

Object Oriented
Methodologies Lab
ETCT 257

SUBMITTED TO:

MS. ARUSHI JAIN

ASSISTANT PROFESSOR

SUBMITTED BY:

KHUSHI JHA

ITE2



Maharaja Agrasen Institute of Technology, PSP
Area, Sector – 22, Rohini, New Delhi – 110085

S. No.	Lab Session	Program	Page Number	Date	Teacher's Signature
1	1	Write a program to add and subtract two complex numbers	4	22/09/21	
2		Write a program to add, subtract and multiply two matrices.	7		
3		Write a program to find greatest of three numbers.	13		
4		Write a program to find out square of given numbers of different data types.	17		
5	2	Write a program to read class student info such as name, age, gender, height and weight from the keyboard and to store them on a specified file using read() and write() functions. Again, the same file is opened for reading and displaying the contents of the file on the screen.	18	29/09/21	
6	3	Write a program to print the area and perimeter of a triangle, having sides of 3, 4 and 5 units by creating a class named 'Triangle', with a function to print the area and perimeter.	22	06/10/21	
7		Write a C++ program to compare two strings using overloading.	24		
8		Write a C++ program to find the factorial of a number using class and function declared outside the class.	28		
9	4	Implement the class string containing the following functions: <ul style="list-style-type: none"> • Overload + operator to carry out the concatenation of strings • Overload = operator to carry out string copy • Overload <= operator to carry out comparison of strings • Function to display the length of the string • Function tolower() to convert uppercase letter to lowercase 	30	13/10/21	

		<ul style="list-style-type: none"> Function toupper() to convert lowercase letter to uppercase 			
10		Create a class called LIST with functions store() and retrieve(). To store a value, call store function and to retrieve a value, call retrieve function. Derive two classes stack and queue from it and override store and retrieve.	35		
11	5	Write a program for multiplication of two matrices using OOP.	43	27/10/21	
12		Write a program to perform addition of two complex numbers using OOP.	47		

LAB1

1. Write a program to add and subtract two complex numbers.

```
2.
3. #include <bits/stdc++.h>
4. using namespace std;
5.
6. struct complexNumber
7. {
8.     int real;
9.     int imag;
10. };
11.
12. void create(struct complexNumber *C, int a, int b)
13. {
14.     C->real = a;
15.     C->imag = b;
16.     cout << "Complex Number " << C->real << " + " << C-
    >imag << "i created." << endl;
17. }
18.
19. void add(struct complexNumber *C1, struct complexNumber
    *C2, struct complexNumber *res)
20. {
21.     res->real = C1->real + C2->real;
22.     res->imag = C1->imag + C2->imag;
23.     cout << "Addition of two complex Number created = "
    << res->real << " + " << res->imag << "i" << endl;
24. }
25.
26. void subtract(struct complexNumber *C1, struct
    complexNumber *C2, struct complexNumber *res)
27. {
28.     res->real = C1->real - C2->real;
29.     res->imag = C1->imag - C2->imag;
30.     cout << "Subtraction of two complex Number created =
    " << res->real << " - " << res->imag << "i" << endl;
31. }
32.
33. int main()
34. {
35.     struct complexNumber a, b, res;
```

```
36.         create(&a, 2, 7);
37.         create(&b, 2, 2);
38.         add(&a, &b, &res);
39.         subtract(&a, &b, &res);
40.         return 0;
41.     }
42.
```

OUTPUT:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\DELL> cd "g:\my_codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Complex Number 2 + 7i created.
Complex Number 2 + 2i created.
Addition of two complex Number created = 4 + 9i
Subtraction of two complex Number created = 0 - 5i
PS G:\my_codes>
```

2. Write a program to add, subtract and multiply two matrices.

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int N;
    cout << "Enter rows and columns of both matrices : ";
    cin >> N;
    int a1[N][N];
    int a2[N][N];
    cout << "Enter elements of first matrix : ";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            cin >> a1[i][j];
        }
    }
    cout << "Enter elements of second matrix : ";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            cin >> a2[i][j];
        }
    }
    cout << "First matrix :\n";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            cout << a1[i][j] << " ";
        }
        cout << endl;
    }
    cout << "Second matrix :\n";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
```

```

        {
            cout << a2[i][j] << " ";
        }
        cout << endl;
    }
    int sum[N][N];
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            sum[i][j] = a1[i][j] + a2[i][j];
        }
    }
    cout << "Resultant Matrix after Addition : \n";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            cout << sum[i][j] << " ";
        }
        cout << endl;
    }
    int sub[N][N];
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            sub[i][j] = a1[i][j] - a2[i][j];
        }
    }
    cout << "Resultant Matrix after Subtraction : \n";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            cout << sub[i][j] << " ";
        }
        cout << endl;
    }
    int ans[N][N];
    for (int i = 0; i < N; i++)
    {

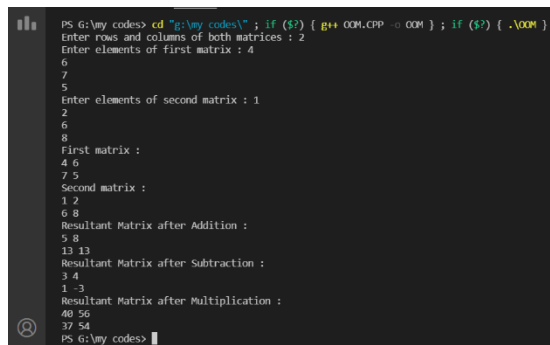
```

```

        for (int j = 0; j < N; j++)
        {
            ans[i][j] = 0;
        }
    }
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            for (int k = 0; k < N; k++)
            {
                ans[i][j] += a1[i][k] * a2[k][j];
            }
        }
    }
    cout << "Resultant Matrix after Multiplication : \n";
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            cout << ans[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```

OUTPUT:



```

PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Enter rows and columns of both matrices : 2
Enter elements of first matrix : 4
6
7
5
Enter elements of second matrix : 1
2
6
8
First matrix :
4 6
7 5
Second matrix :
1 2
6 8
Resultant Matrix after Addition :
5 8
13 13
Resultant Matrix after Subtraction :
3 4
1 -3
Resultant Matrix after Multiplication :
40 56
37 54
PS G:\my codes>

```


3. Write a program to find greatest of three numbers.

