

(SE – OOPS CONCEPT)

{C++ LANGUAGE}

MODULES:- [4.1,4.2,4.2]

Submitted to :-

Mrs. Heena Malek

Submitted by :-

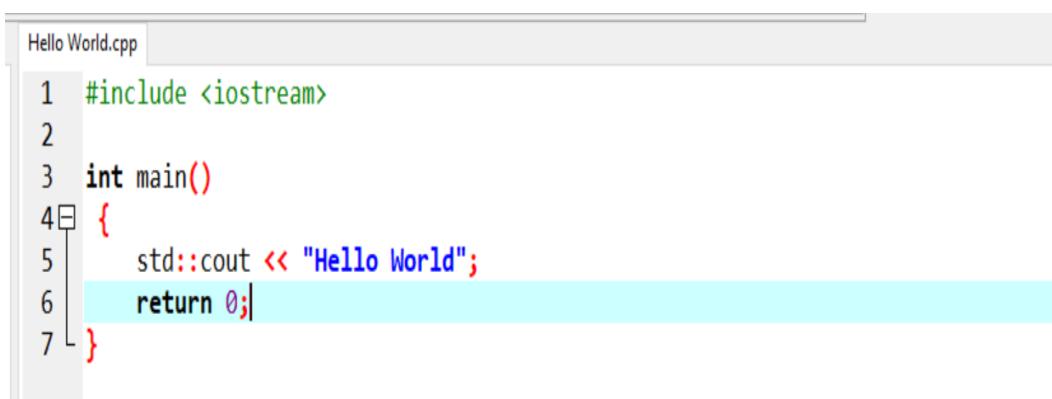
Janvi Panchal

MODULE: 4.1 (C++ BASIC)

1. WAP to print “Hello World” using C++

Ans:-

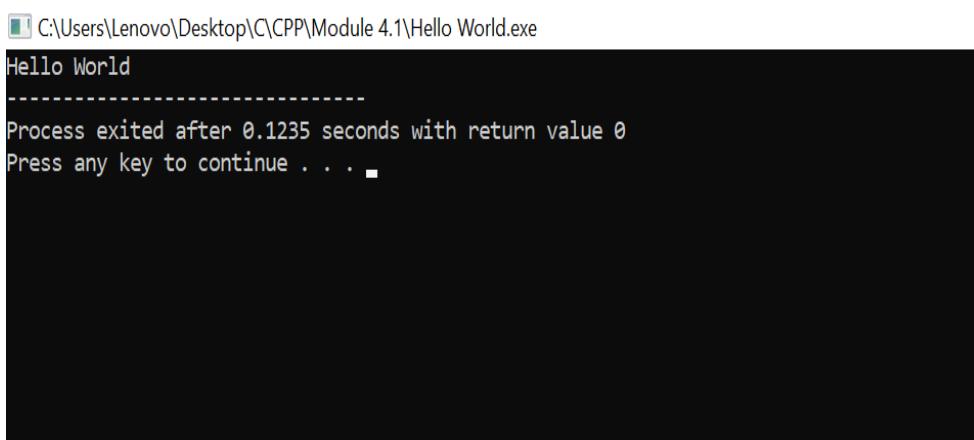
Practical:-



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

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello World";
6     return 0;
7 }
```

Output:-



The screenshot shows a terminal window with the following output:

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.1>Hello World.exe
Hello World
-----
Process exited after 0.1235 seconds with return value 0
Press any key to continue . . .
```

2. What is OOP? List OOP concepts.

Ans:-

What is OOP?

OOPS, or Object-oriented programming is an approach or a **programming** pattern where the programs are structured around objects rather than functions and logic. It makes the data partitioned into two memory areas, i.e., data and functions, and helps make the code flexible and modular.

Object-oriented programming mainly focuses on objects that are required to be manipulated. In OOPs, it can represent data as objects that have attributes and functions.

