
public:
  Seminar()       //Function 1  will be called by Seminar s1. (Default constructor )
  {
    time = 30;
    cout << "Seminar starts now" << endl;
  }
void lecture()       //Function 2
  {
    cout << "Lectures in the seminar on" << endl;
  }
 Seminar(int duration)       //Function 3 will be called by Seminar s2 ((Parameterized Constructor)
  {
    time = duration;
    cout << "Seminar starts now" << endl;
  }
 ~Seminar()       //Function 4 Destructor
  {
    cout << "Thanks" << endl;
  }
};
```

i. Write statements in C++ that would execute Function 1 and Function 3 of class Seminar.

Answer-

```
SEMINAR s1   //Execute function 1. Default Constructor
SEMINAR s2(30)    //Execute function 3. Parameterized constructor
```

ii. In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called?

Function 4 Referred as Destructor, it is invoked as soon as the scope of the object
gets over.

iii. In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together?

Answer-Constructor Overloading (Polymorphism) where the Constructor with same name is displayed in different forms because their abilities can be assigned & used in different ways.

2.Answer the questions (i) and (ii) after going through the following class:

```
class Test
{
   char paper[20];
   int marks;
public:
   Test ()    // Function 1  will be called by Test t1. (Default constructor )
 {
      strcpy (paper, "Computer");
      marks = 0;
   }

   Test (char p[])    // Function 2  will be called by Constructor Test t2 (Parameterized Constructor )
   {
      strcpy(paper, p);
      marks = 0;
   }

Test (int m)    // Function 3  will be called by Constructor Test t3 (Parameterized Constructor )

   {
      strcpy(paper,"Computer");
      marks = m;
   }

   Test (char p[], int m)    // Function 4 will be called by Constructor Test t4 (Parameterized
Constructor )

   {
      strcpy (paper, p);
      marks = m;
   }
};
```

i.Write statements in C++ that would execute Function 1, Function 2, Function 3 and Function 4 of class Test.

Function 1 will be called by Test t1. (Default constructor)
Function 2 will be called by Constructor Test t2 (Parameterized Constructor)
Function 3 will be called by Constructor Test t3 (Parameterized Constructor)
Function 4 will be called by Constructor Test t4 (Parameterized Constructor)

ii.Which feature of Object Oriented Programming is demonstrated using Function 1, Function 2, Function 3 and Function 4 together in the above class Test?

1-Constructor Overloading (Polymorphism) where the Constructor with same name is displayed in different forms

3.Consider the definition of the following class:

```
class Sample
{
private:
   int x;
   double y;
public :
   Sample(); //Constructor 1
   Sample(int); //Constructor 2
   Sample(int, int); //Constructor 3
   Sample(int, double); //Constructor 4
};
```

i.Write the definition of the constructor 1 so that the private member variables are initialized to 0.

```
   i.   Sample :: Sample()
{
    x = 0;
    y = 0;
}
```

ii.Write the definition of the constructor 2 so that the private member variable x is initialized according to the value of the parameter, and the private member variable y is initialized to 0.

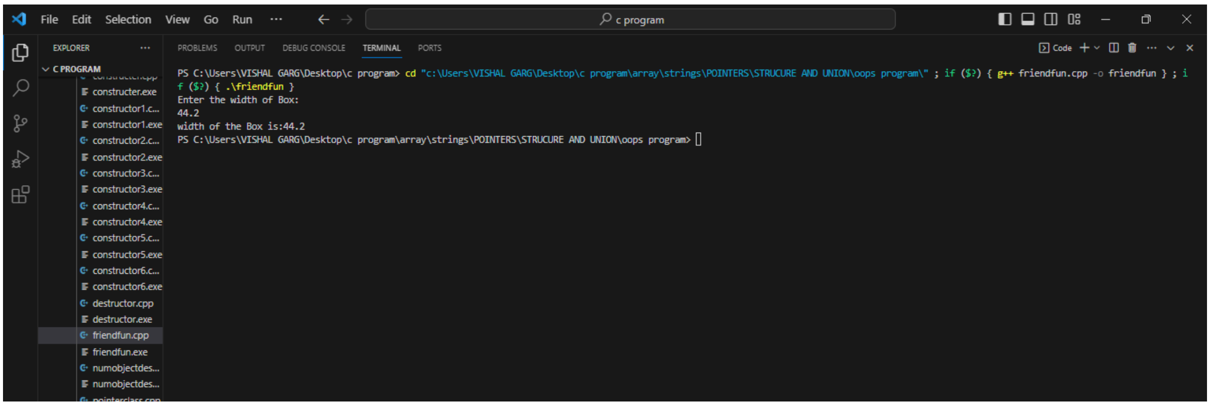
```
Sample :: Sample(int a)
{
    x = a;
    y = 0;
}
```

iii.Write the definition of the constructors 3 and 4 so that the private member variables are initialized according to the values of the parameters.