```
4. #include <bits/stdc++.h>
5. using namespace std;
6.
7. void greatestOfThreeNumbers(int a, int b, int c)
8. {
9.     if (a >= b && a >= c)
10.    {
11.        if (a > b && a > c)
12.        {
13.            cout << "Greatest of three numbers is " << a
14.            << endl;
15.        }
16.        else
17.        {
18.            if (a == b && a > c)
19.            {
20.                cout << "A and B both are greatest
21.                numbers which is equal to " << a << endl;
22.            }
23.            else if (a == b && a == c)
24.            {
25.                cout << "A , B and C all are equal to
26.                each other and their value is " << a << endl;
27.            }
28.            else
29.            {
30.                cout << "A and C both are greatest
31.                numbers which is equal to " << c << endl;
32.            }
33.        }
34.    }
35.    else if (b >= a && b >= c)
36.    {
37.        if (b > c && b > a)
38.        {
39.            cout << "Greatest of three numbers is " << b;
40.        }
41.        else
42.        {
43.            if (b == a && b > c)
44.            {
45.                cout << "Greatest of three numbers is " << b;
46.            }
47.            else if (b == a && b == c)
48.            {
49.                cout << "A , B and C all are equal to " << b << endl;
50.            }
51.        }
52.    }
53. }
```

```
41.             cout << "A and B both are greatest
               numbers which is equal to " << a << endl;
42.             }
43.             else
44.             {
45.                 cout << "B and C both are greatest
               numbers which is equal to " << c << endl;
46.             }
47.         }
48.     }
49.     else
50.     {
51.         if (c > a && c > b)
52.         {
53.             cout << "Greatest of three numbers is " << c
               << endl;
54.         }
55.         else
56.         {
57.             if (c == a && c > b)
58.             {
59.                 cout << "A and C both are greatest
               numbers which is equal to " << c << endl;
60.             }
61.             else
62.             {
63.                 cout << "B and C both are greatest
               numbers which is equal to " << c << endl;
64.             }
65.         }
66.     }
67. }
68.
69. int main()
70. {
71.     int a, b, c;
72.     cout << "Enter three numbers : ";
73.     cin >> a >> b >> c;
74.     greatestOfThreeNumbers(a, b, c);
75.     return 0;
76. }
77.
```

OUTPUT:



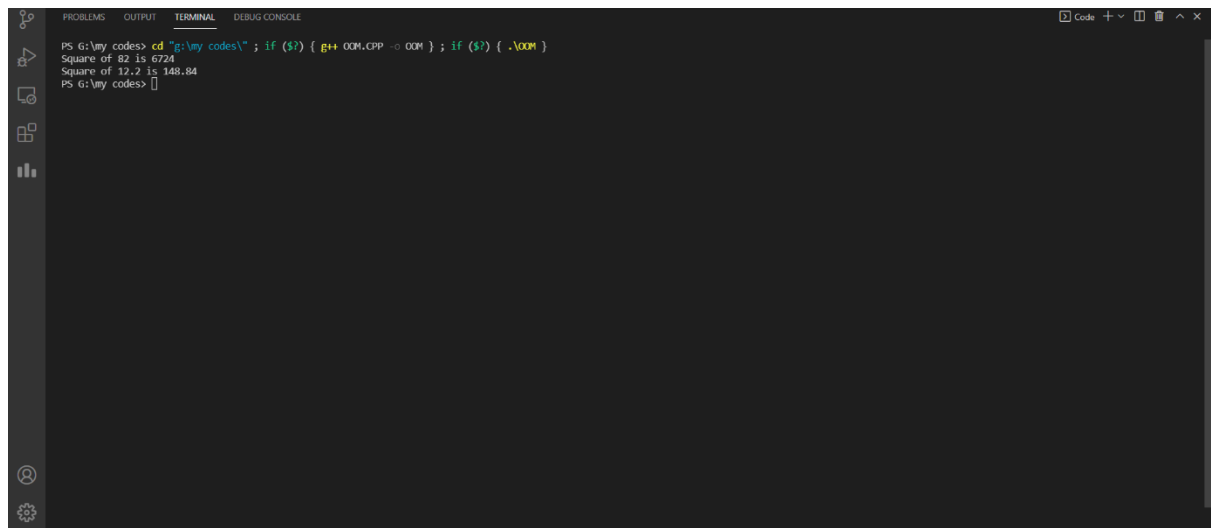
The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal has tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active, showing a PowerShell prompt at G:\my_codes. The user has run a C++ program named OOM.CPP. The program prompts the user to enter three numbers, which are 24, 67, and 88. The program then outputs that the greatest of the three numbers is 88.

```
PS G:\my_codes> cd "G:\my_codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Enter three numbers : 24 67 88
Greatest of three numbers is 88
PS G:\my_codes>
```

4. Write a program to find out square of given numbers of different data types.