List OOP concepts

- Classes & Objects
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

3. What is the difference between OOP and POP?

Ans:-

Type	Procedure Oriented Programming	Object-Oriented Programming
Divided Into	In POP, the program is divided into small parts called functions.	In OOP, the program is divided into parts called objects.
Importance	In POP, functions and the order of operations to be performed take precedence over data.	Because it works in the actual world, data takes priority over procedures and functions in OOP.
Approach	POP follows the Top-Down approach.	OOP follows the Bottom-Up approach.
Access Specifiers	POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely from function to function in the system.	In OOP, objects can move and communicate with each other through member functions.
Expansion	To add new data and functions in POP is not so easy.	OOP provides an easy way to add new data and functions.
Data Access	Most functions in POP employ global data for sharing, which can be accessed freely from one function to the next.	Data in OOP cannot simply flow from one function to function; it can be kept public or private, allowing us to regulate data access.
Data Hiding	POP does not have any proper way for hiding data so it is less secure.	OOP provides Data Hiding so provides more security.
Overloading	In POP, Overloading is not possible.	In OOP, Overloading is possible in the form of Operator Overloading and Function Overloading.
Examples	Examples of POP are C, VB, FORTRAN, Pascal.	Examples of OOP are C++, JAVA, VB.NET, C#.NET.

MODULE: 4.2 (PROGRAMMING WITH C++)

1. WAP to create simple calculator using class.

Ans:-

Practical:-

```
Simple Calculator pro1.cpp
1 //WAP to create simple calculator using class
2 # include <iostream>
3 using namespace std;
4
5 int main() {
6
7     char op;
8     float num1, num2;
9
10    cout << "Enter operator: +, -, *, /: ";
11    cin >> op;
12
13    cout << "Enter two operands: ";
14    cin >> num1 >> num2;
15
16    switch(op) {
17
18        case '+':
19            cout << num1 << " + " << num2 << " = " << num1 + num2;
20            break;
21
22        case '-':
23            cout << num1 << " - " << num2 << " = " << num1 - num2;
24            break;
25
26        case '*':
27            cout << num1 << " * " << num2 << " = " << num1 * num2;
28            break;
29
30        case '/':
31            cout << num1 << " / " << num2 << " = " << num1 / num2;
32            break;
33
34        default:
35            // If the operator is other than +, -, * or /, error message is shown
36            cout << "Error! operator is not correct";
37            break;
38    }
39
40    return 0;
41 }
```

Activate Window

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Simple Calculator pro1.exe
```

```
Enter operator: +, -, *, /: *
```

```
Enter two operands: 78
```

```
54
```

```
78 * 54 = 4212
```

```
-----
```

```
Process exited after 19.32 seconds with return value 0
```

```
Press any key to continue . . .
```

2. Define a class to represent a bank account.

Include the following members:

i. Data Member:

- Name of the depositor**
- Account Number**
- Type of Account**
- Balance amount in the account**

ii. Member Functions

- To assign values**
- To deposited an amount**
- To withdraw an amount after checking balance**
- To display name and balance**

Ans:-

Practical:-

```
Bank Account pro2.cpp
1 //Define a class to represent a bank account. Include the following members:
2 #include<iostream>
3 #include<stdio.h>
4 #include<string.h>
5
6 using namespace std;
7
8 class bank
9 {
10     int acno;
11     char nm[100], acctype[100];
12     float bal;
13 public:
14     bank(int acc_no, char *name, char *acc_type, float balance) //Parameterized Constructor
15     {
16         acno=acc_no;
17         strcpy(nm, name);
18         strcpy(acctype, acc_type);
19         bal=balance;
20     }
21     void deposit();
22     void withdraw();
23     void display();
24 };
25 void bank::deposit() //depositing an amount
26 {
27     int damt1;
28     cout<<"\n Enter Deposit Amount = ";
29     cin>>damt1;
30     bal+=damt1;
31 }
32 void bank::withdraw() //withdrawing an amount
33 {
34     int wamt1;
35     cout<<"\n Enter Withdraw Amount = ";
36     cin>>wamt1;
37     if(wamt1>bal)
38         cout<<"\n Cannot Withdraw Amount";
39     bal-=wamt1;
40 }
41 void bank::display() //displaying the details
42 {
43     cout<<"\n -----";
44     cout<<"\n Accout No. : "<<acno;
45     cout<<"\n Name : "<<nm;
46     cout<<"\n Account Type : "<<acctype;
47     cout<<"\n Balance : "<<bal;
48 }
49 int main()
50 {
```

```
51     int acc_no;
52     char name[100], acc_type[100];
53     float balance;
54     cout<<"\n Enter Details: \n";
55     cout<<"-----";
56     cout<<"\n Accout No. ";
57     cin>>acc_no;
58     cout<<"\n Name : ";
59     cin>>name;
60     cout<<"\n Account Type : ";
61     cin>>acc_type;
62     cout<<"\n Balance : ";
63     cin>>balance;
64
65     bank b1(acc_no, name, acc_type, balance); //object is created
66     b1.deposit(); //
67     b1.withdraw(); // calling member functions
68     b1.display(); //
69
70 }
```

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Bank Account pro2.exe

Enter Details:
-----
Accout No. 1678

Name : Janvi

Account Type : 876

Balance : 25,000

Enter Deposit Amount =
Enter Withdraw Amount =
-----
Accout No. : 1678
Name : Janvi
Account Type : 876
Balance : 25
-----
Process exited after 20.03 seconds with return value 0
Press any key to continue . . .
```

