

Lingaya's Vidyapeeth, Faridabad

(Deemed to be University under Section 3 of UGC Act, 1956)

Object-Oriented programming using C++

LAB MANUAL

BCA - 2nd Semester



Roll no = 21BCA44

Branch = BCA

Name = Deepak Kumar

DEPT. OF COMPUTER SCIENCE & ENGINEERING

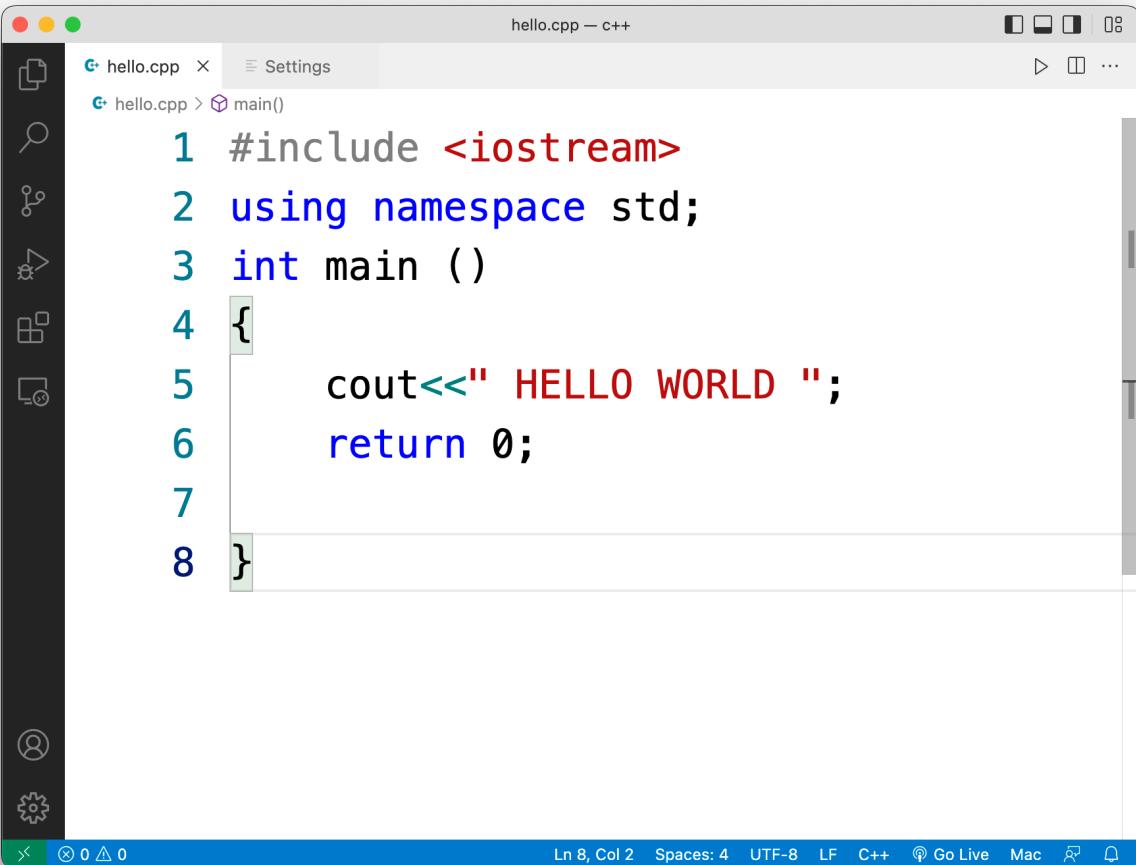
INDEX

Series	Programs	Remark
1	Write a program to print Hello World.	
2	<i>> Write a program to print your age.</i>	
3	<i>> Write a program to Check Whether the Number is Even or Odd.</i>	
4	<i>> Write a program to print result of a Student.</i>	
5	<i>> Write a program to use Switch Statement.</i>	
6	<i>> Write a program to print First five Natural Numbers using For loop.</i>	
7	<i>> Write a program to use While loop.</i>	
8	<i>> Write a program to use do-while Statement.</i>	
9	Write a program to use break Statement.	

Experiment – 01

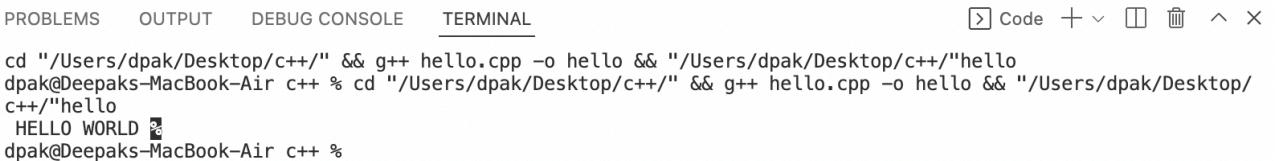
Aim :- Write a program to print Hello C++

Source Code :-



```
hello.cpp — c++
hello.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main ()
4 {
5     cout<<" HELLO WORLD ";
6     return 0;
7 }
8 }
```

Output:-

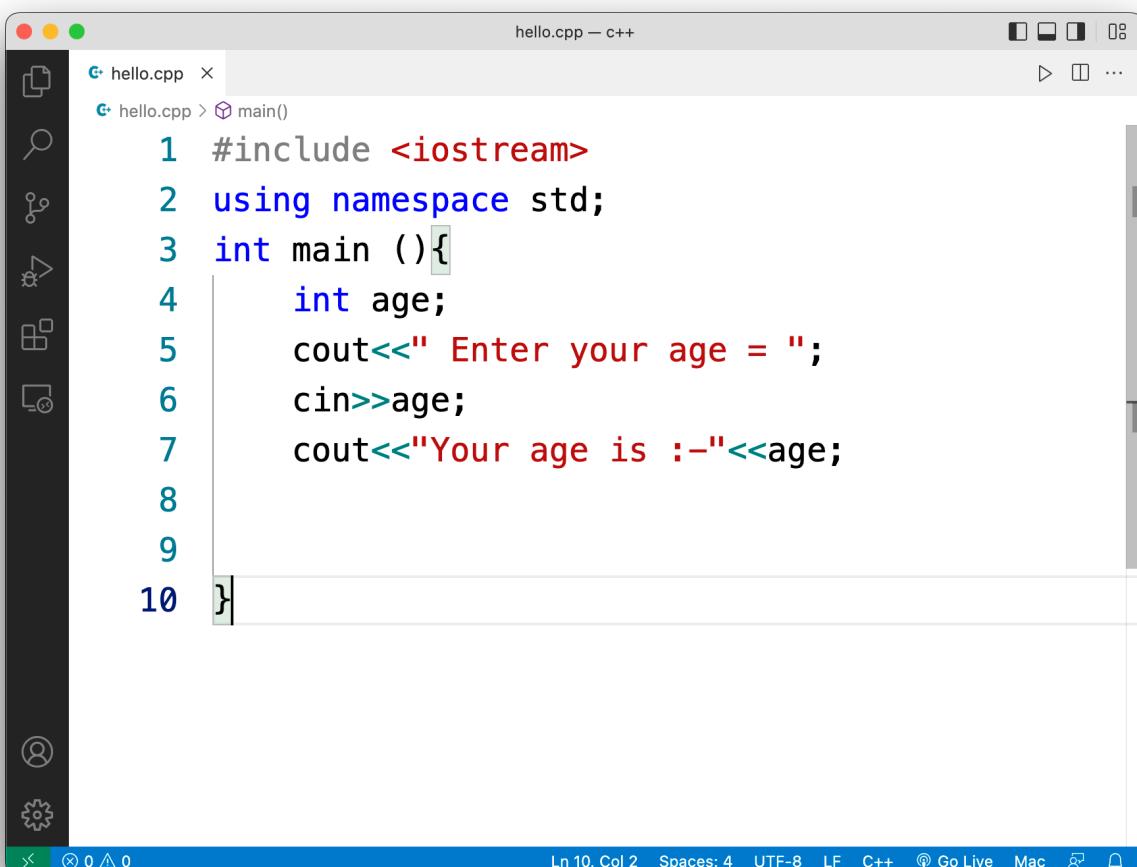


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello
HELLO WORLD %
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 02

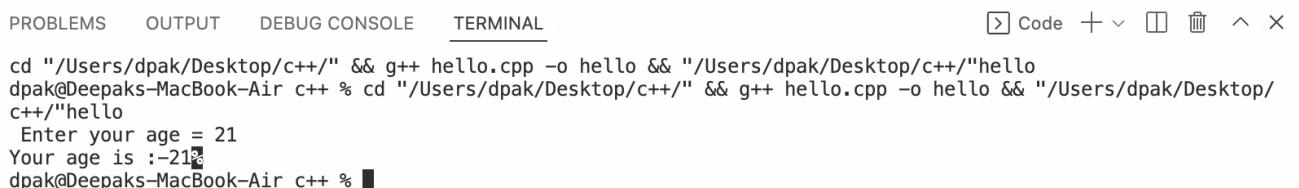
Aim :- Write a program to print your age.

Source Code :-



```
hello.cpp — c++
hello.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main (){
4     int age;
5     cout<<" Enter your age = ";
6     cin>>age;
7     cout<<"Your age is :-"<<age;
8
9
10 }
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello
Enter your age = 21
Your age is :-21%
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 03

Aim :- Write a program to find weather the number is odd or even.

Source Code :-

The screenshot shows a code editor window titled "hello.cpp — c++". The left sidebar contains icons for file operations, search, and settings. The main area displays the following C++ code:

```
1 #include <iostream>
2 using namespace std;
3 int main (){
4     int a;
5     cout<<" Enter your Number = ";
6     cin>>a;
7     if (a % 2 == 0 )
8     {
9         cout<<"It is Even Number";
10    }
11    else {
12        cout<< "It is Odd Number";
13    }
14    return 0;
15
16 }
```

The code uses standard input and output streams to prompt the user for a number and then determine if it is even or odd. The output is displayed in red text.

Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

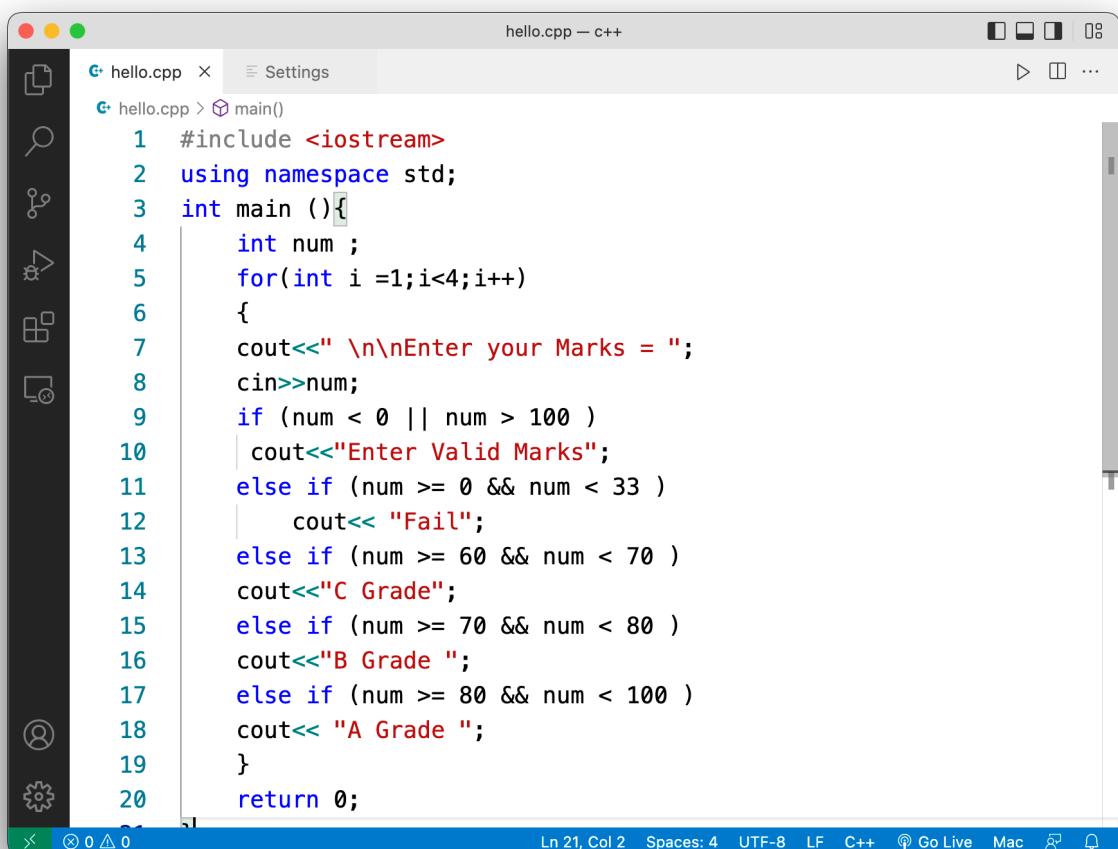
Code + ▾

```
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/hello"
Enter your Number = 15
It is Odd Number
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 04

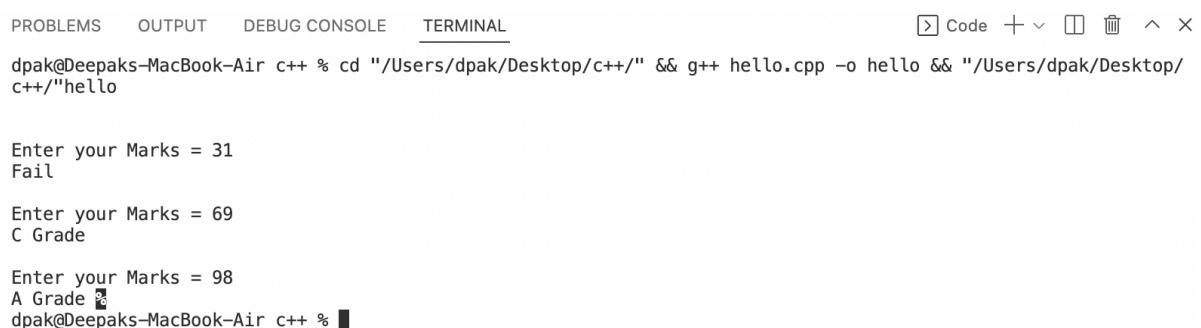
Aim :- Write a program to calculate the grade of student.

Source Code :-



```
hello.cpp — c++
hello.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main (){
4     int num ;
5     for(int i =1;i<4;i++)
6     {
7         cout<<" \n\nEnter your Marks = ";
8         cin>>num;
9         if (num < 0 || num > 100 )
10            cout<<"Enter Valid Marks";
11        else if (num >= 0 && num < 33 )
12            cout<< "Fail";
13        else if (num >= 60 && num < 70 )
14            cout<<"C Grade";
15        else if (num >= 70 && num < 80 )
16            cout<<"B Grade ";
17        else if (num >= 80 && num < 100 )
18            cout<< "A Grade ";
19        }
20    return 0;
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello

Enter your Marks = 31
Fail

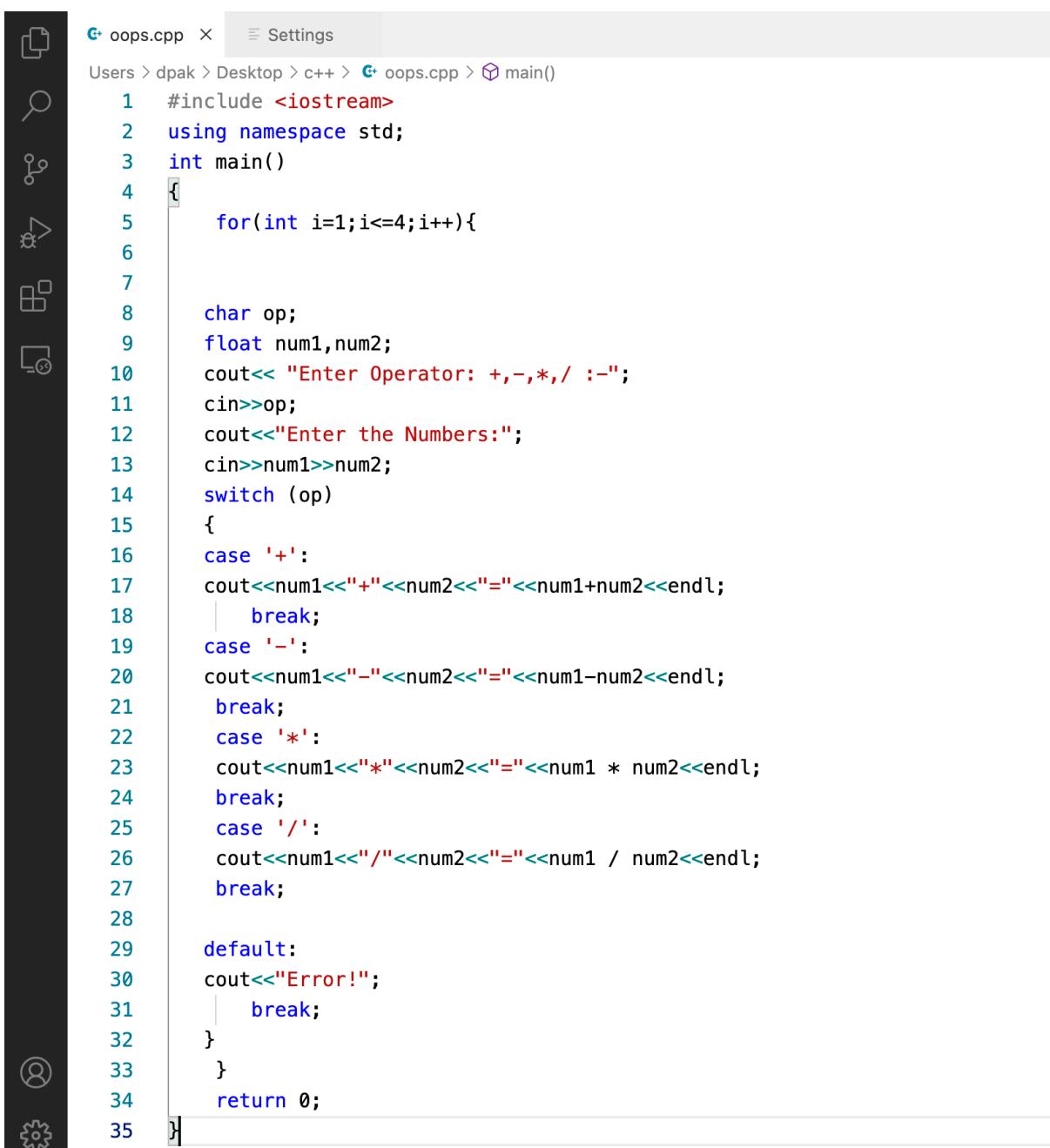
Enter your Marks = 69
C Grade

Enter your Marks = 98
A Grade
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 05

Aim :- Write a program to use switch statement.

Source Code :-



The screenshot shows a code editor window with a dark theme. On the left is a vertical toolbar with icons for file operations, search, and other development tools. The main area displays the following C++ code:

```
oops.cpp  ×  Settings
Users > dpak > Desktop > c++ > oops.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     for(int i=1;i<=4;i++){
6
7         char op;
8         float num1,num2;
9         cout<< "Enter Operator: +,-,*,/ :-";
10        cin>>op;
11        cout<<"Enter the Numbers:";
12        cin>>num1>>num2;
13        switch (op)
14        {
15            case '+':
16                cout<<num1<<"+"<<num2<<"="<<num1+num2<<endl;
17                break;
18            case '-':
19                cout<<num1<<"-"<<num2<<"="<<num1-num2<<endl;
20                break;
21            case '*':
22                cout<<num1<<"* "<<num2<<"="<<num1 * num2<<endl;
23                break;
24            case '/':
25                cout<<num1<<"/"<<num2<<"="<<num1 / num2<<endl;
26                break;
27
28            default:
29                cout<<"Error!";
30                break;
31        }
32    }
33
34    return 0;
35 }
```

Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
dpak@Deepaks-MacBook-Air ~ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
Enter Operator: +,-,*,/ :+ 
Enter the Numbers:23 45
23+45=68
Enter Operator: +,-,*,/ :-*
Enter the Numbers:25 2
25*2=50
Enter Operator: +,-,*,/ :--
Enter the Numbers:50 25
50-25=25
Enter Operator: +,-,*,/ :/
Enter the Numbers:50 2
50/2=25
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 06

Aim :- Write a program to print first 5 natural number using for loop.

Source Code :-