```
5. #include <bits/stdc++.h>
6. using namespace std;
7.
8. template <class T>
9. void squareNumber(T val)
10. {
11.     T result = val * val;
12.     cout << "Square of " << val << " is " << result <<
    endl;
13. }
14.
15. int main()
16. {
17.     squareNumber<int>(82);
18.     squareNumber<float>(12.2);
19.     return 0;
20. }
21.
```

OUTPUT:

A screenshot of a terminal window with a dark background. The terminal shows the execution of a C++ program. The first line is a command to compile and run the program: `PS G:\my codes> cd "p:\my codes\" ; if ($?) { g++ 001.cpp -o 001 } ; if ($?) { .\001 }`. The output consists of two lines: `Square of 82 is 6724` and `Square of 12.2 is 148.84`. The prompt `PS G:\my codes>` is visible at the end of the output.

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS G:\my codes> cd "p:\my codes\" ; if ($?) { g++ 001.cpp -o 001 } ; if ($?) { .\001 }
Square of 82 is 6724
Square of 12.2 is 148.84
PS G:\my codes>
```

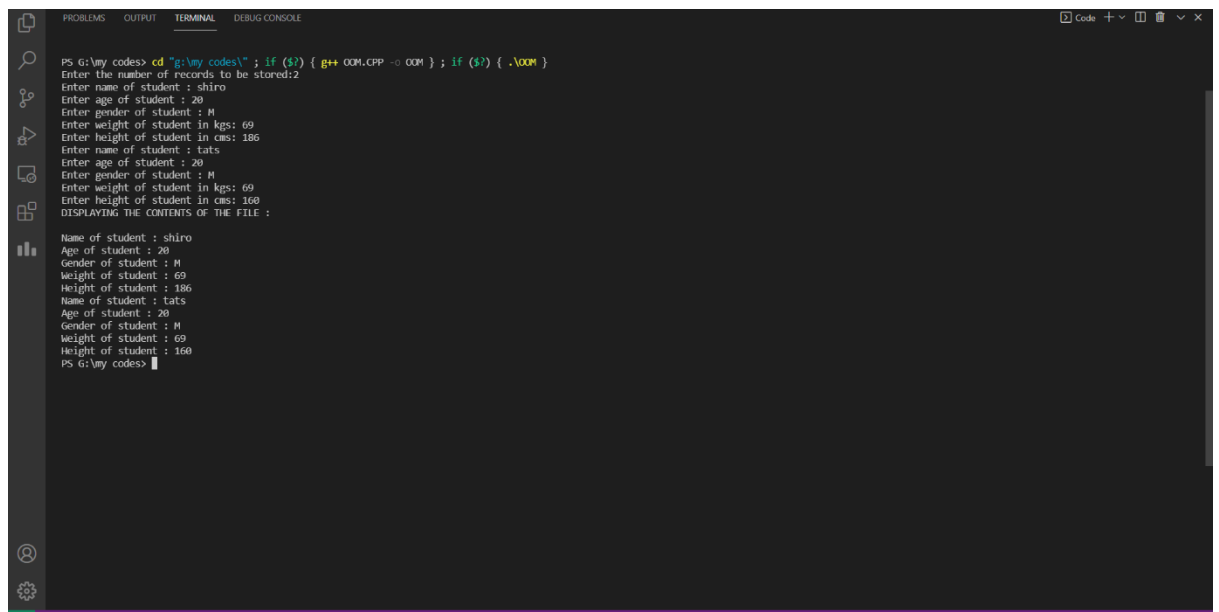
Lab 2

5. Write a program to read class student info such as name, age, gender, height and weight from the keyboard and to store them on a specified file using read() and write() functions. Again, the same file is opened for reading and displaying the contents of the file on the screen.

```
6. #include <bits/stdc++.h>
7. using namespace std;
8.
9. class student
10. {
11.     string name;
12.     int age;
13.     string gender;
14.     float weight;
15.     float height;
16.
17. public:
18.     void read()
19.     {
20.         cout << "Enter name of student : ";
21.         cin >> name;
22.         cout << "Enter age of student : ";
23.         cin >> age;
24.         cout << "Enter gender of student : ";
25.         cin >> gender;
26.         cout << "Enter weight of student in kgs: ";
27.         cin >> weight;
28.         cout << "Enter height of student in cms: ";
29.         cin >> height;
30.     };
31.     void write()
32.     {
33.         cout << "Name of student : " << name << endl;
34.         cout << "Age of student : " << age << endl;
35.         cout << "Gender of student : " << gender << endl;
36.         cout << "Weight of student : " << weight << endl;
37.         cout << "Height of student : " << height << endl;
38.     };
```

```
39.     };
40.
41.     int main()
42.     {
43.
44.         int n;
45.         cout << "Enter the number of records to be stored:";
46.         cin >> n;
47.         student s[n];
48.         ofstream fout;
49.         fout.open("data.txt");
50.         for (int i = 0; i < n; i++)
51.         {
52.             s[i].read();
53.             fout.write((char *)&s[i], sizeof(s[i]));
54.         }
55.         fout.close();
56.         ifstream fin;
57.         cout << "DISPLAYING THE CONTENTS OF THE FILE :\n"
58.              << endl;
59.         fin.open("data.txt");
60.         for (int i = 0; i < n; i++)
61.         {
62.             while (fin.read((char *)&s[i], sizeof(s[i])))
63.             {
64.                 s[i].write();
65.             }
66.         }
67.         fin.close();
68.         return 0;
69.     }
70.
```