3. Write a program to find the multiplication values and the cubic values using inline function.

Ans:-

Practical:-

```
Multiplication values and the Cubic values pro3.cpp
1  /* Write a program to find the multiplication values and the cubic values using
2   inline function */
3
4  #include<iostream>
5  #include<conio.h>
6
7  class line
8  {
9      public:
10     inline float mul(float x, float y)
11     {
12         ...           return(x*y);
13     }
14     inline float cube(float x)
15     {
16         ...           return(x*x*x);
17     }
18 };
19
20 int main()
21 {
22     line obj;
23     float val1, val2;
24     std::cout << "Enter two values:" ;
25     std::cin >> val1 >> val2;
26     std::cout << "\nMultiplication value is:" << obj.mul(val1, val2);
27     std::cout << "\nCube value is:" << obj(cube(val1)) << "\t" << obj(cube(val2));
28     getch();
29 }
```

Activate Windows

Ar

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Multiplication values and the Cubic values pro3.exe
Enter two values:45
78

Multiplication value is:3510

Cube value is:91125      474552
-----
Process exited after 13.25 seconds with return value 0
Press any key to continue . . .
```

4. Write a program of Addition, Subtraction, Division, Multiplication using constructor.

Ans:-

Practical:-

```
Addition, Subtraction, Division, Multiplication 4.cpp
1 /* Write a program of Addition, Subtraction, Division, Multiplication using
2 constructor. */
3 #include <iostream>
4 using namespace std;
5
6 int main(){
7     /* Variable declaration */
8     int x, y;
9     int sum, difference, product, modulo;
10    float quotient;
11
12    // Taking input from user and storing it
13    // in x and y
14    cout << "Enter First Number\n";
15    cin >> x;
16    cout << "Enter Second Number\n";
17    cin >> y;
18
19    // Adding two numbers
20    sum = x + y;
21    // Subtracting two numbers
22    difference = x - y;
23    // Multiplying two numbers
```

Activate Windows

```

24 product = x * y;
25 // Dividing two numbers by typecasting one operand to float
26 quotient = (float)x / y;
27 // returns remainder of after an integer division
28 modulo = x % y;
29
30 cout << "\nSum = " << sum;
31 cout << "\nDifference = " << difference;
32 cout << "\nMultiplication = " << product;
33 cout << "\nDivision = " << quotient;
34 cout << "\nRemainder = " << modulo;
35
36 return 0;
37 }

```

Activate Windows

Output:-

```

C:\Users\Lenovo\Desktop\C\CPP\Module 4.2>Addition, Subtraction, Division, Multiplication 4.exe
Enter First Number
12
Enter Second Number
67

Sum = 79
Difference = -55
Multiplication = 804
Division = 0.179104
Remainder = 12
-----
Process exited after 4.236 seconds with return value 0
Press any key to continue . . .

```

5. Assume a class cricketer is declared. Declare a derived class batsman from cricketer. Data member of batsman. Total runs, Average runs and best performance. Member functions input data, calculate average runs, Display data. (Single Inheritance)

Ans:-

Practical:

```
Data member of batsman pro5.cpp
1  /* Assume a class cricketer is declared. Declare a derived class batsman from
2   cricketer. Data member of batsman. Total runs, Average runs and best
3   performance. Member functions input data, calculate average runs, Display
4   data. (Single Inheritance) */
5
6  #include <iostream>
7  #include <string.h>
8  #include <conio.h>
9  #include<iomanip> //Header file
10 using namespace std;
11
12 class CCricket{
13     private:
14         char country[20], player[20];
15         int avg;
16     public:
17         void getTeams();
18         void compTeams(CCricket *, const int);
19         void dispTeams(const CCricket *, const int);
20     };
21
22 void CCricket :: getTeams(){
23     cout<<"\n Enter the Name of a Country: ";
24     cin>>country;
25     cout<<"\n Enter a Player Name: ";
26     cin>>player;
27     cout<<"\n Enter the Batting Average: ";
28     cin>>avg;
29 }
30
31 void CCricket :: compTeams(CCricket *Ock, const int t_pls){
32     int i, j;
33     CCricket Otemp;
34     // Sorting By Players Name.
35     for(i=0; i<t_pls; i++){
36         for(j=i+1; j<t_pls; j++){
37             if(Ock[i].avg < Ock[j].avg){
38                 Otemp = Ock[i];
39                 Ock[i] = Ock[j];
40                 Ock[j] = Otemp;
41             }
42         }
43     }
44     // Sorting By Country Name.
45     for(i=0; i<t_pls; i++){
46         for(j=i+1; j<t_pls; j++){
47             if(strcmp(Ock[i].country, Ock[j].country) > 0){
48                 Otemp = Ock[i];
49                 Ock[i] = Ock[j];
50             }
51         }
52     }
53 }
```