```
Sample :: Sample(int a, int b)
{
    x = a;
    y = b;
}
Sample :: Sample(int a, double b)
{
    x = a;
    y = b;
}
```

4. Create a class called Box with a variable: width of type double. Inside the class define a constructor and a friend that prints the width value(printWidth). In the main() define a Box instance, set values and call printWidth.

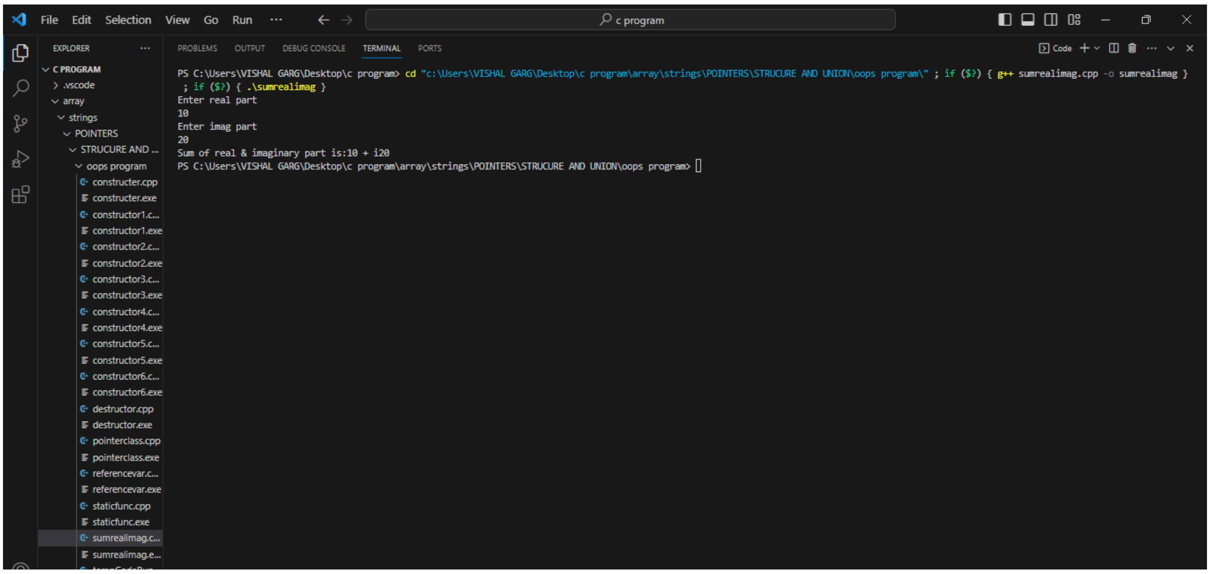
```
#include <iostream>
using namespace std;
class box
{
    double width;

public:
    box()
    {
        cout << "Enter the width of Box:\n";
        cin >> width;
    }
    friend void printwidth(const box &);
};
void printwidth(const box &b1)
{
    cout << "width of the Box is:" << b1.width << endl;
}
int main()
{
    box b;
    printwidth(b);
}
```