OUTPUT:



```
PS G:\my codes> cd "G:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Enter the number of records to be stored:2
Enter name of student : shiro
Enter age of student : 20
Enter gender of student : M
Enter weight of student in kgs: 69
Enter height of student in cms: 186
Enter name of student : tats
Enter age of student : 20
Enter gender of student : M
Enter weight of student in kgs: 69
Enter height of student in cms: 160
DISPLAYING THE CONTENTS OF THE FILE :

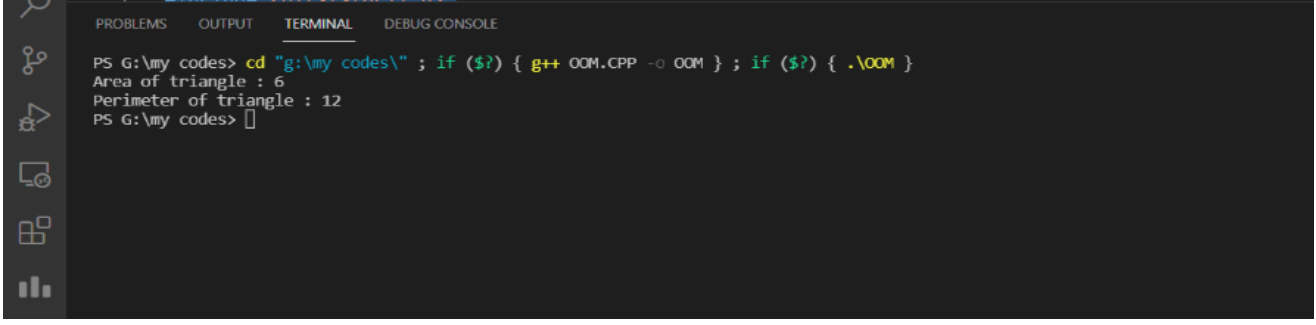
Name of student : shiro
Age of student : 20
Gender of student : M
Weight of student : 69
Height of student : 186
Name of student : tats
Age of student : 20
Gender of student : M
Weight of student : 69
Height of student : 160
PS G:\my codes>
```

Lab 3

6. Write a program to print the area and perimeter of a triangle, having sides of 3, 4 and 5 units by creating a class named 'Triangle', with a function to print the area and perimeter.

```
7. #include <bits/stdc++.h>
8. using namespace std;
9.
10. class Triangle
11. {
12.
13.     float a, b, c;
14.
15. public:
16.     void area(float a, float b, float c)
17.     {
18.         float s = (a + b + c) / 2;
19.         float area;
20.         area = sqrt(s * (s - a) * (s - b) * (s - c));
21.         cout << "Area of triangle : " << area << endl;
22.     }
23.     void perimeter(float a, float b, float c)
24.     {
25.         float perimeter;
26.         perimeter = a + b + c;
27.         cout << "Perimeter of triangle : " << perimeter
28.         << endl;
29.     }
30. };
31. int main()
32. {
33.     Triangle T;
34.     T.area(3, 4, 5);
35.     T.perimeter(3, 4, 5);
36.     return 0;
37. }
38.
```


OUTPUT:



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal has tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The prompt is 'PS G:\my codes>'. The user has entered a command to run a C++ program: `cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }`. The program's output is displayed: 'Area of triangle : 6' and 'Perimeter of triangle : 12'. The prompt returns to 'PS G:\my codes>'.

```
PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Area of triangle : 6
Perimeter of triangle : 12
PS G:\my codes>
```

7. Write a C++ program to compare two strings using overloading.

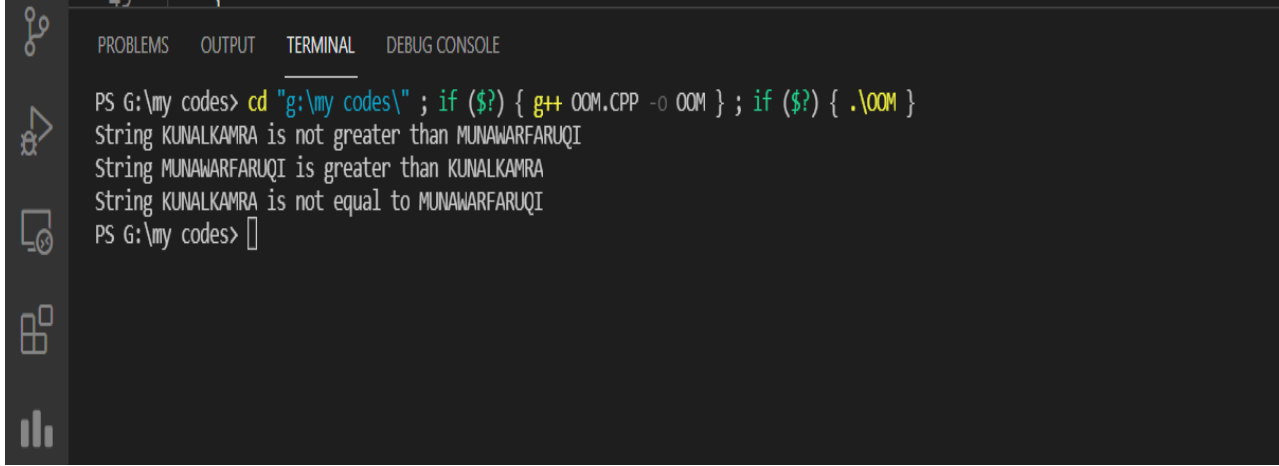
```
8. #include <bits/stdc++.h>
9. using namespace std;
10.
11. class String
12. {
13. public:
14.     string s;
15.     void operator=(String s1)
16.     {
17.         if (s.size() == s1.s.size())
18.         {
19.             int count = 0;
20.             for (int i = 0; i < s1.s.size(); i++)
21.             {
22.                 if (s[i] == s1.s[i])
23.                 {
24.                     count++;
25.                 }
26.                 else
27.                 {
28.                     cout << "String " << s << " is not
equal to " << s1.s << endl;
29.                     break;
30.                 }
31.             }
32.
33.             if (count == s.size())
34.             {
35.                 cout << "String " << s << " is equal to "
<< s1.s << endl;
36.             }
37.         }
38.         else
39.         {
40.             cout << "String " << s << " is not equal to "
<< s1.s << endl;
41.         }
42.     }
43.
```

```

44.     void operator>(String s1)
45.     {
46.         if (s.size() > s1.s.size())
47.         {
48.             cout << "String " << s << " is greater than "
49.             << s1.s << endl;
50.         }
51.         else
52.         {
53.             cout << "String " << s << " is not greater
54.             than " << s1.s << endl;
55.         }
56.     }
57.     void operator<(String s1)
58.     {
59.         if (s.size() < s1.s.size())
60.         {
61.             cout << "String " << s1.s << " is greater
62.             than " << s << endl;
63.         }
64.         else
65.         {
66.             cout << "String " << s1.s << " is not greater
67.             than " << s << endl;
68.         }
69.     };
70.
71.     int main()
72.     {
73.         String s1, s2;
74.         s1.s = "KUNALKAMRA";
75.         s2.s = "MUNAWARFARUQI";
76.         s1 > s2;
77.         s1 < s2;
78.         s1 = s2;
79.         return 0;
80.     }