```
hello.cpp — c++
hello.cpp X Settings
hello.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main (){
4     int i;
5     for(int i=1;i<=5;i++)
6         cout<<"\n"<<i;
7     return 0;
8 }
```

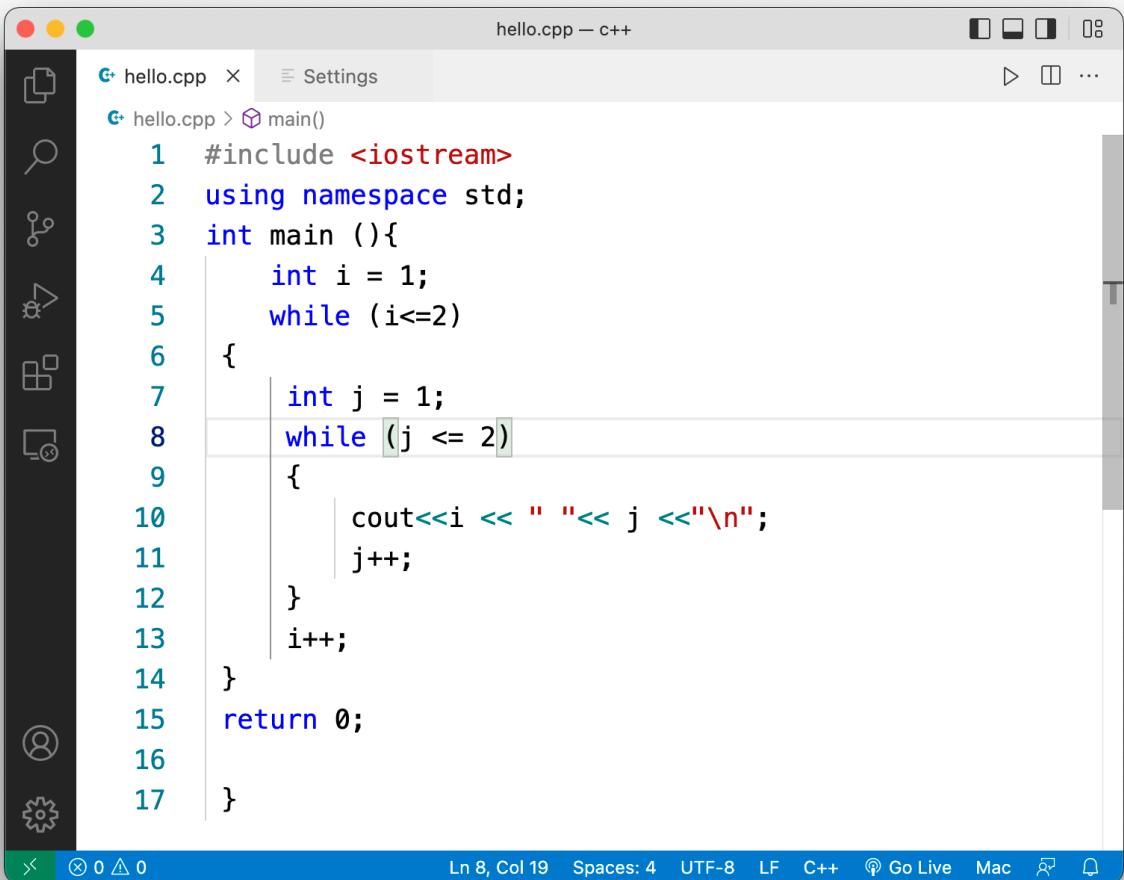
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o he
llo && "/Users/dpak/Desktop/c++/"hell
0
1
2
3
4
5%
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 07

Aim :- Write a program to use while loop.

Source Code :-

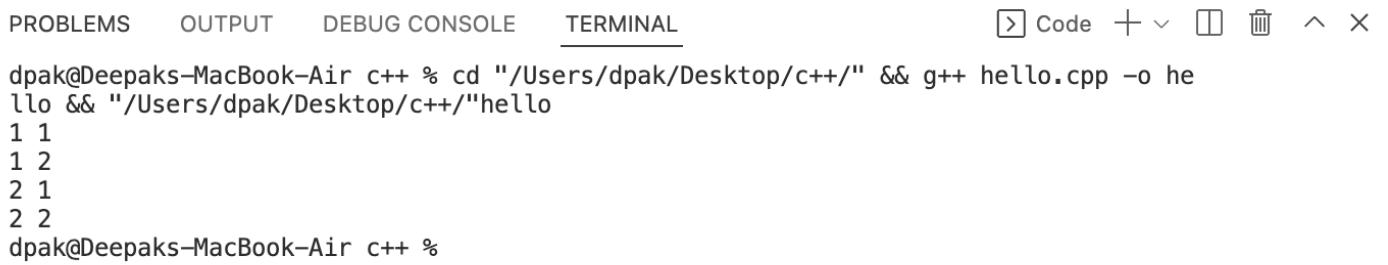


The screenshot shows a code editor window titled "hello.cpp — c++". The left sidebar contains icons for file operations like Open, Save, Find, and Settings. The main pane displays the following C++ code:

```
#include <iostream>
using namespace std;
int main (){
    int i = 1;
    while (i<=2)
    {
        int j = 1;
        while (j <= 2)
        {
            cout<<i << " " << j <<"\n";
            j++;
        }
        i++;
    }
    return 0;
}
```

The code uses nested while loops to print a 2x2 grid of numbers. The terminal status bar at the bottom indicates the code has 17 lines, 0 errors, and 0 warnings, and is in Ln 8, Col 19. The status bar also shows settings like Spaces: 4, UTF-8, LF, C++, Go Live, Mac, and a refresh icon.

Output:-



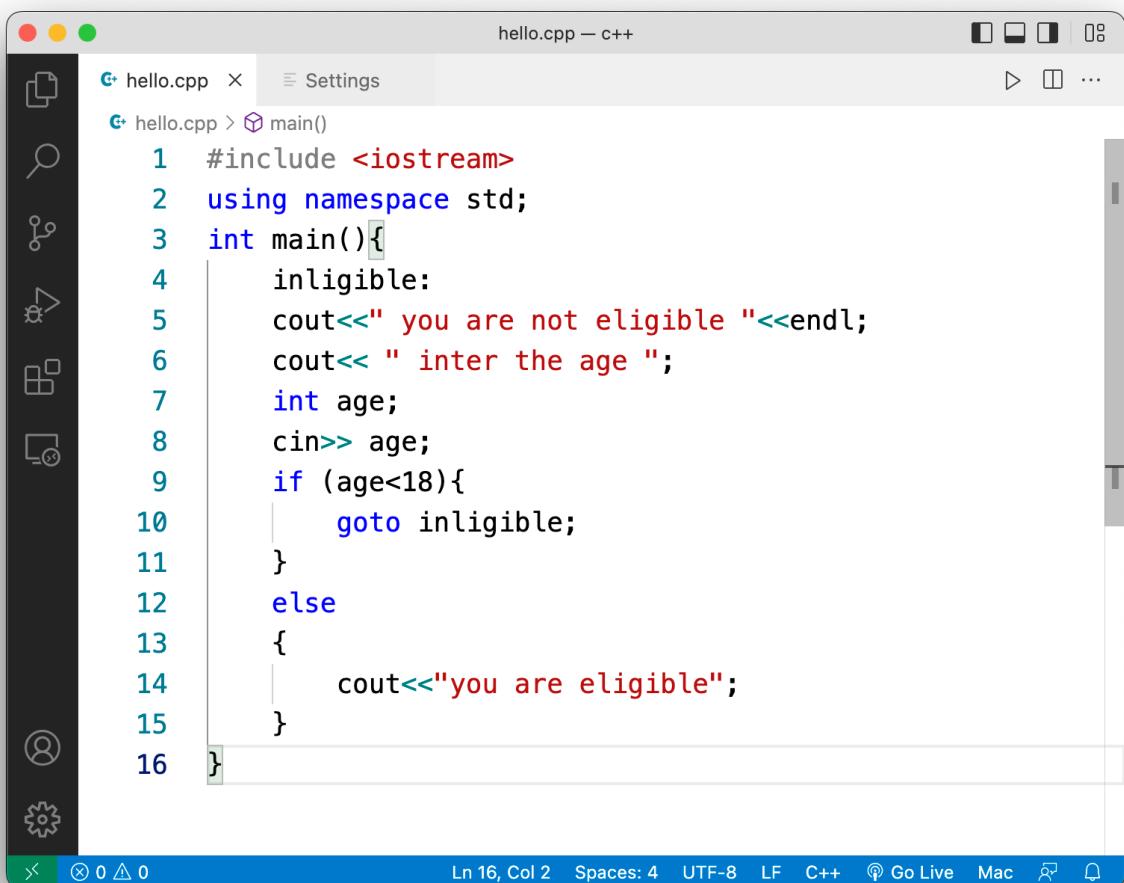
The terminal window shows the command to compile and run the "hello.cpp" program. The output displays the following 4 lines of text:

```
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello
1 1
1 2
2 1
2 2
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 08

Aim :- Write a program to use goto statement.

Source Code :-

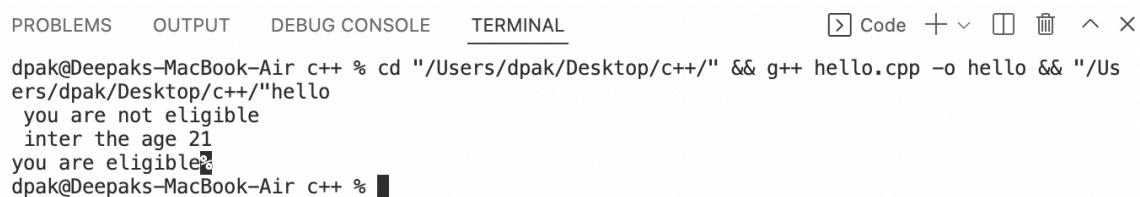


The screenshot shows a code editor window titled "hello.cpp — c++". The left sidebar contains icons for file operations like Open, Save, Find, and Settings. The main pane displays the following C++ code:

```
hello.cpp x Settings
hello.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main(){
4     ineligible:
5     cout<<" you are not eligible "<<endl;
6     cout<< " inter the age ";
7     int age;
8     cin>> age;
9     if (age<18){
10         goto ineligible;
11     }
12     else
13     {
14         cout<<"you are eligible";
15     }
16 }
```

The code uses a goto statement to skip the age input and output for users under 18. The code editor interface includes a status bar at the bottom showing "Ln 16, Col 2" and other settings.

Output:-



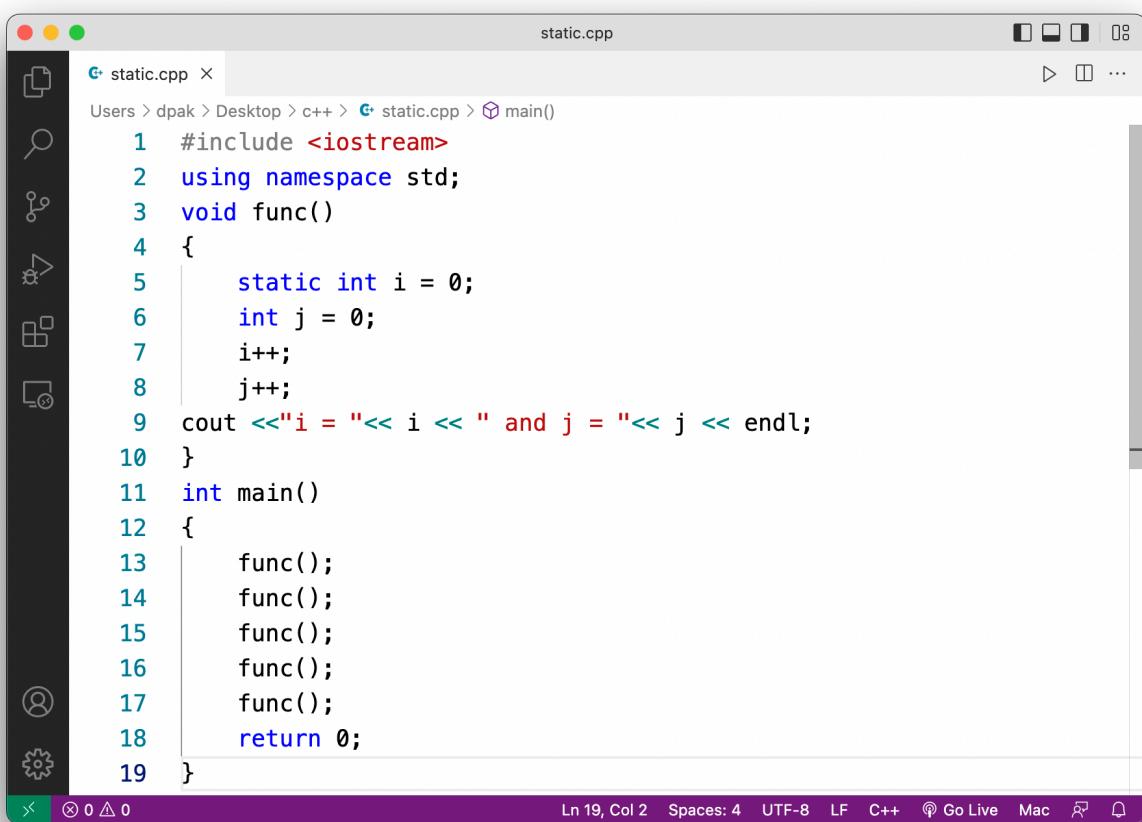
The screenshot shows a terminal window with the following command and output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ hello.cpp -o hello && "/Users/dpak/Desktop/c++/"hello
you are not eligible
inter the age 21
you are eligible
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 09

Aim :- Write a program to use Static Keyword.

Source Code :-

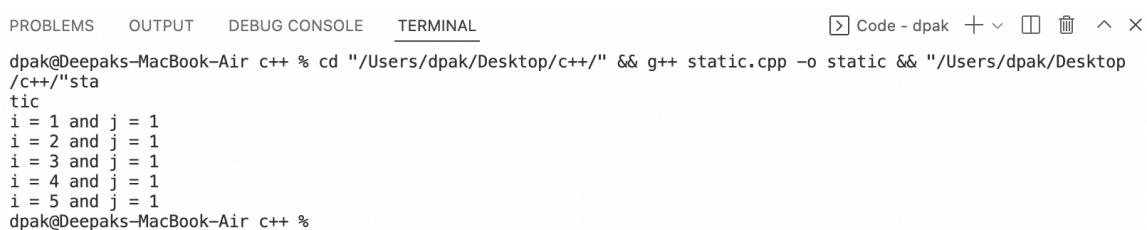


The screenshot shows a code editor window titled "static.cpp". The file path is "Users > dpak > Desktop > c++ > static.cpp > main()". The code is as follows:

```
static.cpp
static int i = 0;
int j = 0;
void func()
{
    static int i = 0;
    int j = 0;
    i++;
    j++;
    cout << "i = " << i << " and j = " << j << endl;
}
int main()
{
    func();
    func();
    func();
    func();
    func();
    return 0;
}
```

The status bar at the bottom shows "Ln 19, Col 2" and "Spaces: 4".

Output:-



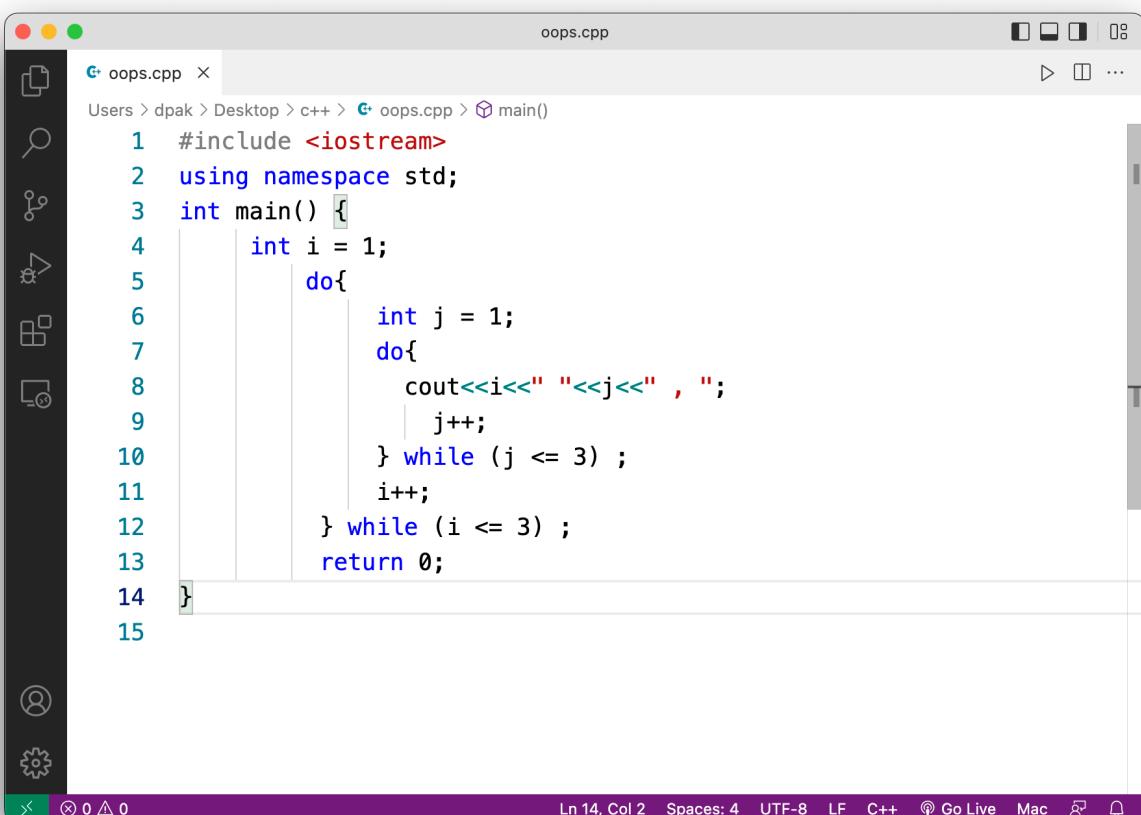
The terminal output shows the execution of the program:

```
dpak@Deepaks-MacBook-Air ~ % cd "/Users/dpak/Desktop/c++/" && g++ static.cpp -o static && "/Users/dpak/Desktop/c++/"static
i = 1 and j = 1
i = 2 and j = 1
i = 3 and j = 1
i = 4 and j = 1
i = 5 and j = 1
dpak@Deepaks-MacBook-Air ~ %
```

Experiment – 10

Aim :- Write a program to use do - while Statement.