```
PS C:\Users\VISHAL GARG\Desktop\c program> cd "c:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program\" ; if ($?) { g++ friendfun.cpp -o friendfun } ; i
f ($?) { .\friendfun }
Enter the width of Box:
44.2
width of the Box is:44.2
PS C:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program> 
```

5. Perform addition operation on complex data using class and object. The program should ask for real and imaginary part of two complex numbers, and display the real and imaginary parts of their sum.

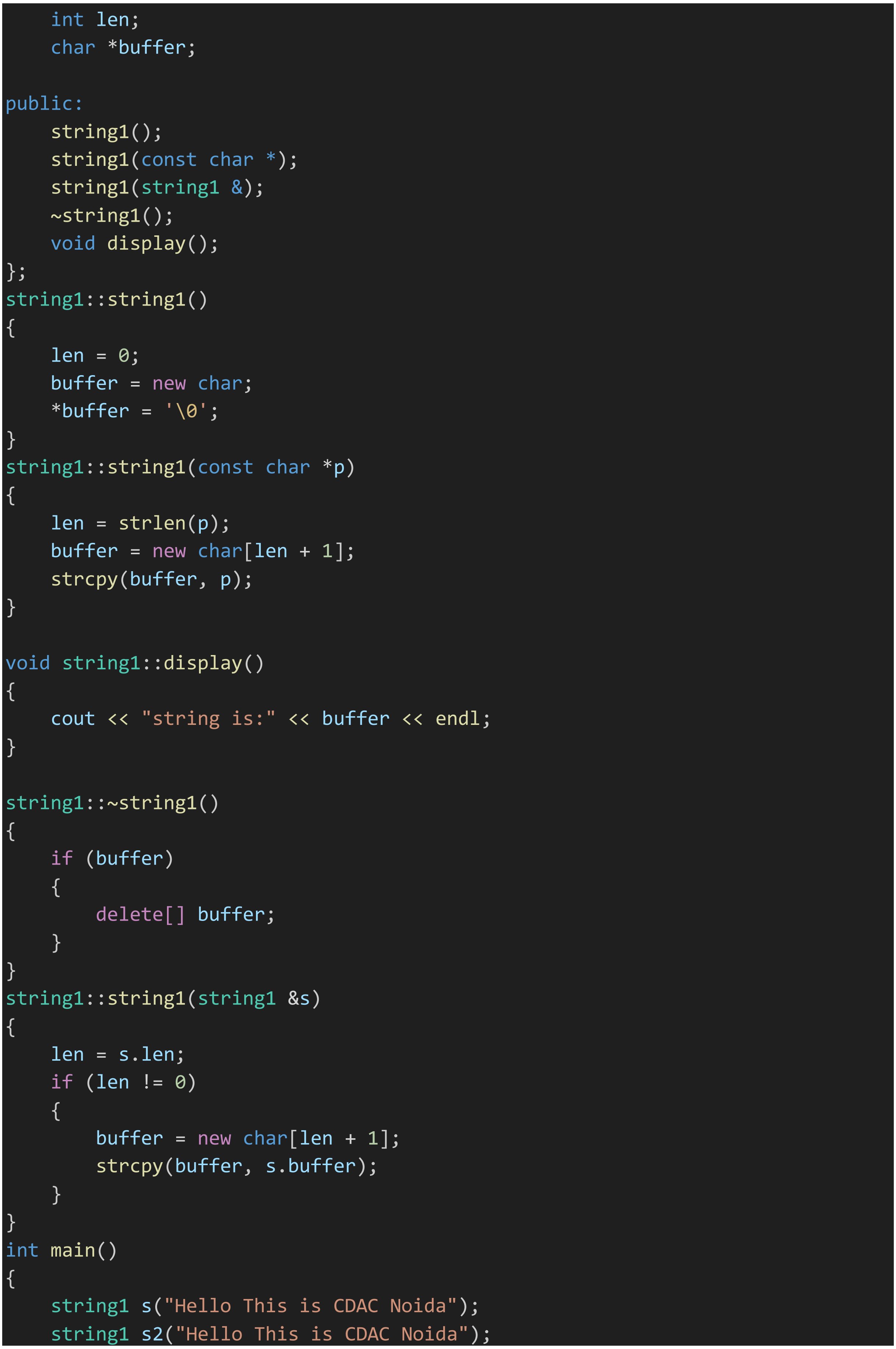
```
#include <iostream>
using namespace std;
class Complex1
{
    int real; // data members
    int imag;