```

OUTPUT:



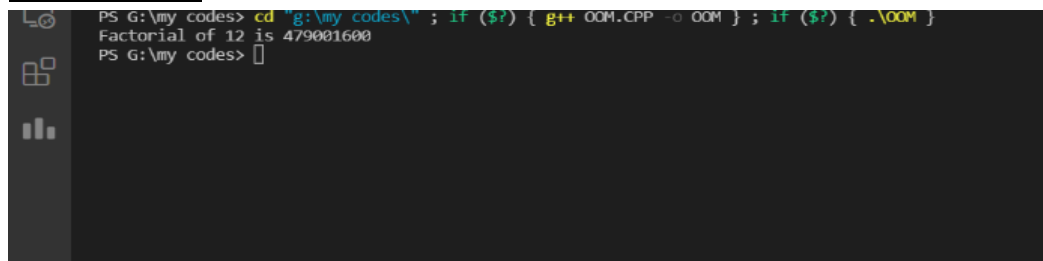
The screenshot shows a terminal window with a dark background. On the left is a vertical sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Testing. The terminal has tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE, with TERMINAL selected. The terminal text is as follows:

```
PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }  
String KUNALKAMRA is not greater than MUNAWARFARUQI  
String MUNAWARFARUQI is greater than KUNALKAMRA  
String KUNALKAMRA is not equal to MUNAWARFARUQI  
PS G:\my codes> 
```

8. Write a C++ program to find the factorial of a number using class and function declared outside the class.

```
9. #include <bits/stdc++.h>
10.     using namespace std;
11.
12.     class Factorial
13.     {
14.         int n;
15.
16.     public:
17.         void factorial(int n);
18.     };
19.
20.     void Factorial ::factorial(int n)
21.     {
22.         int fact = 1;
23.         for (int i = n; i > 0; i--)
24.         {
25.             fact *= i;
26.         }
27.         cout << "Factorial of " << n << " is " << fact <<
endl;
28.     }
29.
30.     int main()
31.     {
32.         Factorial f;
33.         f.factorial(12);
34.         return 0;
35.     }
36.
```

OUTPUT:



```
PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Factorial of 12 is 479001600
PS G:\my codes>
```

Lab 4

9. Implement the class string containing the following functions:

- Overload + operator to carry out the concatenation of strings
- Overload = operator to carry out string copy
- Overload <= operator to carry out comparison of strings
- Function to display the length of the string
- Function tolower() to convert uppercase letter to lowercase
- Function toupper() to convert lowercase letter to uppercase

```
• #include <bits/stdc++.h>
• using namespace std;
•
• class String
• {
• public:
•     string s;
•     void operator+(String S)
•     {
•         s = s + S.s;
•         cout << s << endl;
•     }
•     void operator=(String S)
•     {
•         s = S.s;
•     }
•     void operator<=(String S)
•     {
•         if (S.s.size() > s.size())
•         {
•             cout << S.s << " is greater than " << s <<
endl;
•         }
•     }
• }
```

```

•     else if (S.s.size() == s.size())
•     {
•         int count = 0;
•         for (int i = 0; i < S.s.size(); i++)
•         {
•             if (s[i] == S.s[i])
•             {
•                 count++;
•             }
•             else
•             {
•                 cout << "String " << s << " is not
equal to " << S.s << endl;
•                 break;
•             }
•         }

•         if (count == s.size())
•         {
•             cout << "String " << s << " is equal to "
<< S.s << endl;
•         }
•         else
•         {
•             cout << S.s << " is neither equal to nor
greater than " << s << endl;
•         }
•     }
•     void length()
•     {
•         cout << "Length of string " << s << " is " <<
s.size() << endl;
•     }
•     void toLower()
•     {
•         for (int i = 0; i <= s.size(); i++)
•         {
•             if (isupper(s[i]))
•             {
•                 s[i] += 32;
•             }

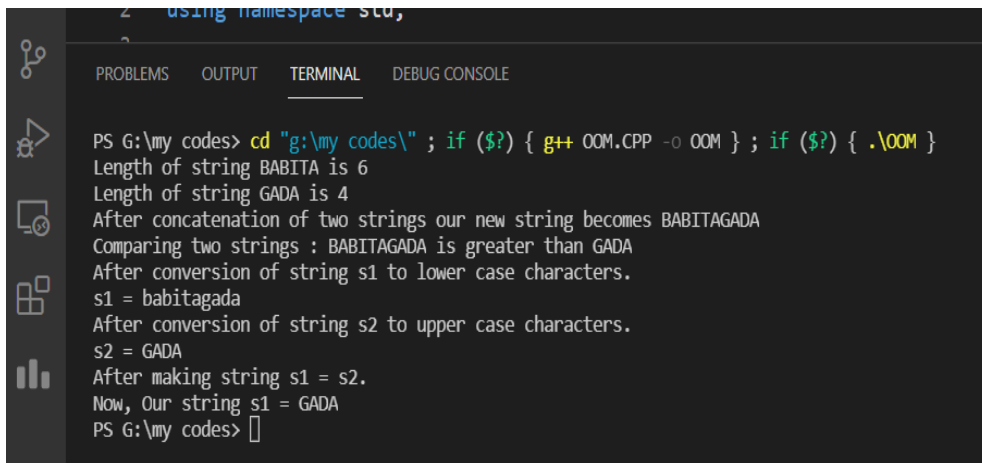
```

```

•     }
•
•     }
•     void toUpper()
•     {
•         for (int i = 0; i < s.size(); i++)
•         {
•             if (islower(s[i]))
•                 s[i] -= 32;
•         }
•     }
•     void display()
•     {
•         cout << s << endl;
•     }
• };
•
• int main()
• {
•     String s1, s2;
•     s1.s = "BABITA";
•     s2.s = "GADA";
•     s1.length();
•     s2.length();
•     cout << "After concatenation of two strings our new
string becomes ";
•     s1 + s2;
•     cout << "Comparing two strings : ";
•     s2 <= s1;
•     s1.toLower();
•     cout << "After conversion of string s1 to lower case
characters.\ns1 = ";
•     s1.display();
•     s2.toUpper();
•     cout << "After conversion of string s2 to upper case
characters.\ns2 = ";
•     s2.display();
•     cout << "After making string s1 = s2.\nNow, Our
string s1 = ";
•     s1 = s2;
•     s1.display();
•     return 0;
• }