Source Code :-



```
oops.cpp
Users > dpak > Desktop > c++ > oops.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int i = 1;
5     do{
6         int j = 1;
7         do{
8             cout<<i<<" "<<j<<" , ";
9             j++;
10        } while (j <= 3) ;
11        i++;
12    } while (i <= 3) ;
13    return 0;
14 }
15
```

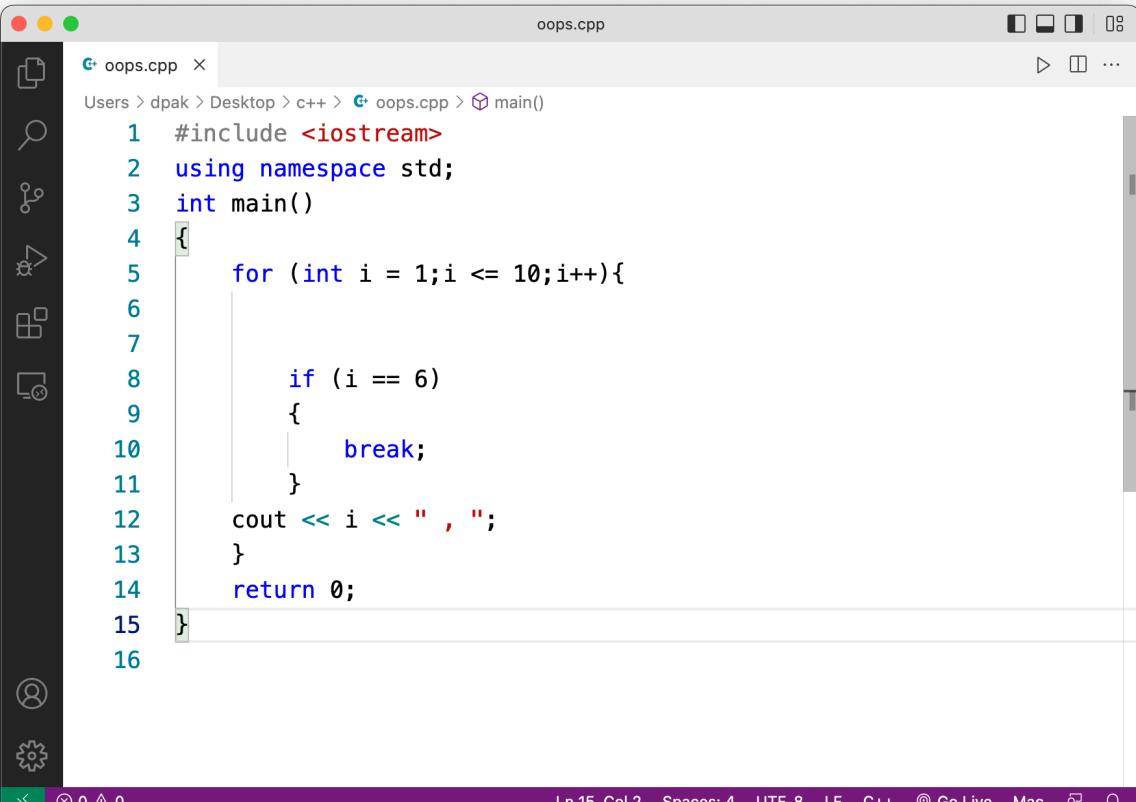
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/oops"
1 1 , 1 2 , 1 3 , 2 1 , 2 2 , 2 3 , 3 1 , 3 2 , 3 3 ,
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 11

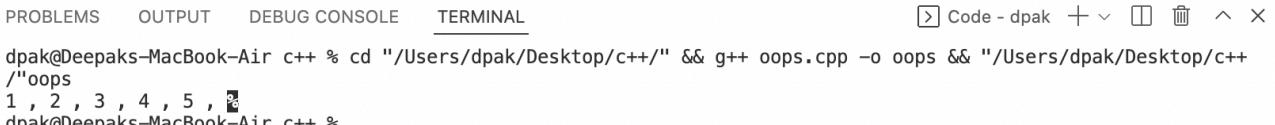
Aim :- Write a program to use break Statement.

Source Code :-



```
oops.cpp
Users > dpak > Desktop > c++ > oops.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     for (int i = 1;i <= 10;i++){
6
7         if (i == 6)
8         {
9             break;
10        }
11        cout << i << " , ";
12    }
13    return 0;
14 }
15
16
```

Output:-

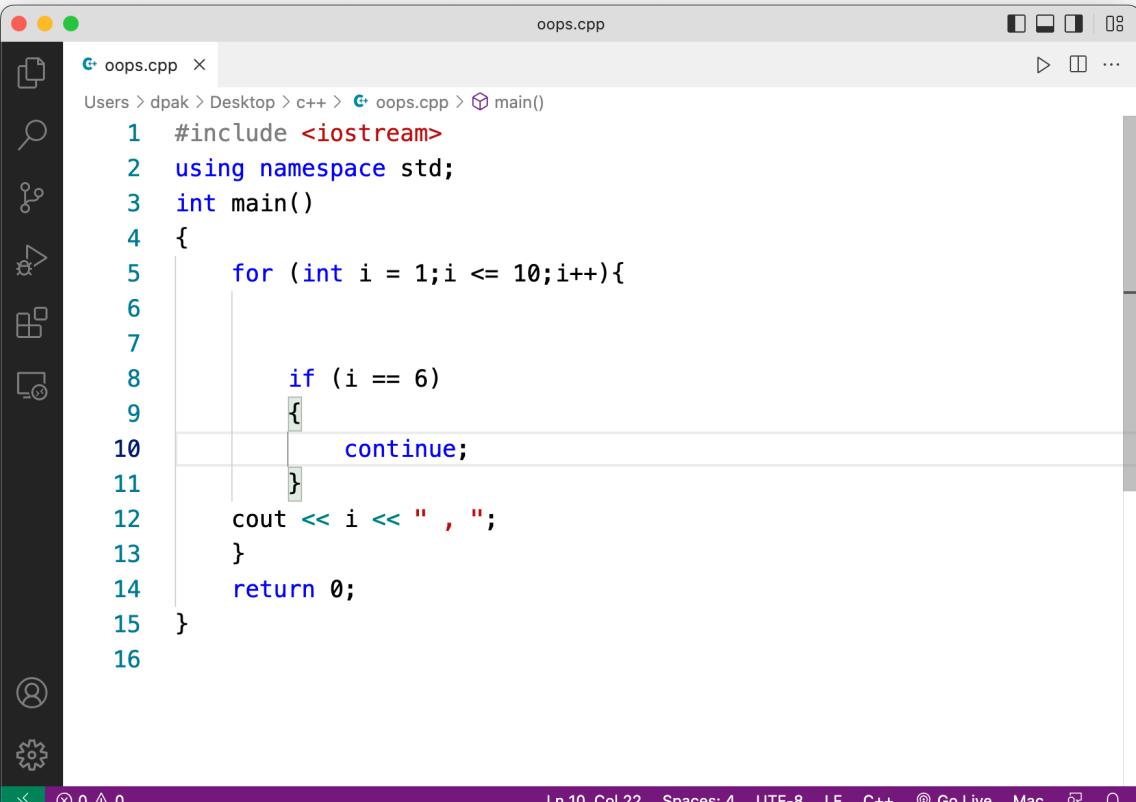


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code - dpak + ∨ ×
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && ./oops
/oops
1 , 2 , 3 , 4 , 5 ,
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 12

Aim :- Write a program to use continue Statement.

Source Code :-



```
oops.cpp
Users > dpak > Desktop > c++ > oops.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     for (int i = 1; i <= 10; i++) {
6
7         if (i == 6)
8             {
9                 continue;
10            }
11
12         cout << i << " , ";
13     }
14
15 }
16
```

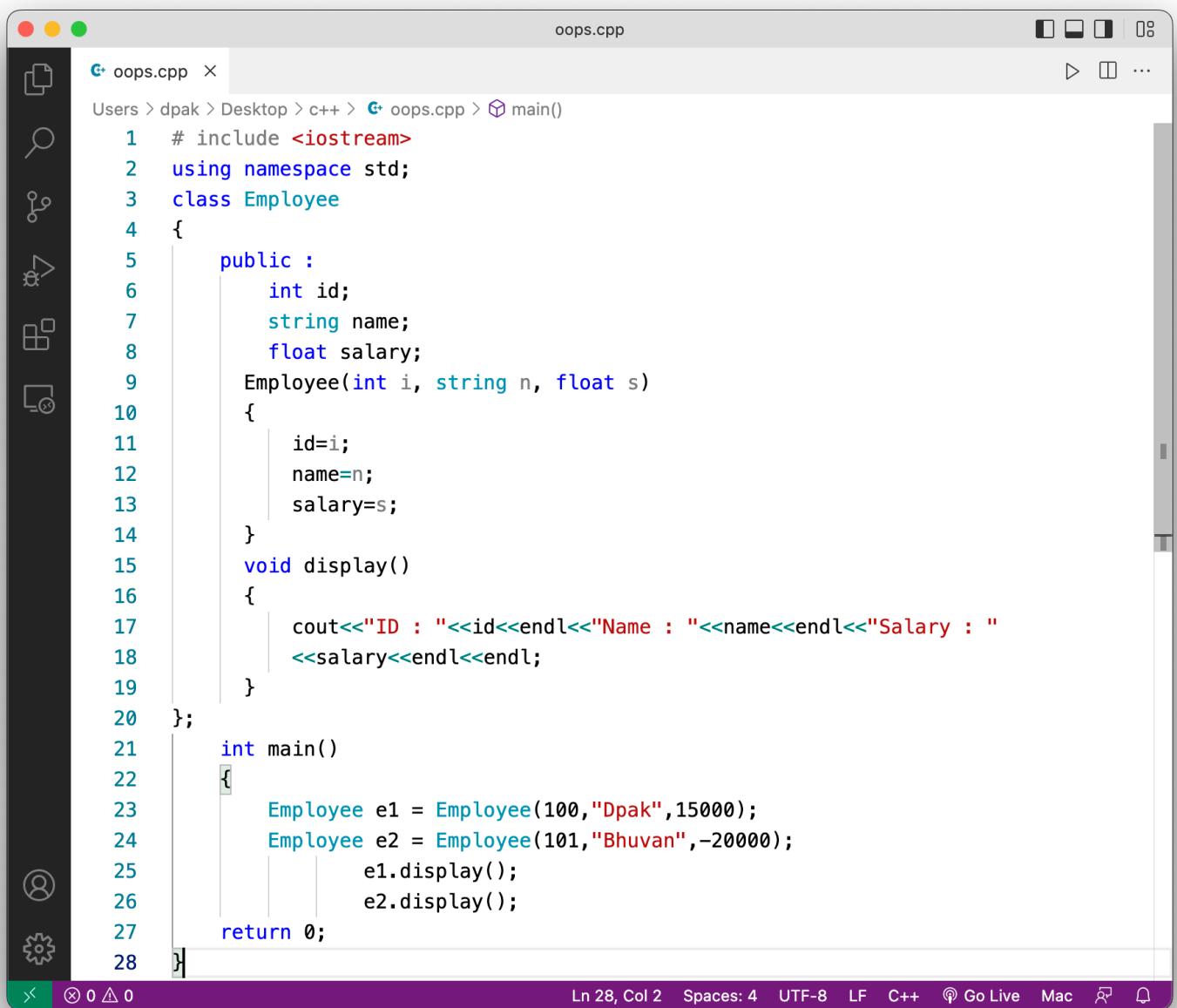
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/oops"
1 , 2 , 3 , 4 , 5 , 7 , 8 , 9 , 10 ,
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 13

Aim :- Write a program to use Constructor.

Source Code :-



The screenshot shows a code editor window titled "oops.cpp". The code defines a class Employee with a constructor and a display method. The main function creates two Employee objects and calls their display methods. The code is as follows:

```
oops.cpp
Users > dpak > Desktop > c++ > oops.cpp > main()
1 # include <iostream>
2 using namespace std;
3 class Employee
4 {
5     public :
6         int id;
7         string name;
8         float salary;
9         Employee(int i, string n, float s)
10    {
11        id=i;
12        name=n;
13        salary=s;
14    }
15    void display()
16    {
17        cout<<"ID : "<<id<<endl<<"Name : "<<name<<endl<<"Salary : "
18        <<salary<<endl<<endl;
19    }
20};
21 int main()
22 {
23     Employee e1 = Employee(100,"Dpak",15000);
24     Employee e2 = Employee(101,"Bhuvan",-20000);
25     e1.display();
26     e2.display();
27     return 0;
28 }
```

The status bar at the bottom indicates: Ln 28, Col 2 Spaces: 4 UTF-8 LF C++ ⚡ Go Live Mac 🔍 📰

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/oops
ID : 100
Name : Dpak
Salary : 15000

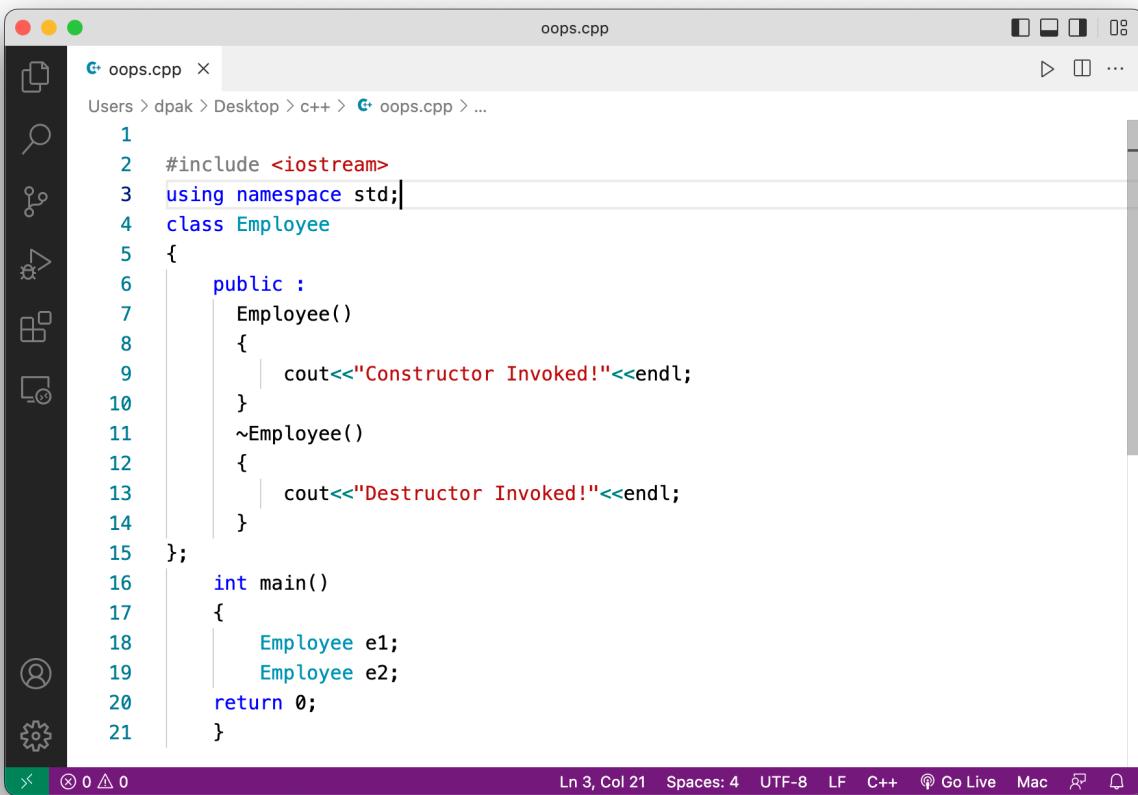
ID : 101
Name : Bhuvan
Salary : -20000

dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 14

Aim :- Write a program to use Destructor.

Source Code :-



```
oops.cpp
Users > dpak > Desktop > c++ > oops.cpp > ...
1
2 #include <iostream>
3 using namespace std;
4 class Employee
5 {
6     public :
7         Employee()
8     {
9         cout<<"Constructor Invoked!"<<endl;
10    }
11 ~Employee()
12 {
13     cout<<"Destructor Invoked!"<<endl;
14 }
15 };
16 int main()
17 {
18     Employee e1;
19     Employee e2;
20     return 0;
21 }
```

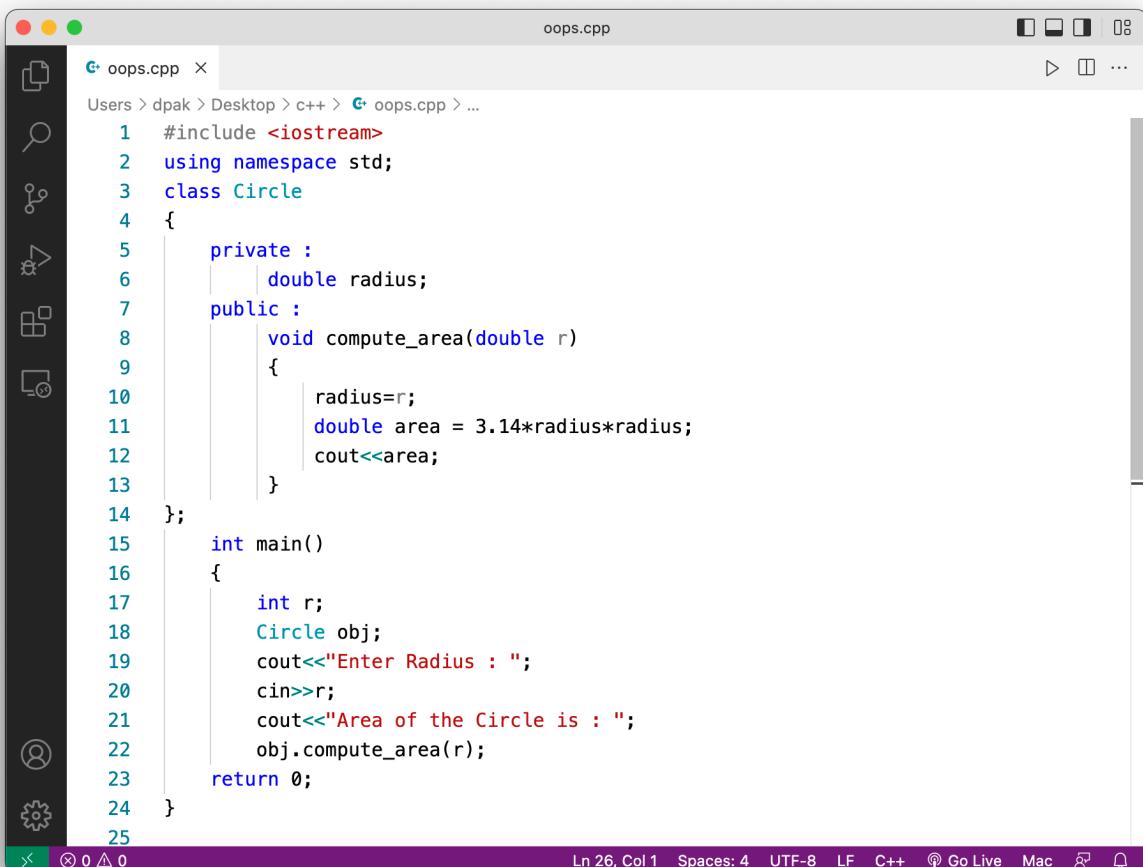
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++"
/oops
Constructor Invoked!
Constructor Invoked!
Destructor Invoked!
Destructor Invoked!
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 15

Aim :- Write a program to access Private Class Publicly.