public:
    Complex1();
    void display();
};
Complex1::Complex1() // default constructor
{
    cout << "Enter real part" << endl;
    cin >> real;
    cout << "Enter imag part" << endl;
    cin >> imag;
}

void Complex1::display()
{

    cout << "Sum of real & imaginary part is:" << real << " + i" << this->imag
<< endl;
}
int main()
{
    Complex1 c1;
    c1.display();
    return 0;
}
```



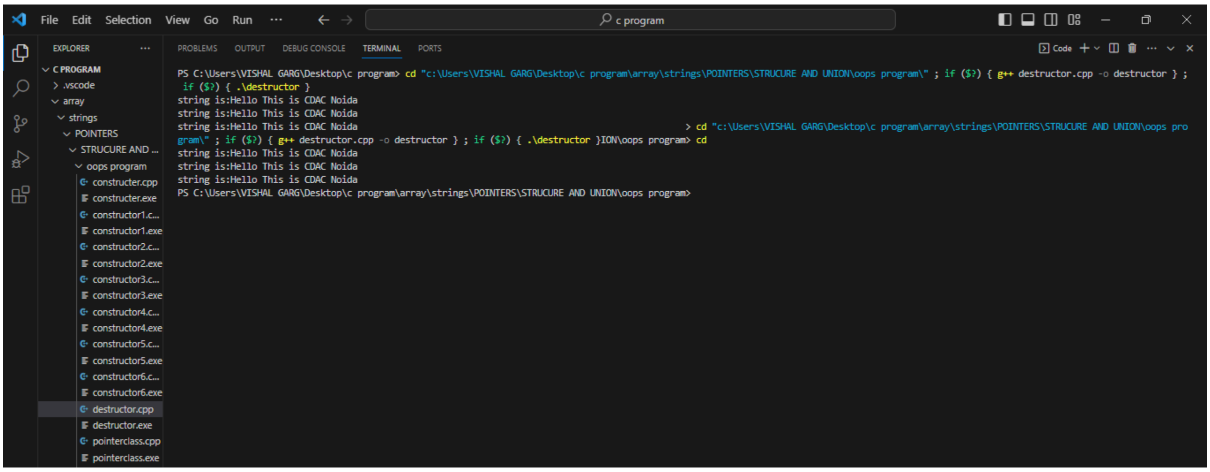
```
PS C:\Users\VISHAL GARG\Desktop\c program> cd "c:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program\" ; if ($?) { g++ sumrealimag.cpp -o sumrealimag }
 ; if ($?) { .\sumrealimag }
Enter real part
10
Enter imag part
20
Sum of real & imaginary part is:10 + i20
PS C:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program>
```

6-Write a program in C++ having class string1 with members as
Int length;
Char * buffer;
Implement this class using copy constructor, destructor , parameterized constructor and default constructor.

```
#include <iostream>
#include <string.h>
using namespace std;
class string1
{
```