```


OUTPUT:



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the output of a C++ program. The code being executed is a C++ program that calculates the length of two strings, concatenates them, compares them, and converts them to lowercase and uppercase. The output shows the results of these operations.

```
2 using namespace std;  
3  
PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }  
Length of string BABITA is 6  
Length of string GADA is 4  
After concatenation of two strings our new string becomes BABITAGADA  
Comparing two strings : BABITAGADA is greater than GADA  
After conversion of string s1 to lower case characters.  
s1 = babitagada  
After conversion of string s2 to upper case characters.  
s2 = GADA  
After making string s1 = s2.  
Now, Our string s1 = GADA  
PS G:\my codes> 
```

- 10. Create a class called LIST with functions store() and retrieve(). To store a value, call store function and to retrieve a value, call retrieve function. Derive two classes stack and queue from it and override store and retrieve.**

```
#include <bits/stdc++.h>
using namespace std;

struct node
{
    int data;
    node *next;
};

class List
{
public:
    node *head = NULL, *tail = NULL;
    void view()
    {
        node *n = head;
        if (head == NULL)
        {
            cout << "\n No elements found...";
        }
        else
        {
            cout << " ";
            while (n != NULL)
            {
                if (n->next == NULL)
                {
                    cout << n->data;
                }
                else
                {
                    cout << n->data << "->";
                }
                n = n->next;
            }
        }
    }
};
```

```

        cout << endl;
    }
}
virtual void store(int n) = 0;
virtual int retrieve() = 0;
};

class Stack : public List
{
public:
    void store(int n)
    {
        node *n1 = new node();
        n1->data = n;
        n1->next = NULL;
        if ((head == NULL) && (tail == NULL))
        {
            head = n1;
            tail = n1;
        }
        else
        {
            tail->next = n1;
            tail = n1;
        }
        cout << n << " stored in Stack." << endl;
    }
    int retrieve()
    {
        if ((tail == NULL) && (head == NULL))
        {
            return -1;
        }
        else
        {
            int n = tail->data;
            node *n1 = head;
            while ((n1->next != tail) && (head != tail))
            {
                n1 = n1->next;
            }
            n1->next = NULL;

```

```

        free(tail);
        if (head != tail)
        {
            tail = n1;
        }
        else
        {
            tail = NULL;
            head = NULL;
        }
        return n;
    }
}

};

class Queue : public List
{
public:
    void store(int n)
    {
        node *n1 = new node();
        n1->data = n;
        n1->next = NULL;
        if ((head == NULL) && (tail == NULL))
        {
            head = n1;
            tail = n1;
        }
        else
        {
            tail->next = n1;
            tail = n1;
        }
        cout << n << " stored in Queue." << endl;
    }
    int retrieve()
    {
        if ((tail == NULL) && (head == NULL))
        {
            return -1;
        }
        else

```

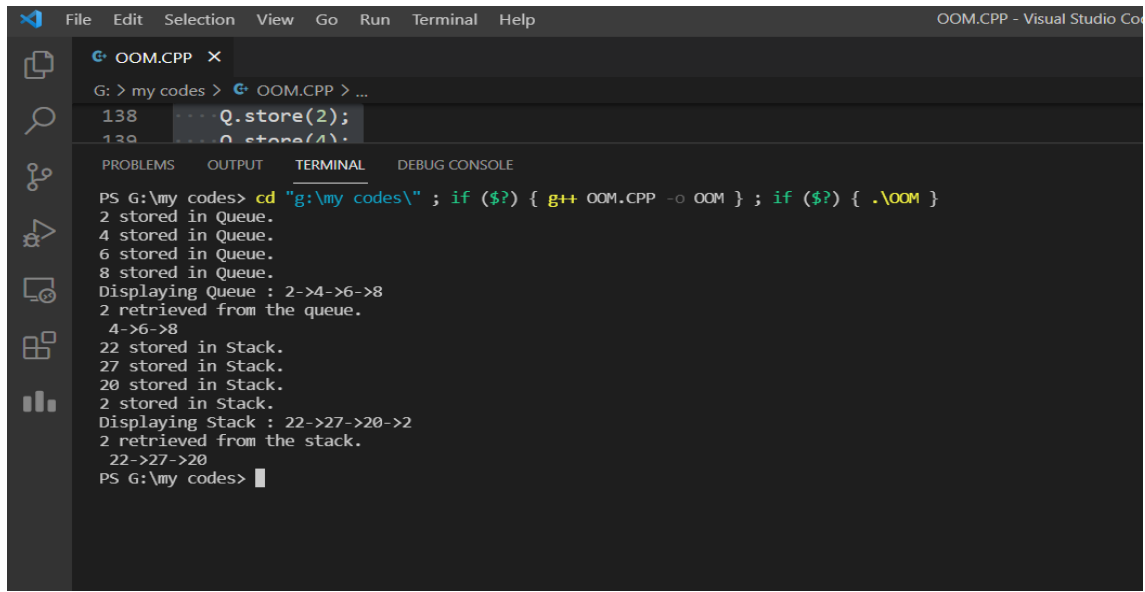
```

        {
            int n = head->data;
            if (head == tail)
            {
                head = tail = NULL;
            }
            else
            {
                head = head->next;
            }
            return n;
        }
    }
};

int main()
{
    Queue Q;
    Q.store(2);
    Q.store(4);
    Q.store(6);
    Q.store(8);
    cout << "Displaying Queue :";
    Q.view();
    cout << Q.retrieve() << " retrieved from the queue." << endl;
    Q.view();
    Stack S;
    S.store(22);
    S.store(27);
    S.store(20);
    S.store(2);
    cout << "Displaying Stack :";
    S.view();
    cout << S.retrieve() << " retrieved from the stack." << endl;
    S.view();
    return 0;
}