Source Code :-



The screenshot shows a code editor window titled "oops.cpp". The code implements a class "Circle" with a private member "radius" and a public member function "compute_area" that calculates the area of a circle given its radius. The main function prompts the user for a radius and prints the computed area. The code is written in C++ and uses standard input/output streams.

```
oops.cpp
1 #include <iostream>
2 using namespace std;
3 class Circle
4 {
5     private :
6         double radius;
7     public :
8         void compute_area(double r)
9         {
10             radius=r;
11             double area = 3.14*radius*radius;
12             cout<<area;
13         }
14     };
15     int main()
16     {
17         int r;
18         Circle obj;
19         cout<<"Enter Radius : ";
20         cin>>r;
21         cout<<"Area of the Circle is : ";
22         obj.compute_area(r);
23     return 0;
24 }
```

Ln 26, Col 1 Spaces: 4 UTF-8 LF C++ ⚙ Go Live Mac ⚙ ⚙

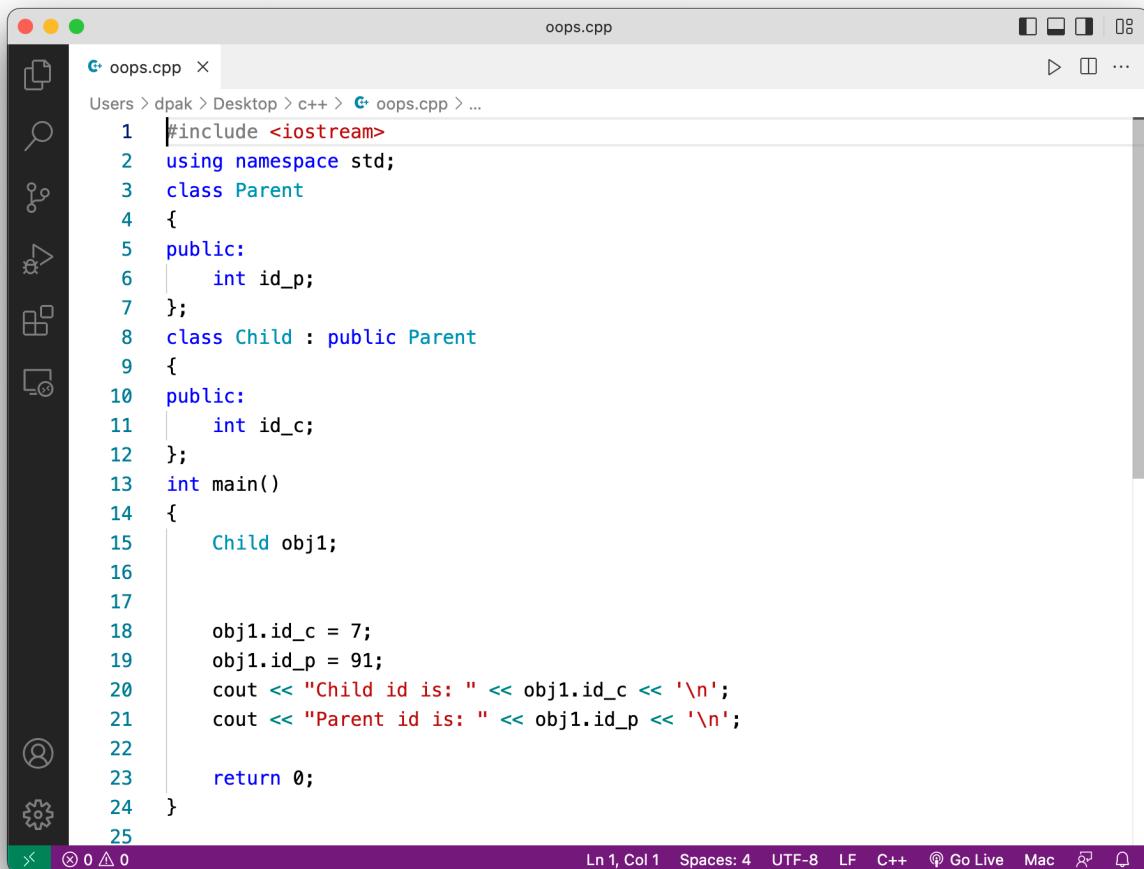
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++"
/"oops
Enter Radius : 2
Area of the Circle is : 12.56%
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 16

Aim :- Write a program to use Single Inheritance.

Source Code :-



```
oops.cpp
Users > dpak > Desktop > c++ > oops.cpp > ...
1 #include <iostream>
2 using namespace std;
3 class Parent
4 {
5 public:
6     int id_p;
7 };
8 class Child : public Parent
9 {
10 public:
11     int id_c;
12 };
13 int main()
14 {
15     Child obj1;
16
17
18     obj1.id_c = 7;
19     obj1.id_p = 91;
20     cout << "Child id is: " << obj1.id_c << '\n';
21     cout << "Parent id is: " << obj1.id_p << '\n';
22
23     return 0;
24 }
```

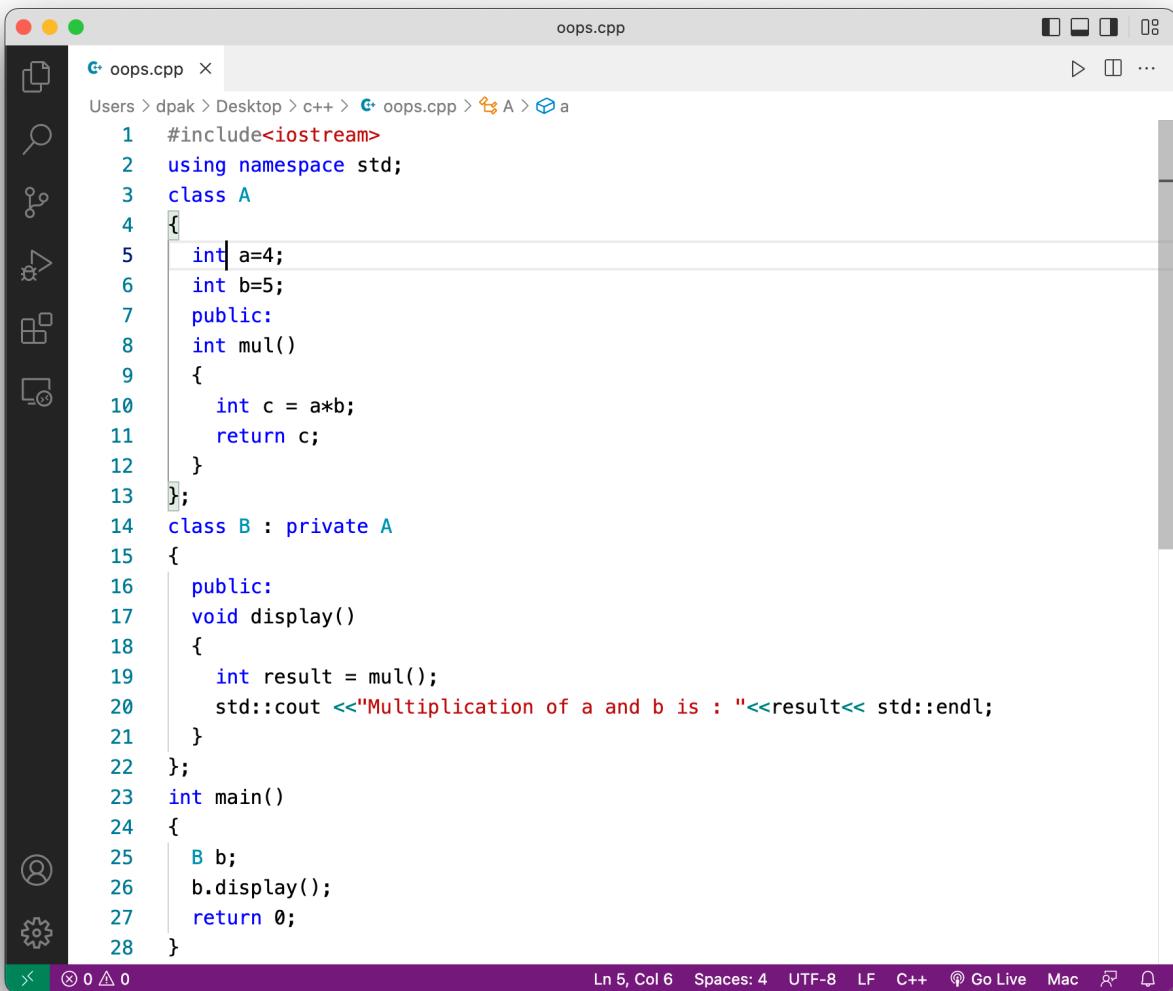
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
Child id is: 7
Parent id is: 91
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 17

Aim :- Write a program to print Multiplication of Two numbers using Single Inheritance.

Source Code :-



```
oops.cpp
1 #include<iostream>
2 using namespace std;
3 class A
4 {
5     int a=4;
6     int b=5;
7     public:
8     int mul()
9     {
10         int c = a*b;
11         return c;
12     }
13 };
14 class B : private A
15 {
16     public:
17     void display()
18     {
19         int result = mul();
20         std::cout <<"Multiplication of a and b is : "<<result<< std::endl;
21     }
22 };
23 int main()
24 {
25     B b;
26     b.display();
27     return 0;
28 }
```

Output:-

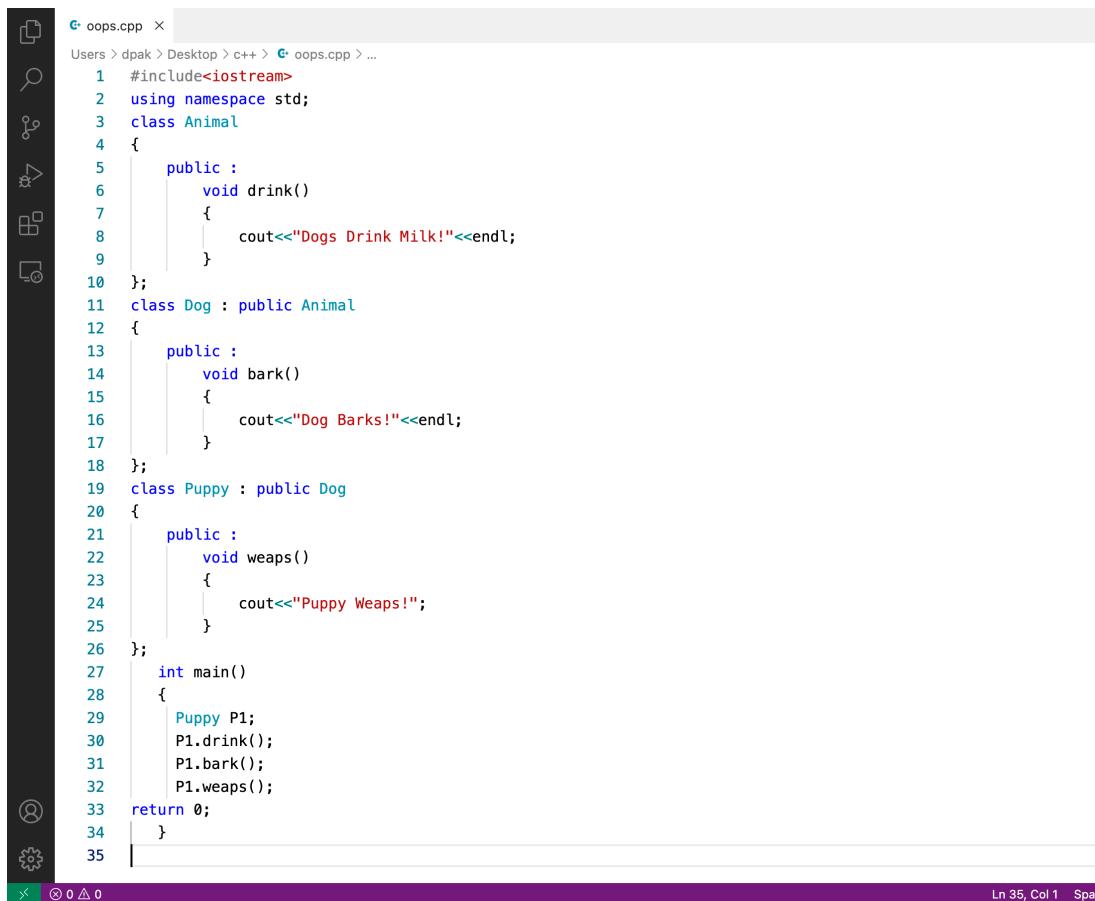
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    Code - c++ + ×
```

```
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
oops.cpp:5:8: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-extensio
ns]
int a=4;
^
oops.cpp:6:8: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-extensio
ns]
int b=5;
^
2 warnings generated.
Multiplication of a and b is : 20
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 18

Aim :- Write a program to use Multilevel Inheritance.

Source Code :-



```
oops.cpp x
Users > dpak > Desktop > c++ > oops.cpp > ...
1 #include<iostream>
2 using namespace std;
3 class Animal
4 {
5     public :
6         void drink()
7     {
8         cout<<"Dogs Drink Milk!"<<endl;
9     }
10 };
11 class Dog : public Animal
12 {
13     public :
14         void bark()
15     {
16         cout<<"Dog Barks!"<<endl;
17     }
18 };
19 class Puppy : public Dog
20 {
21     public :
22         void weaps()
23     {
24         cout<<"Puppy Weaps!";
25     }
26 };
27 int main()
28 {
29     Puppy P1;
30     P1.drink();
31     P1.bark();
32     P1.weaps();
33     return 0;
34 }
```

Output:-

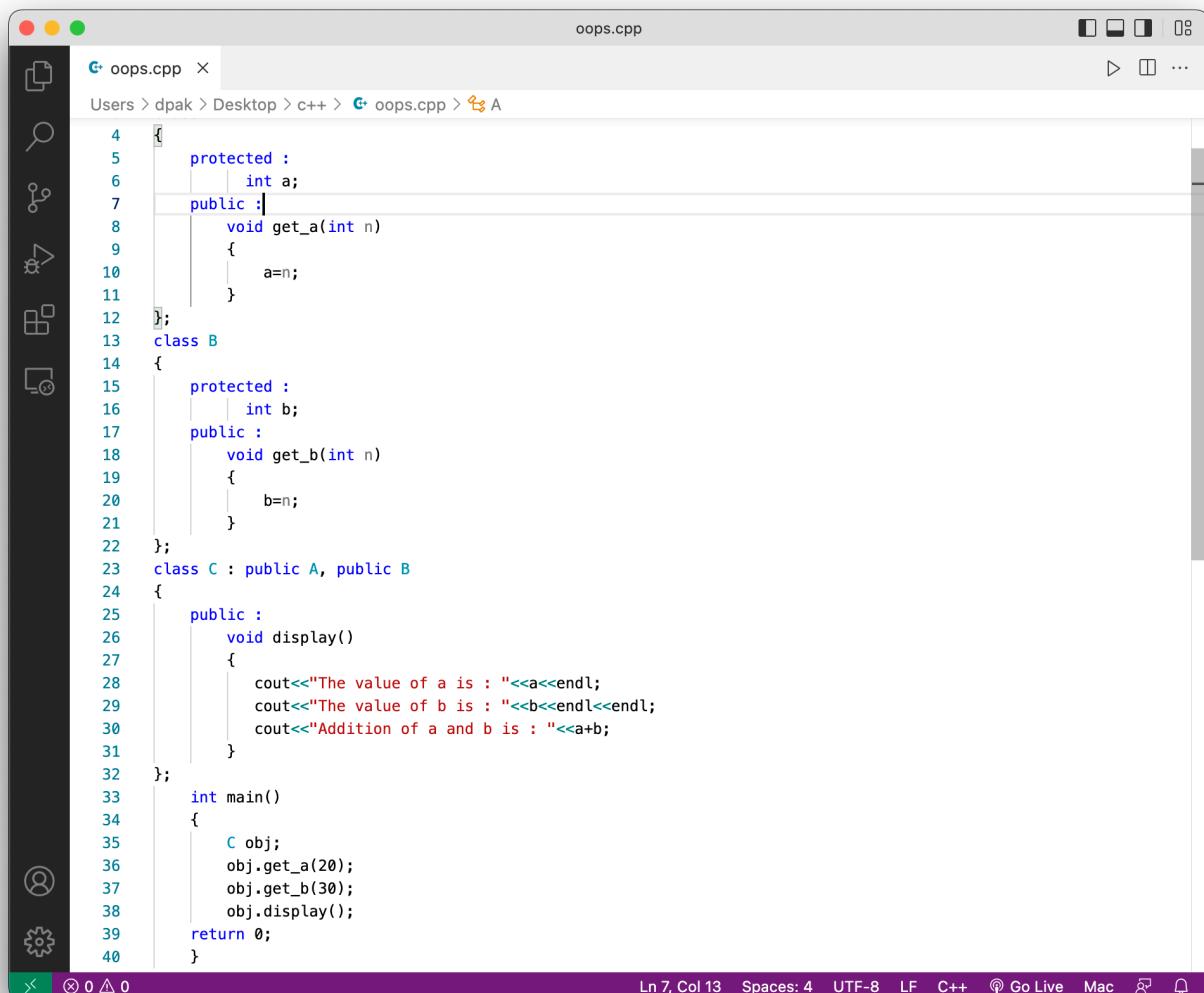
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/" oops
Dogs Drink Milk!
Dog Barks!
Puppy Weaps!
dpak@Deepaks-MacBook-Air c++ %

Experiment – 19

Aim :- Write a program to use Multiple Inheritance.

Source Code :-



The screenshot shows a code editor window titled "oops.cpp". The file contains C++ code demonstrating multiple inheritance. It defines three classes: A, B, and C. Class A has a protected member "a" and a public member function "get_a" that sets "a" to the value passed. Class B has a protected member "b" and a public member function "get_b" that sets "b" to the value passed. Class C inherits from both A and B ("public A, public B") and contains a public member function "display" which prints the values of "a" and "b" and their sum. The "main" function creates an object of class C, calls its member functions, and returns 0.

```
4  {
5      protected :
6          int a;
7      public :
8          void get_a(int n)
9          {
10             a=n;
11         }
12     };
13     class B
14     {
15         protected :
16             int b;
17         public :
18             void get_b(int n)
19             {
20                 b=n;
21             }
22     };
23     class C : public A, public B
24     {
25         public :
26             void display()
27             {
28                 cout<<"The value of a is : "<<a<<endl;
29                 cout<<"The value of b is : "<<b<<endl<<endl;
30                 cout<<"Addition of a and b is : "<<a+b;
31             }
32     };
33     int main()
34     {
35         C obj;
36         obj.get_a(20);
37         obj.get_b(30);
38         obj.display();
39         return 0;
40     }
```

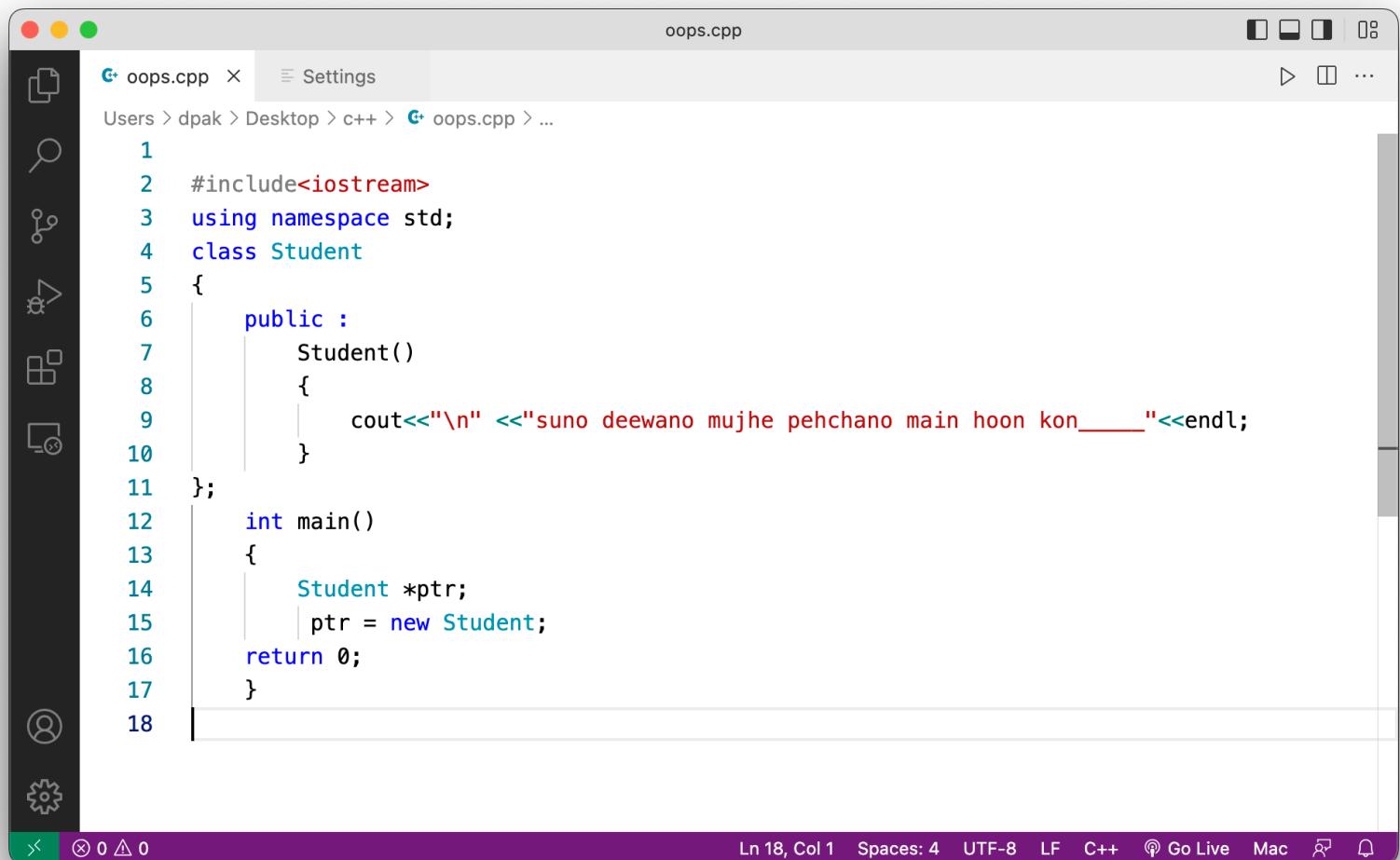
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
The value of a is : 20
The value of b is : 30
Addition of a and b is : 50
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 20

Aim :- Write a program to use New Operator.