```
    int len;
    char *buffer;

public:
    string1();
    string1(const char *);
    string1(string1 &);
    ~string1();
    void display();
};
string1::string1()
{
    len = 0;
    buffer = new char;
    *buffer = '\0';
}
string1::string1(const char *p)
{
    len = strlen(p);
    buffer = new char[len + 1];
    strcpy(buffer, p);
}

void string1::display()
{
    cout << "string is:" << buffer << endl;
}

string1::~string1()
{
    if (buffer)
    {
        delete[] buffer;
    }
}
string1::string1(string1 &s)
{
    len = s.len;
    if (len != 0)
    {
        buffer = new char[len + 1];
        strcpy(buffer, s.buffer);
    }
}
int main()
{
    string1 s("Hello This is CDAC Noida");
    string1 s2("Hello This is CDAC Noida");
    {
        string1 s3(s2);
        s3.display();
    }
    s.display();
    s2.display();
    return 0;
}
```

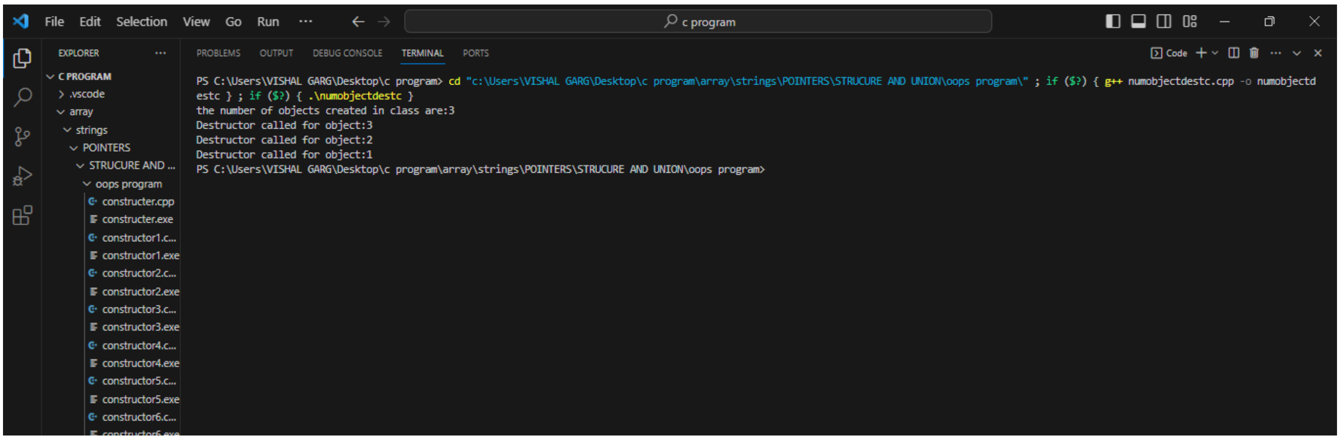


```
PS C:\Users\VISHAL GARG\Desktop\c program> cd "c:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program\" ; if ($?) { g++ destructor.cpp -o destructor } ;
 if ($?) { .\destructor }
string is:Hello This is CDAC Noida
string is:Hello This is CDAC Noida
string is:Hello This is CDAC Noida                                              > cd "c:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops pro
gram\" ; if ($?) { g++ destructor.cpp -o destructor } ; if ($?) { .\destructor }ION\oops program> cd
string is:Hello This is CDAC Noida
string is:Hello This is CDAC Noida
string is:Hello This is CDAC Noida
PS C:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program>
```

7-Write a program in C++ that counts the number of objects created in the class. Illustrate the program with the constructor and destructor both.

```
#include <iostream>
using namespace std;
class object_count
{
    static int count;