```

OUTPUT:



```
File Edit Selection View Go Run Terminal Help OOM.CPP - Visual Studio Co

G: > my codes > OOM.CPP > ...
138 ... Q.store(2);
139 ... Q.store(4);

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
2 stored in Queue.
4 stored in Queue.
6 stored in Queue.
8 stored in Queue.
Displaying Queue : 2->4->6->8
2 retrieved from the queue.
4->6->8
22 stored in Stack.
27 stored in Stack.
20 stored in Stack.
2 stored in Stack.
Displaying Stack : 22->27->20->2
2 retrieved from the stack.
22->27->20
PS G:\my codes>
```

Lab 5

11. Write a program for multiplication of two matrices using OOP.

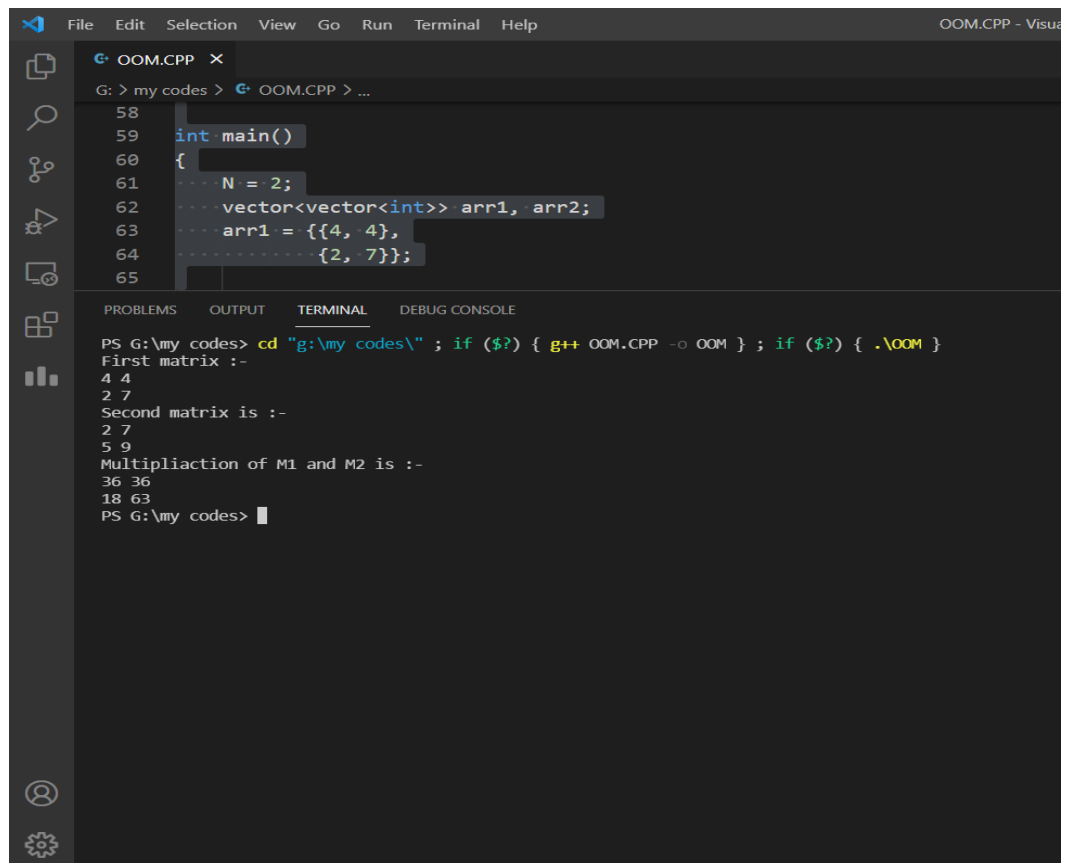
```
12.  #include <bits/stdc++.h>
13.  using namespace std;
14.  #define rows 50
15.  #define cols 50
16.
17.  int N;
18.  class Matrix
19.  {
20.      int mat[rows][cols];
21.
22.  public:
23.      void setMatrix(vector<vector<int>> &A)
24.      {
25.          for (int i = 0; i < N; i++)
26.          {
27.              for (int j = 0; j < N; j++)
28.              {
29.                  mat[i][j] = A[i][j];
30.              }
31.          }
32.      }
33.      void display()
34.      {
35.          for (int i = 0; i < N; i++)
36.          {
37.              for (int j = 0; j < N; j++)
38.              {
39.                  cout << mat[i][j] << " ";
40.              }
41.              cout << endl;
42.          }
43.      }
44.
45.      void operator*(Matrix m)
46.      {
47.          int prod[N][N];
48.          for (int i = 0; i < N; i++)
```

```

49.         {
50.             for (int j = 0; j < N; j++)
51.             {
52.                 prod[i][j] = 0;
53.                 for (int k = 0; k < N; k++)
54.                 {
55.                     prod[i][j] += mat[i][k] * (m.mat[k][j]);
56.                 }
57.             }
58.         }
59.         for (int i = 0; i < N; i++)
60.         {
61.             for (int j = 0; j < N; j++)
62.             {
63.                 cout << prod[i][j] << " ";
64.             }
65.             cout << endl;
66.         }
67.     }
68. };
69.
70. int main()
71. {
72.     N = 2;
73.     vector<vector<int>> arr1, arr2;
74.     arr1 = {{4, 4},
75.             {2, 7}};
76.
77.     arr2 = {{2, 7},
78.             {5, 9}};
79.
80.     Matrix M1, M2;
81.     M1.setMatrix(arr1);
82.     M2.setMatrix(arr2);
83.     cout << "First matrix :- " << endl;
84.     M1.display();
85.     cout << "Second matrix is :- " << endl;
86.     M2.display();
87.     cout << "Multipliaction of M1 and M2 is :-" << endl;
88.     M1 *M2;
89.     return 0;
90. }