Source Code :-



The screenshot shows a code editor window titled "oops.cpp". The file contains C++ code defining a class "Student" with a constructor and a main function. The code is as follows:

```
1  #include<iostream>
2  using namespace std;
3  class Student
4  {
5      public :
6          Student()
7          {
8              cout<<"\n" <<"suno deewano mujhe pehchano main hoon kon_____"<<endl;
9          }
10 };
11 int main()
12 {
13     Student *ptr;
14     ptr = new Student;
15     return 0;
16 }
```

The code editor interface includes a toolbar with icons for file operations, a sidebar with project navigation, and a status bar at the bottom showing line and column counts, encoding, and other settings.

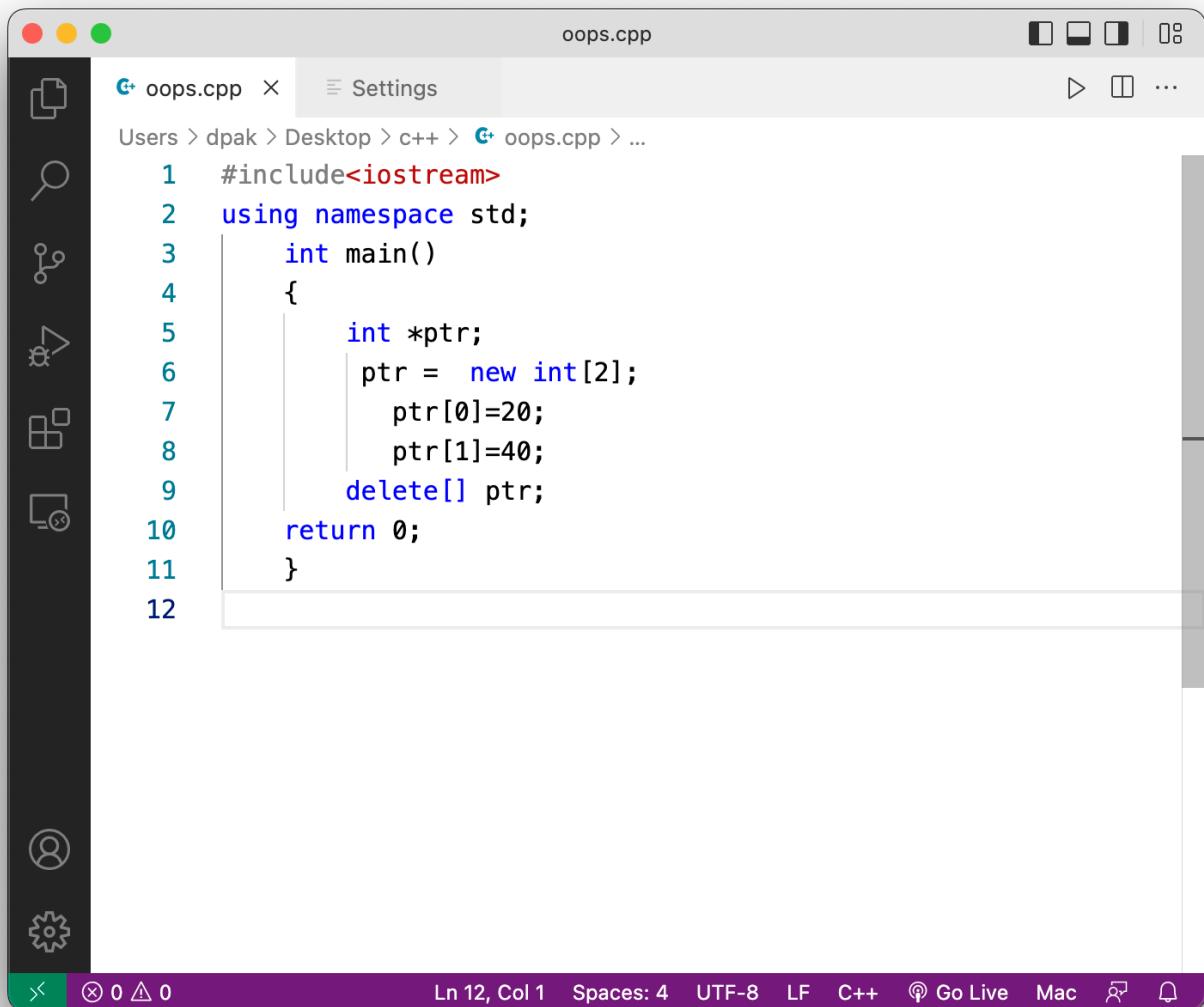
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code - dpak + × ☰ ^ ×
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
suno deewano mujhe pehchanoo main hoon kon_____
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 21

Aim :- Write a program to use Delete Operator.

Source Code :-



```
oops.cpp
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int *ptr;
6     ptr = new int[2];
7     ptr[0]=20;
8     ptr[1]=40;
9     delete[] ptr;
10    return 0;
11 }
12
```

Output:-

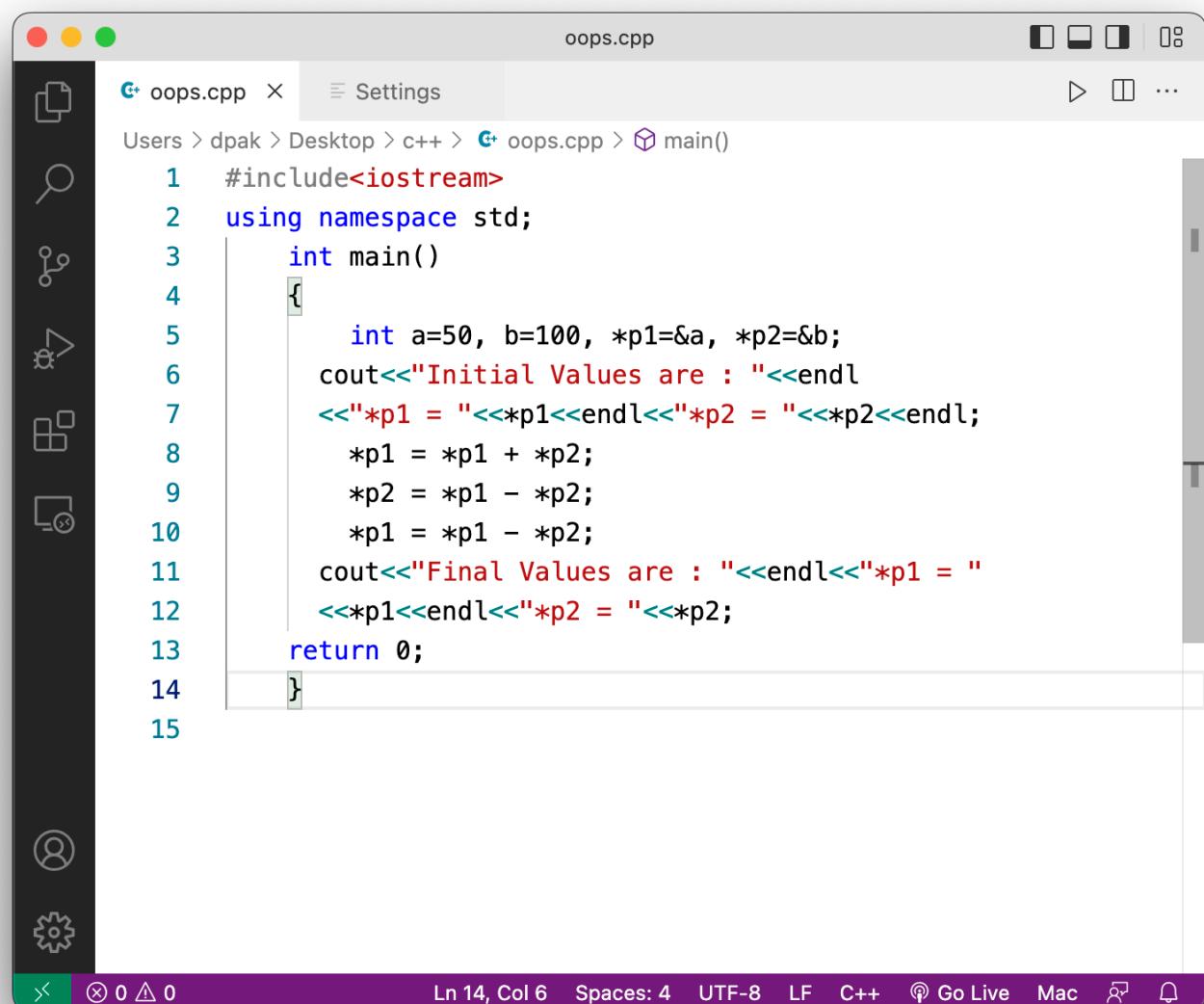


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops &&
"/Users/dpak/Desktop/c++/"oops
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 22

Aim :- Write a program to use Pointer.

Source Code :-



```
oops.cpp
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a=50, b=100, *p1=&a, *p2=&b;
6     cout<<"Initial Values are : "<<endl
7     <<*p1 = "<<*p1<<endl<<*p2 = "<<*p2<<endl;
8     *p1 = *p1 + *p2;
9     *p2 = *p1 - *p2;
10    *p1 = *p1 - *p2;
11    cout<<"Final Values are : "<<endl<<*p1 = "
12    <<*p1<<endl<<*p2 = "<<*p2;
13    return 0;
14 }
15
```

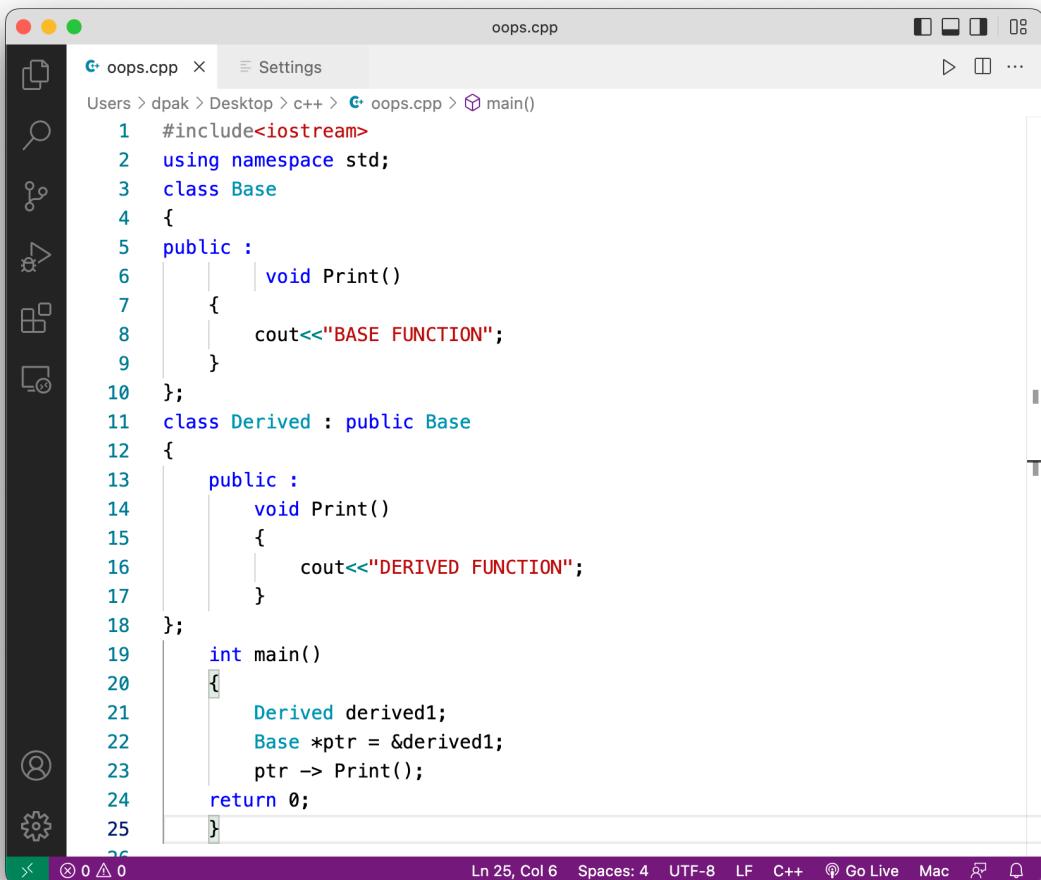
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL > Code - dpak + □ ━ ×  
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops &&  
"/Users/dpak/Desktop/c++/"oops  
Initial Values are :  
*p1 = 50  
*p2 = 100  
Final Values are :  
*p1 = 100  
*p2 = 50%  
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 23

Aim :- Write a program to call Overridden Function using Pointers.

Source Code :-



The screenshot shows a code editor window titled "oops.cpp". The file contains C++ code demonstrating function overriding. It includes declarations for a base class "Base" and a derived class "Derived", both with a "Print" method. The "Derived" class overrides the "Print" method. In the "main" function, a pointer of type "Base" is created to point to an object of type "Derived", and then the "Print" method is called via the pointer.

```
#include<iostream>
using namespace std;
class Base
{
public :
    void Print()
    {
        cout<<"BASE FUNCTION";
    }
};
class Derived : public Base
{
public :
    void Print()
    {
        cout<<"DERIVED FUNCTION";
    }
};
int main()
{
    Derived derived1;
    Base *ptr = &derived1;
    ptr->Print();
    return 0;
}
```

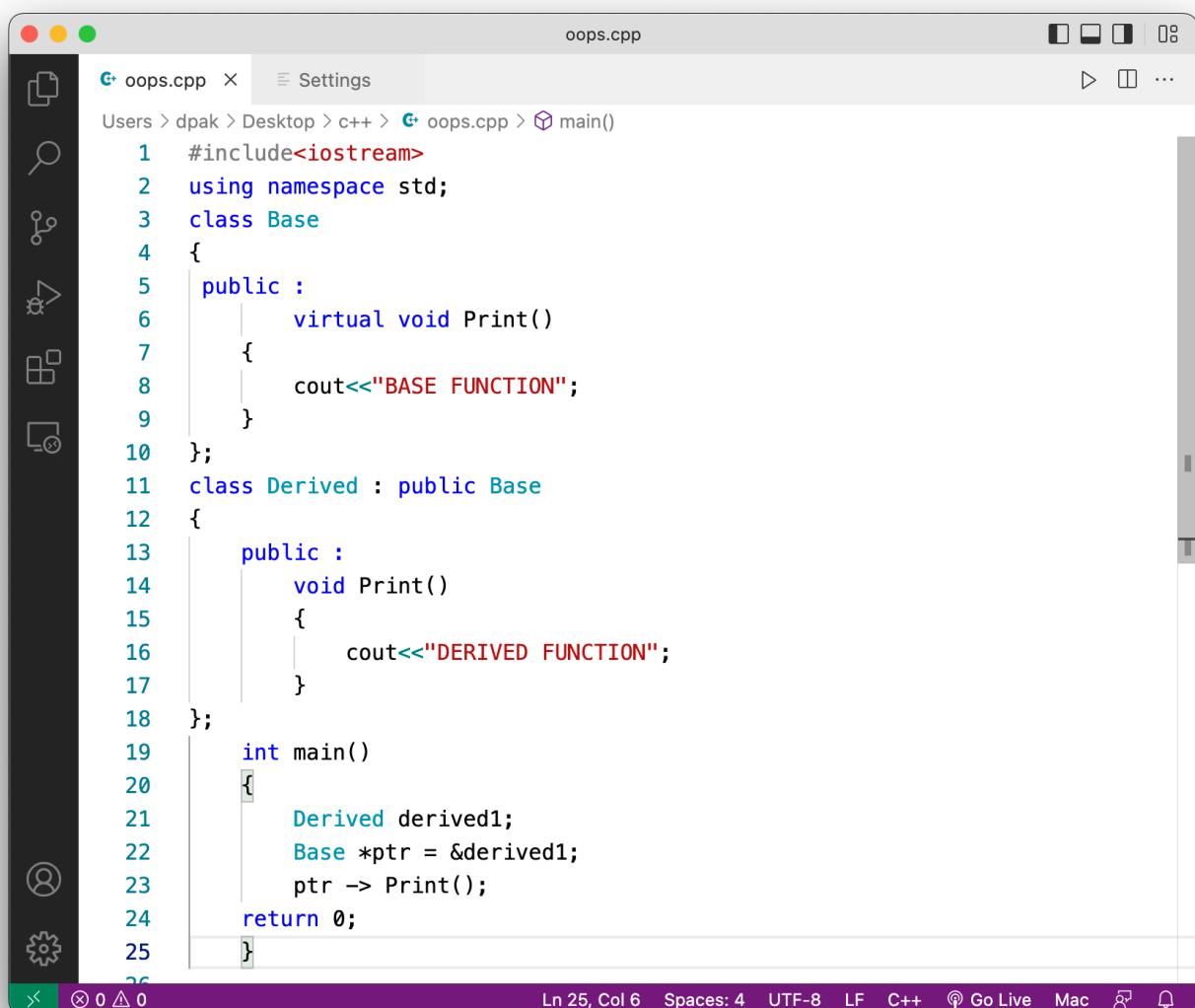
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops && "/Users/dpak/Desktop/c++/"oops
BASE FUNCTION
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 24

Aim :- Write a program to call Overridden Function using Pointers and Virtual Keyword.