public:
    object_count() // Default constructor
    {
        count++;
    }
    ~object_count() // Destructor
    {
        cout << "Destructor called for object:" << count << endl;
        count--;
    }
    void display()
    {
        cout << "the number of objects created in class are:" << count <<
endl;
    }
};
int object_count::count;
int main()
```

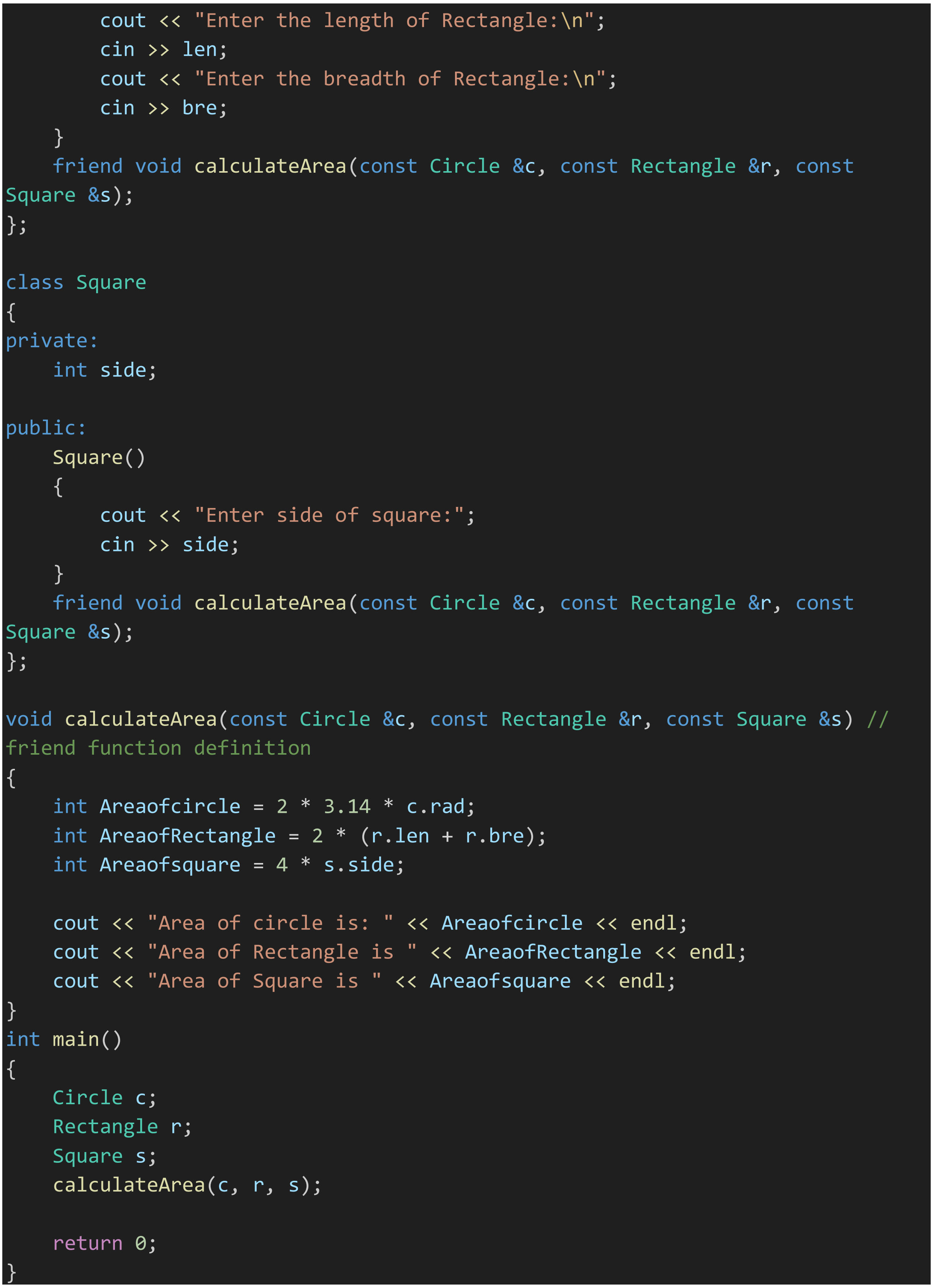
```
{
    object_count c1;
    object_count c2;
    object_count c3;
    c3.display();
    return 0;
}
```



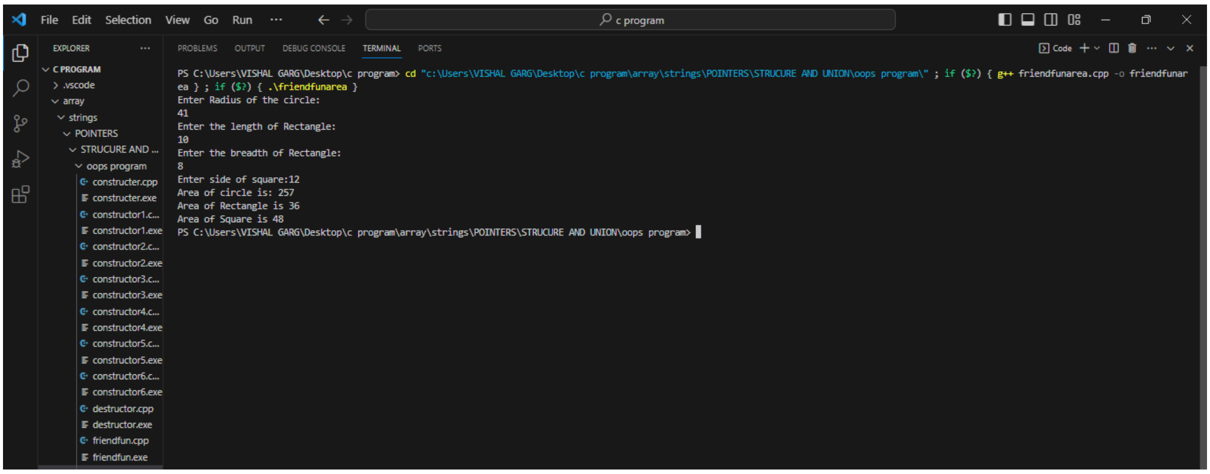
```
PS C:\Users\VISHAL GARG\Desktop\c program> cd "c:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program\" ; if ($?) { g++ numobjectdestc.cpp -o numobjectd
estc } ; if ($?) { .\numobjectdestc }
the number of objects created in class are:3
Destructor called for object:3
Destructor called for object:2
Destructor called for object:1
PS C:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program>
```

8.Write a program in C++ to calculate perimeter of all figures using the concept of friend function.