```


OUTPUT:



The screenshot shows the Visual Studio Code interface with a C++ file named `OOM.CPP` open. The code defines a `main` function that sets `N = 2`, creates two vectors of vectors, `arr1` and `arr2`, and initializes them with specific values. The `arr1` vector contains `{4, 4}` and `{2, 7}`. The `arr2` vector contains `{5, 9}`. The program then calculates the multiplication of `M1` and `M2` and prints the results.

```
58
59 int main()
60 {
61     N = 2;
62     vector<vector<int>> arr1, arr2;
63     arr1 = {{4, 4},
64             {2, 7}};
65
```

The terminal output shows the execution of the program:

```
PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
First matrix :-
4 4
2 7
Second matrix is :-
2 7
5 9
Multipliacion of M1 and M2 is :-
36 36
18 63
PS G:\my codes>
```

12. Write a program to perform addition of two complex numbers using OOP.

```
#include <bits/stdc++.h>
using namespace std;

class complexNumber
{
    int real;
    int imaginary;

public:
    complexNumber()
    {
        real = 0;
        imaginary = 0;
    };
    complexNumber(int a, int b)
    {
        real = a;
        imaginary = b;
    }
    void printComplexNumber()
    {
        cout << real << " + " << imaginary << "i" << endl;
    }
    void addition(complexNumber c)
    {
        int r1, i1;
        r1 = (real + c.real);
        i1 = (imaginary + c.imaginary);
        complexNumber k(r1, i1);
        k.printComplexNumber();
    }
    // void subtraction(complexNumber c)
    // {
    //     int r1, i1;
    //     r1 = (real - c.real);
    //     i1 = (imaginary - c.imaginary);
    //     complexNumber k(r1, i1);
```

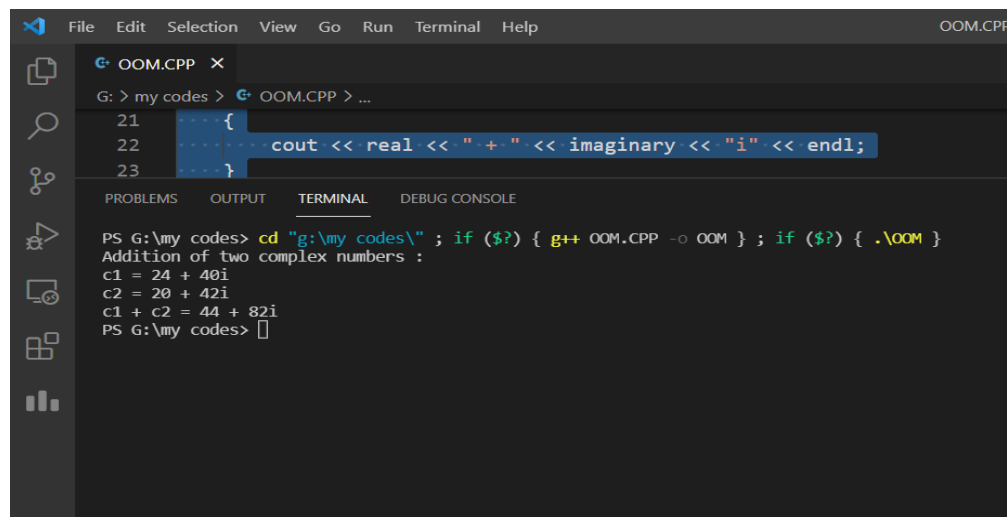
```

        //      k.printComplexNumber();
        // }
};

int main()
{
    complexNumber c1(24, 40), c2(20, 42);
    cout << "Addition of two complex numbers :\n";
    cout << "c1 = ";
    c1.printComplexNumber();
    cout << "c2 = ";
    c2.printComplexNumber();
    cout << "c1 + c2 = ";
    c1.addition(c2);
    // c1.subtraction(c2);
    return 0;
}

```

OUTPUT:



The screenshot shows the Visual Studio Code interface. The editor window displays the source code for `OOM.CPP`. The terminal window shows the output of the program, which is the same as the code in the image above.

```

G: > my codes > G: OOM.CPP > ...
21      {
22      cout << real << " + " << imaginary << "i" << endl;
23      }

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

PS G:\my codes> cd "g:\my codes\" ; if ($?) { g++ OOM.CPP -o OOM } ; if ($?) { .\OOM }
Addition of two complex numbers :
c1 = 24 + 40i
c2 = 20 + 42i
c1 + c2 = 44 + 82i
PS G:\my codes> 

```