Source Code :-



The screenshot shows a code editor window with the file 'oops.cpp' open. The code implements a base class 'Base' with a virtual function 'Print()', which outputs 'BASE FUNCTION'. It also defines a derived class 'Derived' that overrides 'Print()' to output 'DERIVED FUNCTION'. The main() function creates an object of 'Derived' and uses a pointer to call its overridden function. The code is as follows:

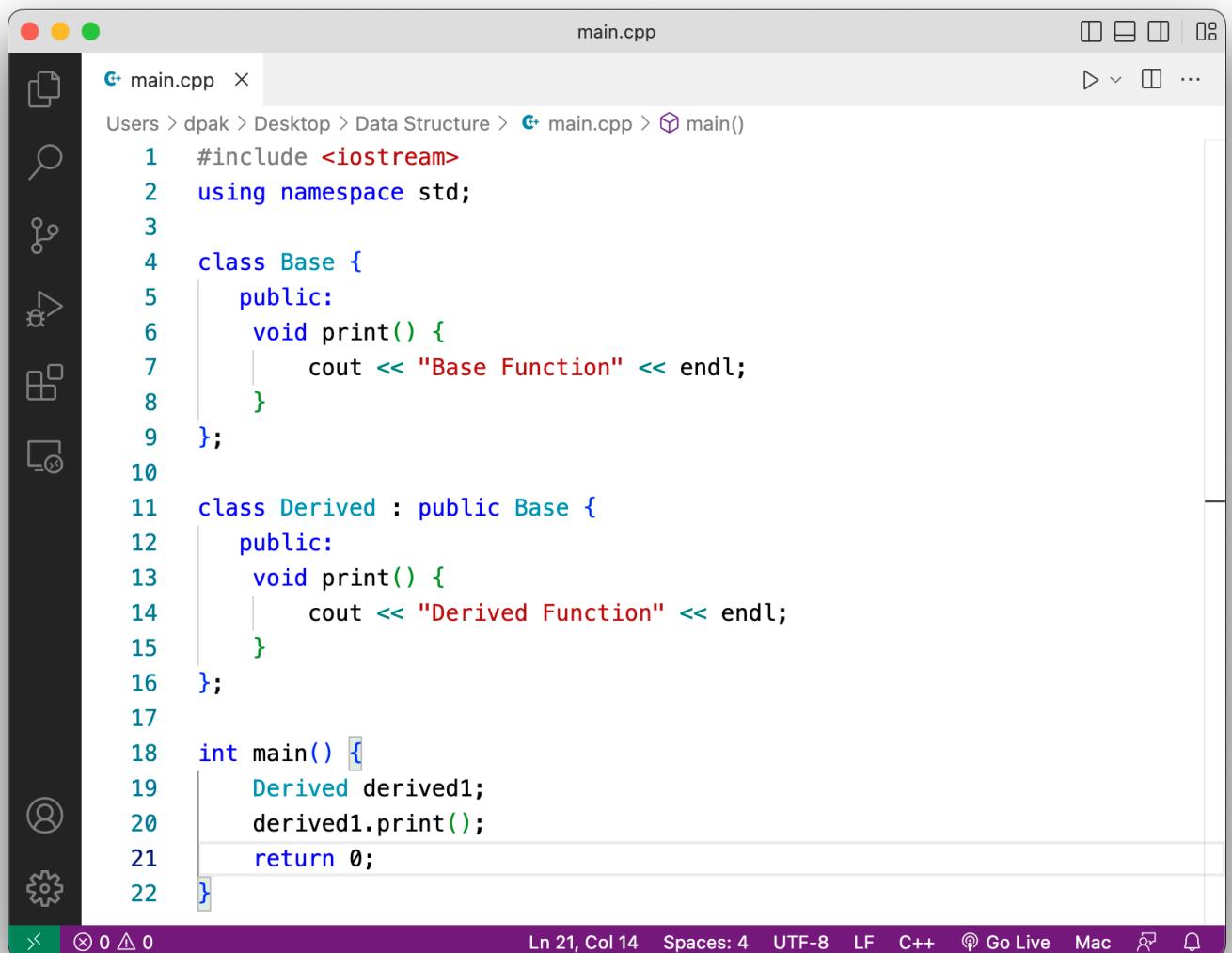
```
1 #include<iostream>
2 using namespace std;
3 class Base
4 {
5     public :
6         virtual void Print()
7     {
8         cout<<"BASE FUNCTION";
9     }
10 };
11 class Derived : public Base
12 {
13     public :
14         void Print()
15     {
16         cout<<"DERIVED FUNCTION";
17     }
18 };
19 int main()
20 {
21     Derived derived1;
22     Base *ptr = &derived1;
23     ptr -> Print();
24     return 0;
25 }
```

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ oops.cpp -o oops
&& "/Users/dpak/Desktop/c++/"oops
DERIVED FUNCTION
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 25

Aim :-Write a program to show function Overriding.



The screenshot shows a code editor window titled "main.cpp". The code implements function overriding:

```
main.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Base {
5 public:
6     void print() {
7         cout << "Base Function" << endl;
8     }
9 };
10
11 class Derived : public Base {
12 public:
13     void print() {
14         cout << "Derived Function" << endl;
15     }
16 };
17
18 int main() {
19     Derived derived1;
20     derived1.print();
21     return 0;
22 }
```

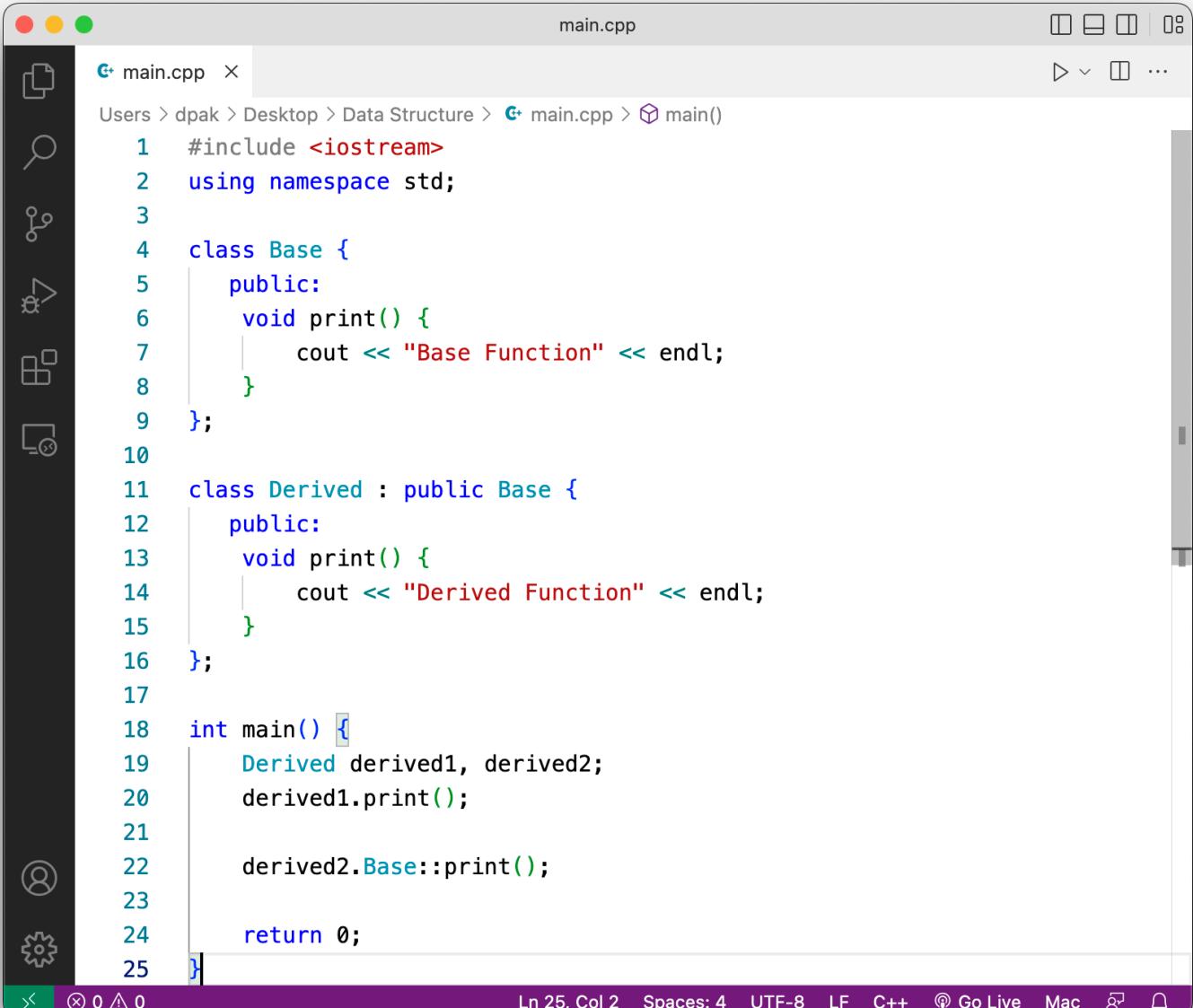
The code defines a base class "Base" with a "print()" method that outputs "Base Function". It then defines a derived class "Derived" that inherits from "Base" and overrides the "print()" method to output "Derived Function". In the "main()" function, an object of "Derived" is created and its "print()" method is called, demonstrating that the derived class's version of the function is executed.

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o
main && "/Users/dpak/Desktop/Data Structure/"main
Derived Function
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 26

Aim :- Write a program to Access Overridden Function to the Base Class.



The screenshot shows the VS Code interface with the following details:

- Title Bar:** main.cpp
- File Explorer:** Shows the file structure: Users > dpak > Desktop > Data Structure > main.cpp > main()
- Code Editor:** Displays the C++ code for main.cpp. The code defines a base class 'Base' with a 'print()' function that outputs "Base Function". It also defines a derived class 'Derived' that inherits from 'Base' and overrides the 'print()' function to output "Derived Function". The 'main()' function creates two objects, 'derived1' and 'derived2', and calls their respective 'print()' functions. The output of the 'derived1.print()' call is "Derived Function", and the output of the 'derived2.Base::print()' call is "Base Function".

```
1 #include <iostream>
2 using namespace std;
3
4 class Base {
5 public:
6     void print() {
7         cout << "Base Function" << endl;
8     }
9 };
10
11 class Derived : public Base {
12 public:
13     void print() {
14         cout << "Derived Function" << endl;
15     }
16 };
17
18 int main() {
19     Derived derived1, derived2;
20     derived1.print();
21
22     derived2.Base::print();
23
24     return 0;
25 }
```
- Bottom Status Bar:** Shows the line number (Ln 25), column (Col 2), spaces (Spaces: 4), encoding (UTF-8), line separator (LF), language (C++), and other status indicators like Go Live, Mac, and a refresh icon.

Output:-

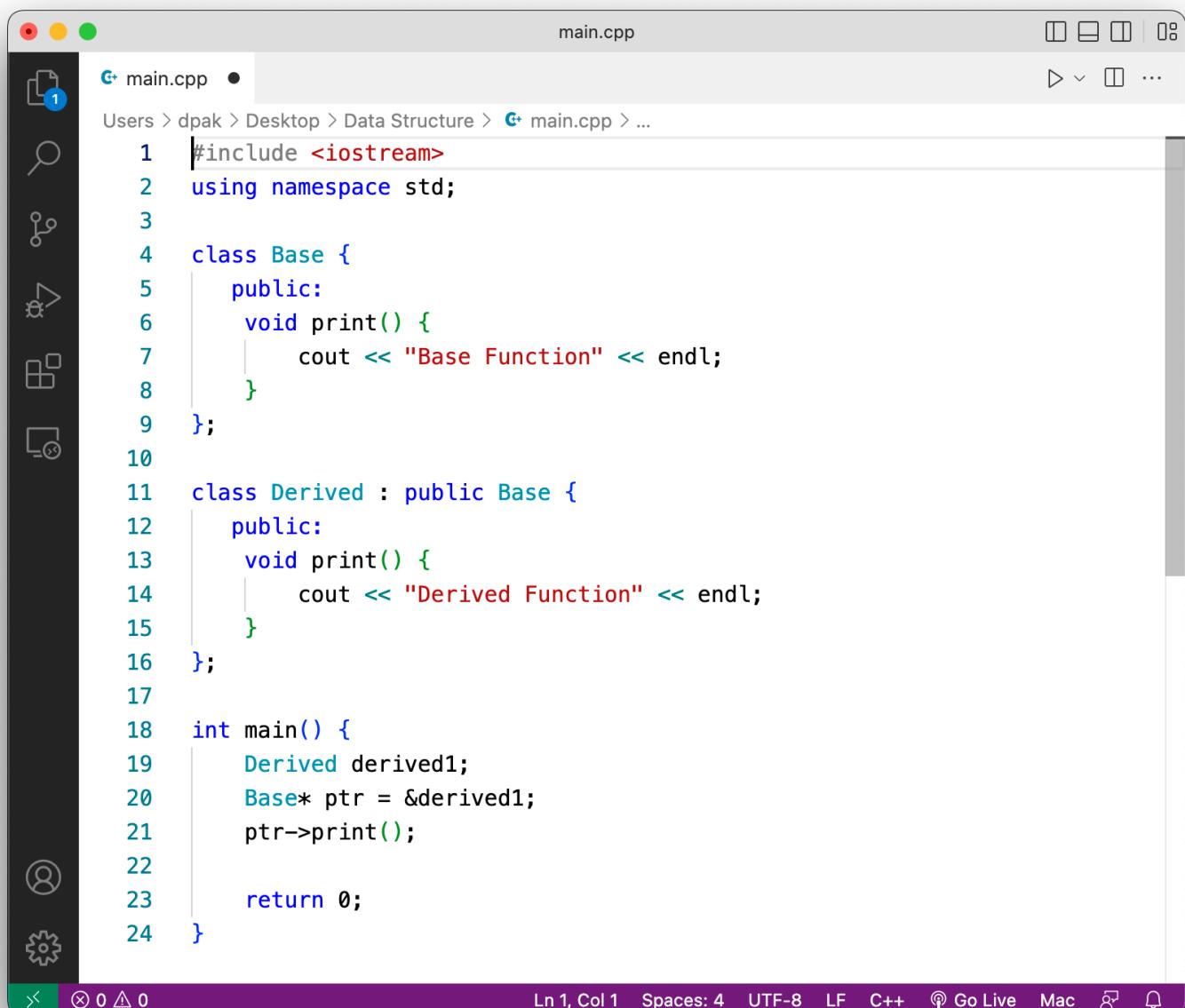
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - T + ▾ ▷ ⌂ ⌄ ⌁ ×

```
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o
main && "/Users/dpak/Desktop/Data Structure/"main
Derived Function
Base Function
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 27

Aim :- Write a program to call overridden function using pointer.



```
main.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Base {
5 public:
6     void print() {
7         cout << "Base Function" << endl;
8     }
9 };
10
11 class Derived : public Base {
12 public:
13     void print() {
14         cout << "Derived Function" << endl;
15     }
16 };
17
18 int main() {
19     Derived derived1;
20     Base* ptr = &derived1;
21     ptr->print();
22
23     return 0;
24 }
```

Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

> Code - T + v □ ✎ ^ ×

```
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main
Base Function
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 28

Aim :- Write a program to access private class using friend function.

The image shows a dark-themed IDE interface. On the left is a vertical toolbar with icons for file operations (New, Open, Save, Find, etc.). The main area has a title bar "main.cpp X" and a status bar at the bottom showing "0 0 0". The central part is a code editor with the following C++ code:

```
1 #include <iostream>
2 using namespace std;
3 class B;
4 class A{
5     int x;
6     public:
7     void assign(int i){
8         x=i;
9     }
10    friend void product(A,B);
11 };
12 class B
13 {
14     int y;
15     public:
16     void assign(int i)
17     {
18         y=i;
19     }
20     friend void product(A,B);
21 };
22 void product(A a,B b)
23 {
24     cout<<a.x * b.y;
25 }
26 int main(){
27     A a;
28     B b;
29     a.assign(45);
30     b.assign(20);
31     product(a,b);
32
33     return 0;
34 }
```

Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code - T +

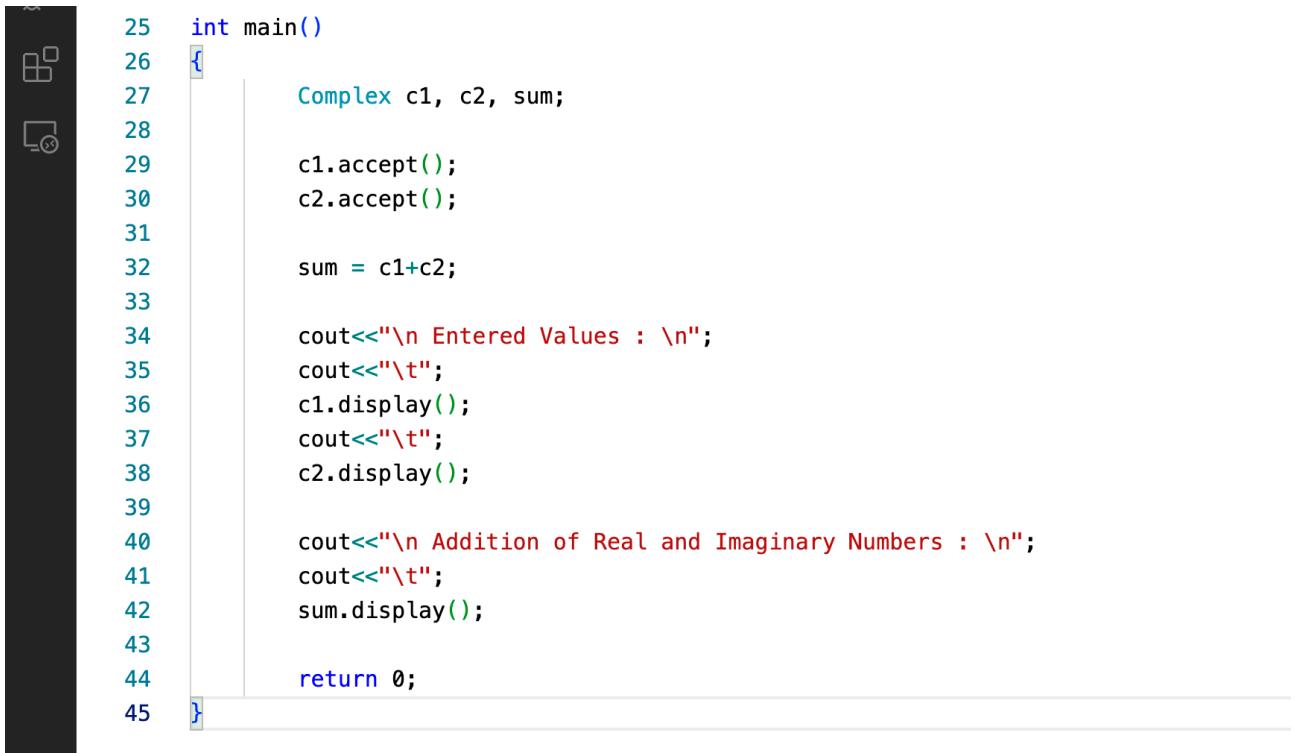
```
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main
900%
```

```
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 29

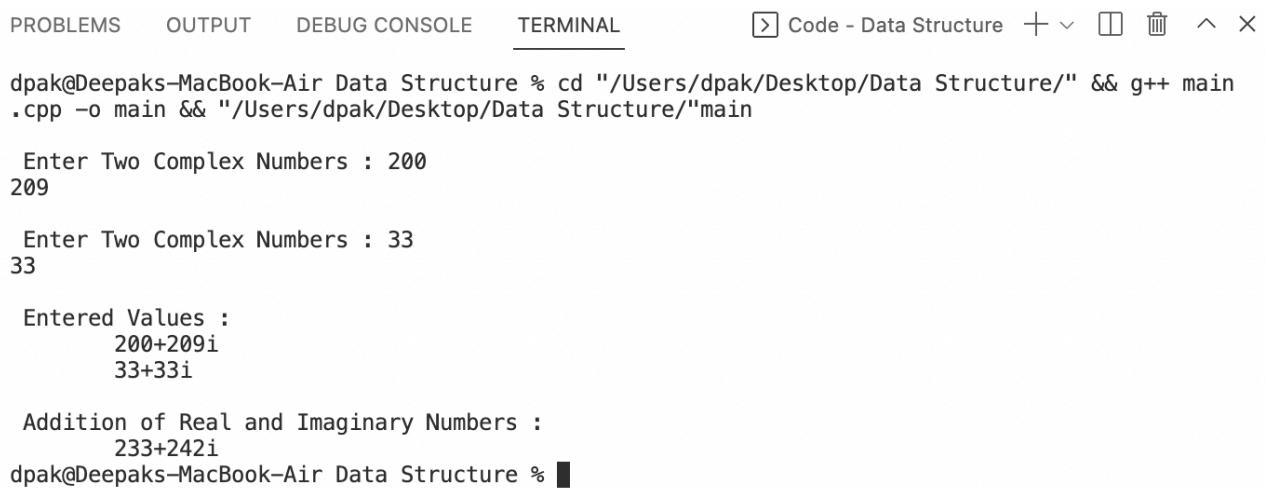
Aim :- Write a program to print sum of complex number using binary operator overloading.