```
#include <iostream>
using namespace std;
class Rectangle; // forward declearation
class Square;    // forward declearation
class Circle
{
private:
    int rad;

public:
    Circle() // default constructor
    {
        cout << "Enter Radius of the circle:\n";
        cin >> rad;
    }
    friend void calculateArea(const Circle &c, const Rectangle &r, const
Square &s); // friends function
};

class Rectangle
{
private:
    int len, bre;

public:
    Rectangle()
    {
```



```
        cout << "Enter the length of Rectangle:\n";
        cin >> len;
        cout << "Enter the breadth of Rectangle:\n";
        cin >> bre;
    }
    friend void calculateArea(const Circle &c, const Rectangle &r, const
Square &s);
};

class Square
{
private:
    int side;

public:
    Square()
    {
        cout << "Enter side of square:";
        cin >> side;
    }
    friend void calculateArea(const Circle &c, const Rectangle &r, const
Square &s);
};

void calculateArea(const Circle &c, const Rectangle &r, const Square &s) //
friend function definition
{
    int Areaofcircle = 2 * 3.14 * c.rad;
    int AreaofRectangle = 2 * (r.len + r.bre);
    int Areaofsquare = 4 * s.side;

    cout << "Area of circle is: " << Areaofcircle << endl;
    cout << "Area of Rectangle is " << AreaofRectangle << endl;
    cout << "Area of Square is " << Areaofsquare << endl;
}
int main()
{
    Circle c;
    Rectangle r;
    Square s;
    calculateArea(c, r, s);

    return 0;
}
```



```
PS C:\Users\VISHAL GARG\Desktop\c program> cd "c:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program\" ; if ($?) { g++ friendfunarea.cpp -o friendfunar
ea } ; if ($?) { .\friendfunarea }
Enter Radius of the circle:
41
Enter the length of Rectangle:
10
Enter the breadth of Rectangle:
8
Enter side of square:12
Area of circle is: 257
Area of Rectangle is 36
Area of Square is 48
PS C:\Users\VISHAL GARG\Desktop\c program\array\strings\POINTERS\STRUCURE AND UNION\oops program> 
```