```
main.cpp  X  Settings
Users > dpak > Desktop > Data Structure > main.cpp > main()
1 #include<iostream>
2 using namespace std;
3
4 class Complex
5 {
6     int num1, num2;
7 public:
8     void accept()
9     {
10         cout<<"\n Enter Two Complex Numbers : ";
11         cin>>num1>>num2;
12     }
13     Complex operator+(Complex obj)
14     {
15         Complex c;
16         c.num1=num1+obj.num1;
17         c.num2=num2+obj.num2;
18         return(c);
19     }
20     void display()
21     {
22         cout<<num1<<"+"<<num2<<"i"<<"\n";
23     }
24 };
```



```
25 int main()
26 {
27     Complex c1, c2, sum;
28
29     c1.accept();
30     c2.accept();
31
32     sum = c1+c2;
33
34     cout<<"\n Entered Values : \n";
35     cout<<"\t";
36     c1.display();
37     cout<<"\t";
38     c2.display();
39
40     cout<<"\n Addition of Real and Imaginary Numbers : \n";
41     cout<<"\t";
42     sum.display();
43
44     return 0;
45 }
```

Output:-

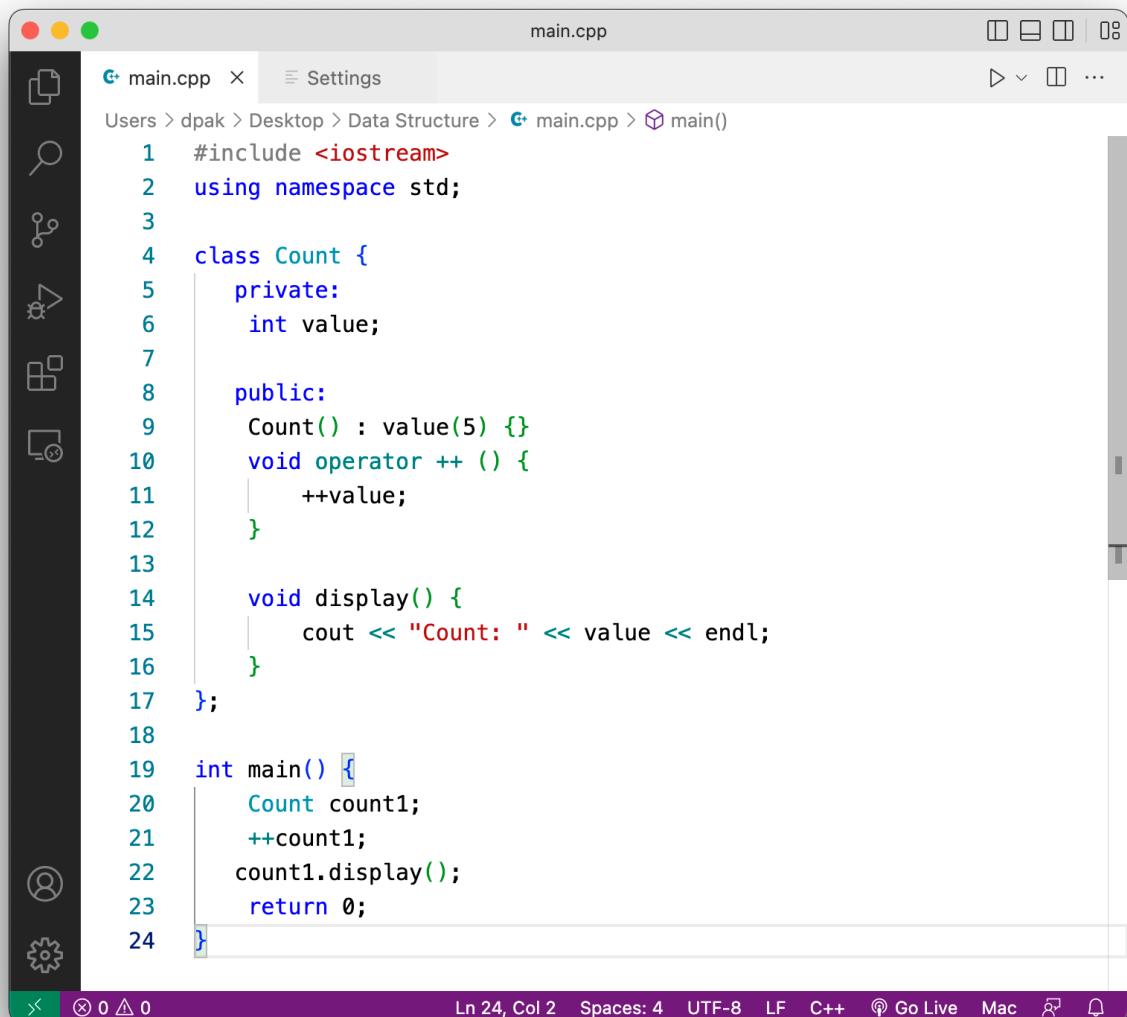


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code - Data Structure + ▾ File Terminal Close

```
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main
Enter Two Complex Numbers : 200
209
Enter Two Complex Numbers : 33
33
Entered Values :
200+209i
33+33i
Addition of Real and Imaginary Numbers :
233+242i
dpak@Deepaks-MacBook-Air Data Structure %
```

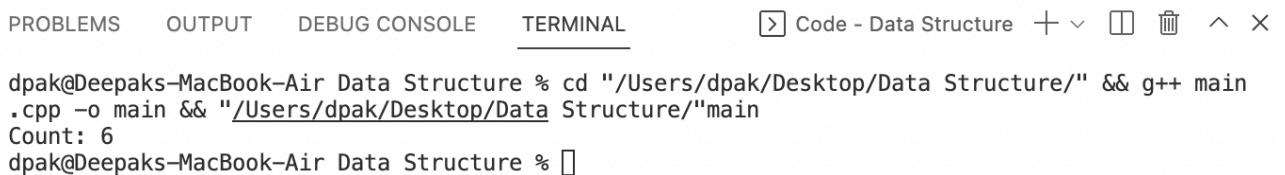
Experiment – 30

Aim :- Write a program to overload unary operator.



```
main.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Count {
5     private:
6         int value;
7
8     public:
9         Count() : value(5) {}
10        void operator ++ () {
11            ++value;
12        }
13
14        void display() {
15            cout << "Count: " << value << endl;
16        }
17    };
18
19 int main() {
20     Count count1;
21     ++count1;
22     count1.display();
23     return 0;
24 }
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code - Data Structure + ▾ □ ⌂ ⌄ ×
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main
Count: 6
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 31

Aim :- write a program to sum two complex numbers using binary operator overloading and friend function.

The screenshot shows a code editor window titled "main.cpp". The file path is "Users > dpak > Desktop > Data Structure > main.cpp". The code implements a class `complex` with methods `set()`, `display()`, and a friend function `sum()`. The `sum()` function adds two `complex` objects and returns their sum. The `display()` method prints the sum in the format "the sum of complex num is <<real<<" + i" <<imag;". The `main()` function demonstrates the usage of these functions by creating three `complex` objects, performing addition, and displaying the result.

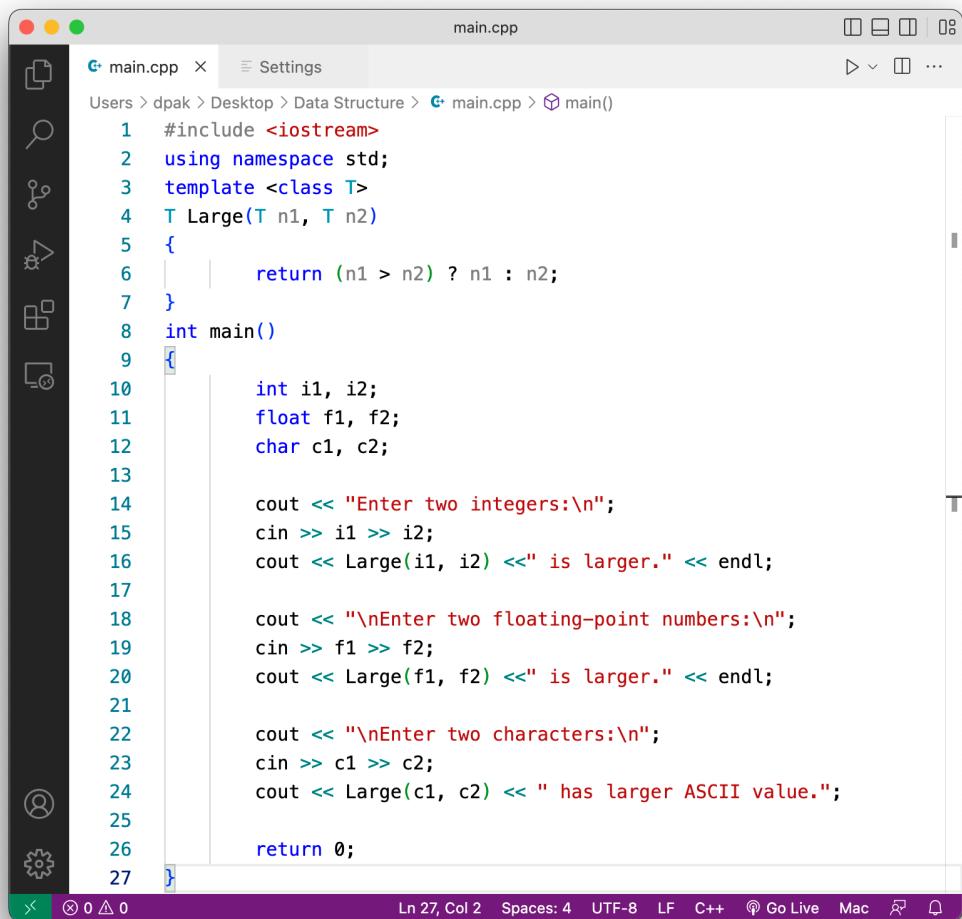
```
1 #include<iostream>
2 using namespace std;
3 class complex {
4 int real,imag;
5 public:
6 void set() {
7 cout<<"enter real and imag part";
8 cin>>real>>imag;
9 }
10 friend complex sum(complex,complex);
11 void display();
12 };
13 void complex::display() {
14 cout<<"the sum of complex num is"<<real<<"+"<<imag;
15 }
16 complex sum(complex a,complex b) {
17 complex t;
18 t.real=a.real+b.real;
19 t.imag=a.imag+b.imag;
20 return t;
21 }
22 int main() {
23 complex a,b,c;
24 a.set();
25 b.set();
26 c=sum(a,b);
27 c.display();
28 return(0);
29 }
```

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main
.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main
enter real and imag part22
22
enter real and imag part33
33
the sum of complex num is55+i55%
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 32

Aim :- write a program to sum two complex numbers using binary operator overloading and friend function.



```
main.cpp
1 #include <iostream>
2 using namespace std;
3 template <class T>
4 T Large(T n1, T n2)
5 {
6     return (n1 > n2) ? n1 : n2;
7 }
8 int main()
9 {
10     int i1, i2;
11     float f1, f2;
12     char c1, c2;
13
14     cout << "Enter two integers:\n";
15     cin >> i1 >> i2;
16     cout << Large(i1, i2) << " is larger." << endl;
17
18     cout << "\nEnter two floating-point numbers:\n";
19     cin >> f1 >> f2;
20     cout << Large(f1, f2) << " is larger." << endl;
21
22     cout << "\nEnter two characters:\n";
23     cin >> c1 >> c2;
24     cout << Large(c1, c2) << " has larger ASCII value.";
25
26
27 }
```

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code - Data Structure + × ☰ 🗑️ ⌂ ⌃ ⌄ dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main  
Enter two integers:  
22  
33  
33 is larger.  
  
Enter two floating-point numbers:  
33.4  
44.9  
44.9 is larger.  
  
Enter two characters:  
a  
b  
b has larger ASCII value. ☰ dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 33

Aim :- write a program to display a function template.

The screenshot shows a dark-themed IDE interface. The top bar displays the file name "main.cpp". The left sidebar contains various icons for file operations like Open, Save, Find, and Settings. The main area is a code editor with the following C++ code:

```
1 #include <iostream>
2 using namespace std;
3 template <typename T> T myMax(T x, T y)
4 {
5     return (x > y) ? x : y;
6 }
7 int main()
8 {
9     cout << myMax<int>(3, 7) << endl;
10    cout << myMax<double>(3.0, 7.0)
11        << endl;
12    cout << myMax<char>('g', 'e')
13        << endl;
14
15    return 0;
16 }
```

The code uses template functions and standard I/O streams to demonstrate function overloading and type deduction. The code editor has syntax highlighting for keywords, comments, and punctuation. The status bar at the bottom shows the current line (Ln 16), column (Col 2), and encoding (UTF-8). There are also tabs for "main.cpp" and "Settings".

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code - Data Structure + × ☰ 🗑 ⌂ ⌃ ⌄ dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main  
Enter two integers:  
22  
33  
33 is larger.  
  
Enter two floating-point numbers:  
33.4  
44.9  
44.9 is larger.  
  
Enter two characters:  
a  
b  
b has larger ASCII value. ☰ dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 34

Aim :- write a program to overload function template to display data

The screenshot shows a dark-themed IDE interface with a vertical toolbar on the left containing icons for file operations, search, and other development tools. The main window title is "main.cpp". The code editor displays the following C++ program:

```
1 #include <iostream>
2 using namespace std;
3 void square(int a)
4 {
5     cout << "Square of " << a
6     << " is " << a * a
7     << endl;
8 }
9 void square(double a)
10 {
11     cout << "Square of " << a
12     << " is " << a * a
13     << endl;
14 }
15 int main()
16 {
17     square(9);
18     square(2.25);
19     return 0;
20 }
```

The code defines two functions, `square`, one for integers and one for doubles, both printing the square of their argument. The `main` function calls these with arguments 9 and 2.25 respectively. The code editor shows syntax highlighting for keywords, comments, and operators.

Output:-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    Code - Data Structure    +    ▾    □    ⚡    ^    ×  
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp -o main && "/Users/dpak/Desktop/Data Structure/"main  
Square of 9 is 81  
Square of 2.25 is 5.0625  
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 35

Aim :- write a program to overload function template to display data using 3 different placeholder.

The screenshot shows a macOS desktop environment. In the foreground, a code editor window titled "main.cpp" is open, displaying C++ code. The code defines a template function "display" that outputs its argument to the console. It also contains two non-template functions that call the template function with different types. The "main" function at the bottom returns 0. The code editor has a dark theme with syntax highlighting. On the left, there's a sidebar with various icons. The top of the screen shows the window title "main.cpp" and the system menu. In the background, a terminal window is visible with some text output.

```
1 #include <iostream>
2 using namespace std;
3 template <class T>
4 void display(T t1)
5 {
6     cout << "Displaying Template: "
7         << t1 << "\n";
8 }
9 void display(int t1)
10 {
11     cout << "Explicitly display: "
12         << t1 << "\n";
13 }
14 int main()
15 {
16     display(200);
17     display(12.40);
18     display('G');
19
20     return 0;
21 }
```

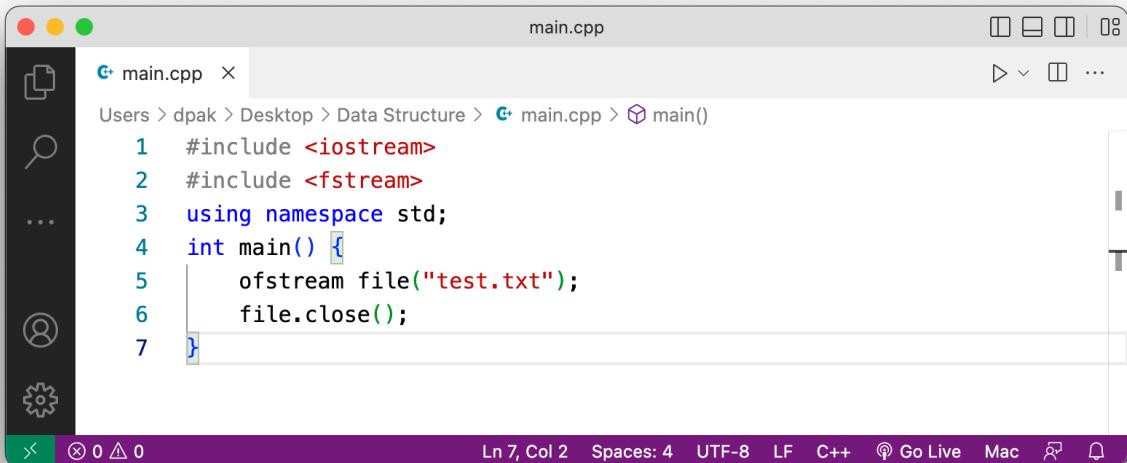
Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
dpak@Deepaks-MacBook-Air Data Structure % cd "/Users/dpak/Desktop/Data Structure/" && g++ main.cpp
p -o main && "/Users/dpak/Desktop/Data Structure/"main
Explicitly display: 200
Displaying Template: 12.4
Displaying Template: G
dpak@Deepaks-MacBook-Air Data Structure %
```

Experiment – 36

Aim :- write a program to Create a File.



```
main.cpp
```

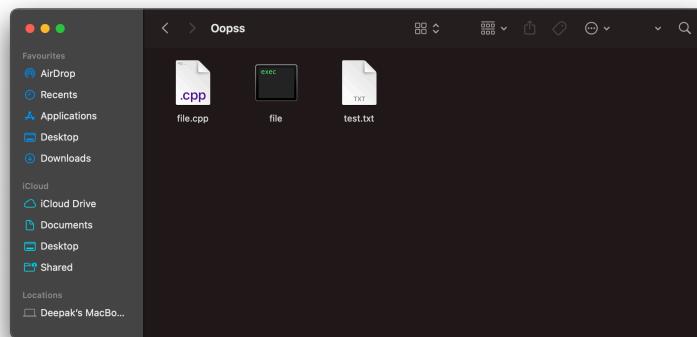
```
main.cpp x
```

```
Users > dpak > Desktop > Data Structure > main.cpp > main()
```

```
1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4 int main() {
5     ofstream file("test.txt");
6     file.close();
7 }
```

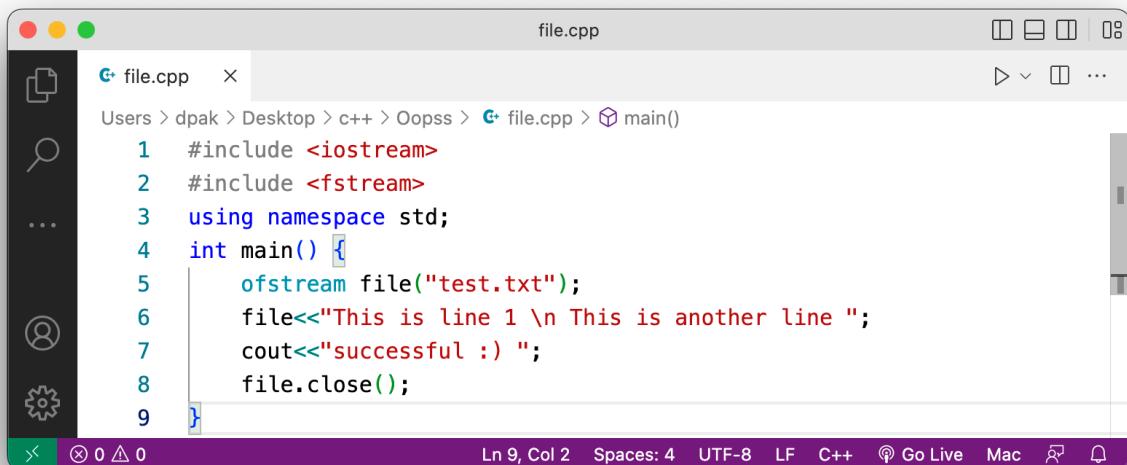
```
Ln 7, Col 2 Spaces: 4 UTF-8 LF C++ Go Live Mac ⌂ ⌂
```

Output:-



Experiment – 37

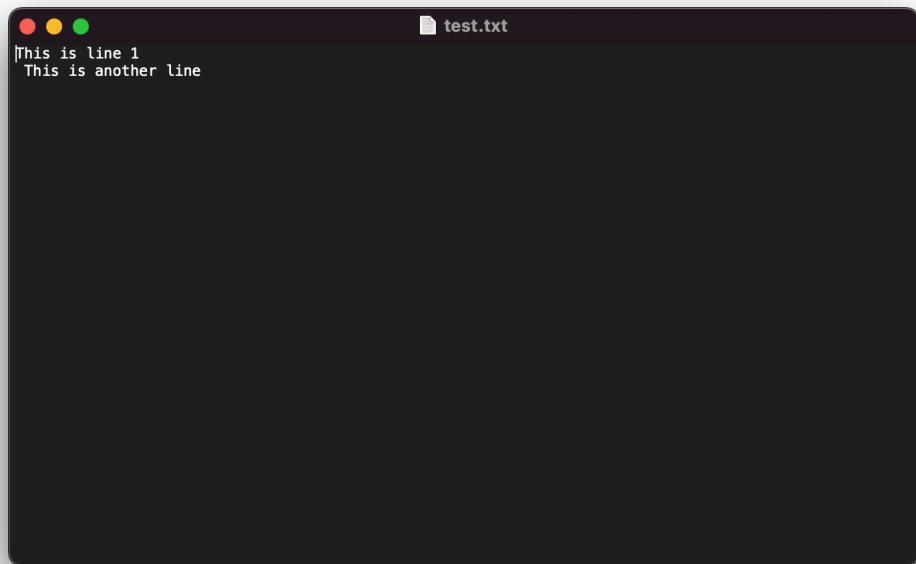
Aim :- write a program to write text in file.



The screenshot shows a code editor window titled "file.cpp". The code is as follows:

```
1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4 int main() {
5     ofstream file("test.txt");
6     file<<"This is line 1 \n This is another line ";
7     cout<<"successful :)";
8     file.close();
9 }
```

The status bar at the bottom indicates "Ln 9, Col 2" and "Spaces: 4".



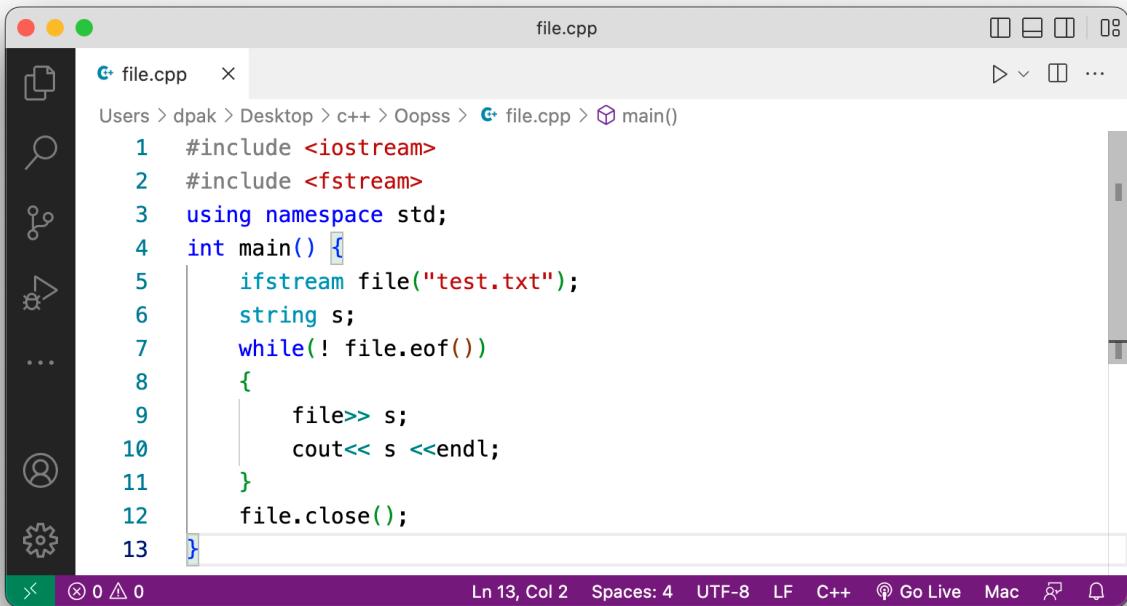
Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code - Data Structure +

```
dpak@Deepaks-MacBook-Air Oopss % cd "/Users/dpak/Desktop/c++/Oopss/" && g++ file.cpp -o file && "
/Users/dpak/Desktop/c++/Oopss/"file
successful :)
dpak@Deepaks-MacBook-Air Oopss %
```

Experiment – 38

Aim :- write a program to Read text file.



```
file.cpp
```

```
file.cpp  x
Users > dpak > Desktop > c++ > Oopss > file.cpp > main()
1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4 int main() {
5     ifstream file("test.txt");
6     string s;
7     while(! file.eof())
8     {
9         file>> s;
10        cout<< s << endl;
11    }
12    file.close();
13 }
```

Ln 13, Col 2 Spaces: 4 UTF-8 LF C++ Go Live Mac ⌂ ⌂

Output:-

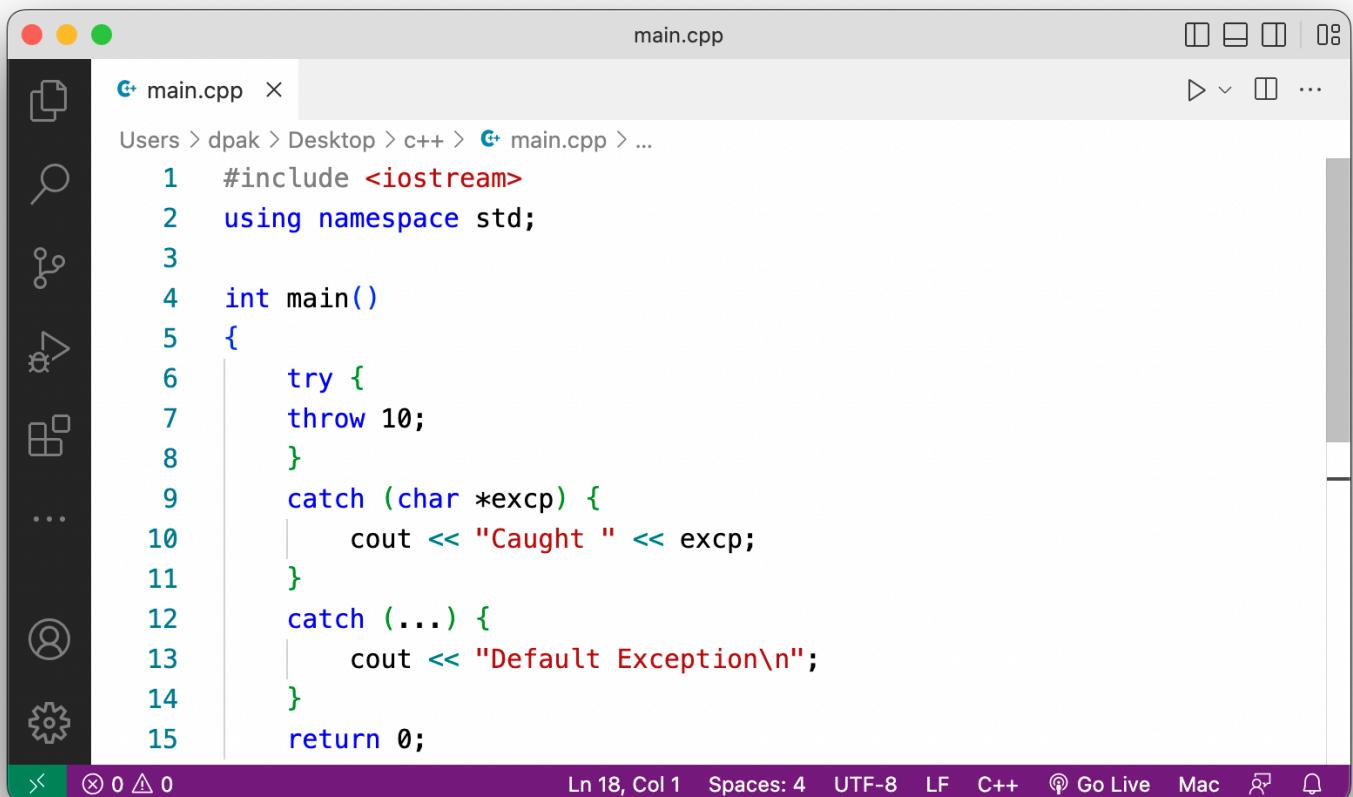


```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   Code - Data Structure + ▾ ⌂ ⌂ ⌂ ×
```

```
dpak@Deepaks-MacBook-Air Oopss % cd "/Users/dpak/Desktop/c++/Oopss/" && g++ file.cpp -o file && "
/Users/dpak/Desktop/c++/Oopss/"file
This
is
line
1
This
is
another
line
line
dpak@Deepaks-MacBook-Air Oopss %
```

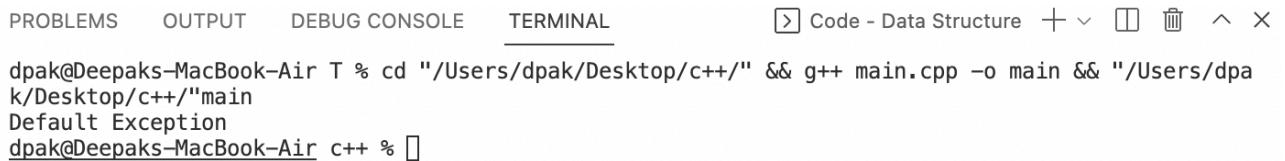
Experiment – 39

Aim :- write a program to .



```
main.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     try {
7         throw 10;
8     }
9     catch (char *excp) {
10         cout << "Caught " << excp;
11     }
12     catch (...) {
13         cout << "Default Exception\n";
14     }
15 }
```

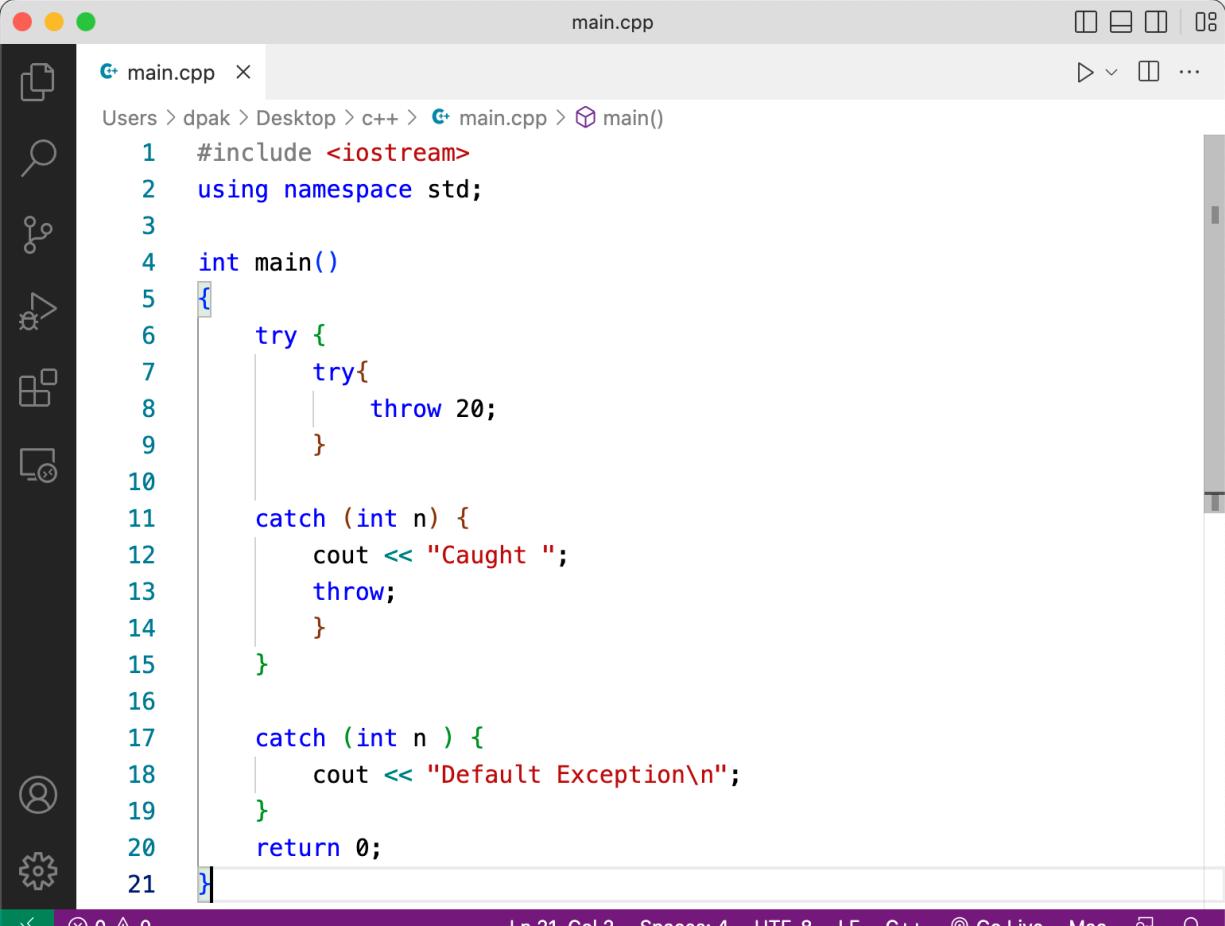
Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code - Data Structure + ▾ □ ×
dpak@Deepaks-MacBook-Air ~ % cd "/Users/dpak/Desktop/c++/" && g++ main.cpp -o main && "/Users/dpak/Desktop/c++/"main
Default Exception
dpak@Deepaks-MacBook-Air ~ %
```

Experiment – 40

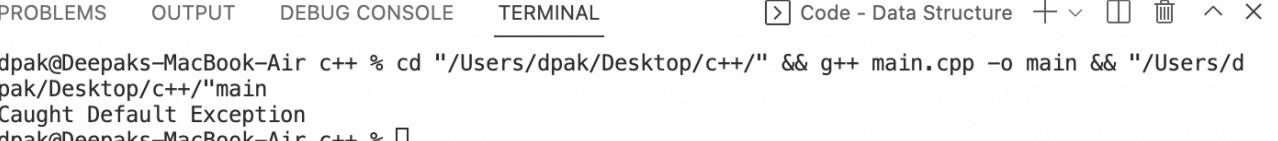
Aim :- write a program to .



```
main.cpp
Users > dpak > Desktop > c++ > main.cpp > main()
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     try {
7         try{
8             throw 20;
9         }
10
11     catch (int n) {
12         cout << "Caught ";
13         throw;
14     }
15
16
17     catch (int n ) {
18         cout << "Default Exception\n";
19     }
20
21 }
```

x ① △ 0 Ln 21, Col 2 Spaces: 4 UTF-8 LF C++ ⚡ Go Live Mac ⚡

Output:-

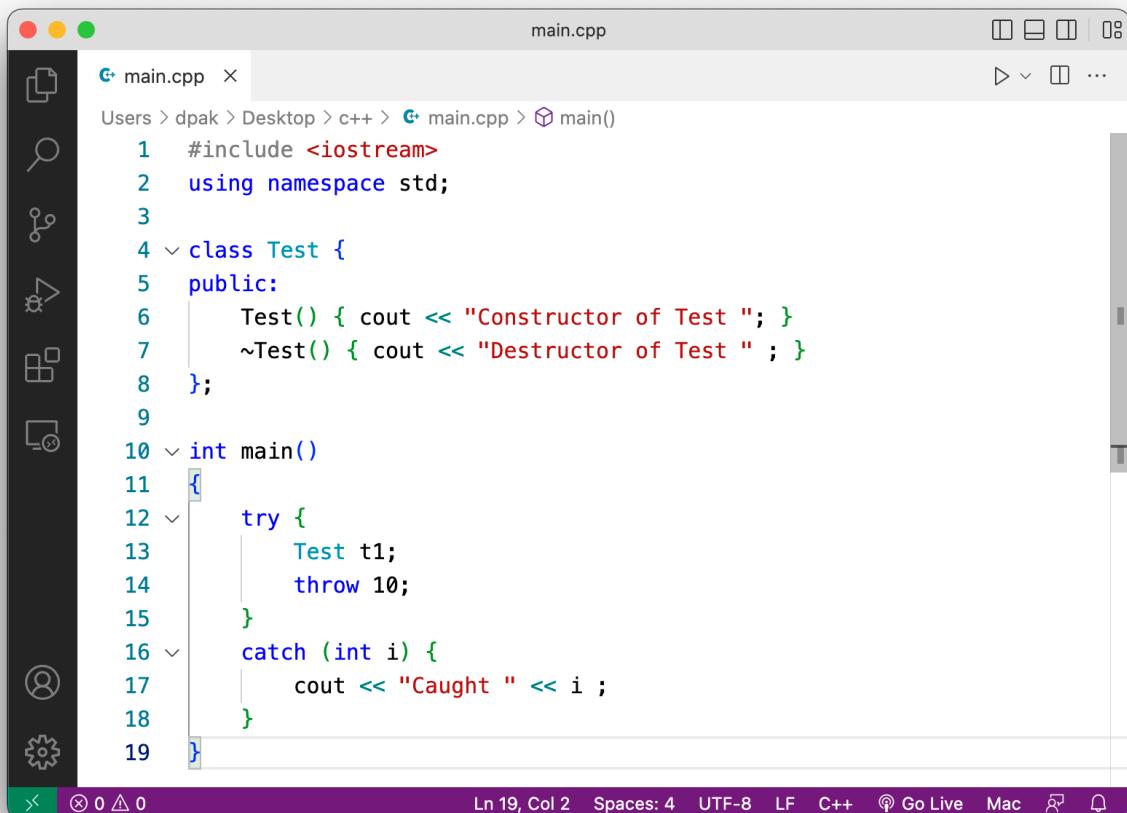


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ⚡ Code - Data Structure + ×

```
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ main.cpp -o main && "/Users/dpak/Desktop/c++/"main
Caught Default Exception
dpak@Deepaks-MacBook-Air c++ %
```

Experiment – 41

Aim :- write a program to .



The screenshot shows a code editor window titled "main.cpp". The code implements a simple exception handling mechanism. It defines a class "Test" with a constructor and a destructor. In the "main" function, a try block creates an instance of "Test" and throws an integer value of 10. A catch block then prints the caught value to the console. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 class Test {
5 public:
6     Test() { cout << "Constructor of Test " ; }
7     ~Test() { cout << "Destructor of Test " ; }
8 };
9
10 int main()
11 {
12     try {
13         Test t1;
14         throw 10;
15     }
16     catch (int i) {
17         cout << "Caught " << i ;
18     }
19 }
```

The status bar at the bottom indicates the code is 19 lines long, column 2, with 4 spaces, using UTF-8 encoding, and is a C++ file.

Output:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code - Data Structure + ✖ ✖ ✖ ✖

```
dpak@Deepaks-MacBook-Air c++ % cd "/Users/dpak/Desktop/c++/" && g++ main.cpp -o main && "/Users/dpak/Desktop/c++/"main
Constructor of Test Destructor of Test Caught 10%
dpak@Deepaks-MacBook-Air c++ %
```