```
50 } Ock[j] = Otemp;
51 }
52 }
53 }
54 }
55
56 void CCricket :: dispTeams(const CCricket *Ock, const int t_pls){
57     int i, j;
58     char t_c_name[10];
59     // Display Players.
60     cout<<"\n\n Players Sorted According to their Country and Average:- \n";
61     cout<<"\n COUNTRY \t TEAM \t AVERAGE" << endl;
62     for(i=1; i<=t_pls; i++){
63         if(strcmp(t_c_name, Ock[i].country) != 0)
64         {
65             cout<<"\n " << Ock[i].country;
66             strcpy(t_c_name, Ock[i].country);
67         }
68         cout<<"\n\t\t" << Ock[i].player << " - " << std::setw(5) << Ock[i].avg << endl;
69     }
70 }
71
72 int main(){
73     int i=0;
74     char ch;
75     CCricket Ock[30], Otemp;
76     while(1){
77
78         Ock[i].getTeams();
79         i++;
80         cout<<"\n Do you want to Enter next Entry (y/n) ? : ";
81         cin>>ch;
82         if(ch == 'n')
83             break;
84     } // End of while Loop.
85
86     cout<<"\n\n Total Players Entered: " << i << endl;
87
88     // Sort Teams.
89     Otemp.compTeams(Ock, i);
90
91     // Display Teams.
92     Otemp.dispTeams(Ock, i);
93
94     getch();
95 }
```

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Data member of batsman pro5.exe
Enter the Name of a Country: abc
Enter a Player Name: Janvi
Enter the Batting Average: 6
Do you want to Enter next Entry (y/n) ? : y
Enter the Name of a Country: def
Enter a Player Name: Falguni
Enter the Batting Average: 5
Do you want to Enter next Entry (y/n) ? : n

Total Players Entered: 2

Players Sorted According to their Country and Average:-
COUNTRY      TEAM      AVERAGE
abc          Janvi    -       6
def          Falguni -       5
```

6. Create a class person having members name and age. Derive a class student having member percentage. Derive another class teacher having member salary. Write necessary member function to initialize, read and write data. Write also Main function (Multiple Inheritance)

Ans:-

Practical:-

```
1 /* Create a class person having members name and age. Derive a class student
2 having member percentage. Derive another class teacher having member
3 salary. Write necessary member function to initialize, read and write data.
4 Write also Main function (Multiple Inheritance) */
5
6 #include<iostream>
7 using namespace std;
8 class Person{
9     private:
10         string name;
11         int age;
12     public:
13         void setPerson(int a, string n){
14             name = n;
15             age = a;
16         }
17         void displayPerson(){
18             cout<<"Name: "<<name<<endl;
19             cout<<"Age: "<<age<<endl;
20         }
21
22
23 };
24 class Student: public Person{
25     private:
26         float percentage;
27     public:
28         void setStudent(string n, int a, float p){
29             percentage = p;
30             setPerson(a,n);
31         }
32         void displayStudent(){
33             cout<<"Student details are: \n";
34             displayPerson();
35             cout<<"Percentage is: "<<percentage<<endl;
36         }
37 };
38 class Teacher: public Person{
39     private:
40         float salary;
41     public:
42         void setTeacher(string n, int a, float s){
43             salary = s;
44             setPerson(a,n);
45         }
46         void displayTeacher(){
47             cout<<"Teachers details are: \n";
48             displayPerson();
49             cout<<"Salary: "<<salary<<endl;
50         }
51     };
52
53
54 int main(){
55     Student s;
56     s.setStudent("Janvi", 12,80);
57     s.displayStudent();
58     Teacher t;
59     t.setTeacher("Manthan", 15, 9000.98);
60     t.displayTeacher();
61 }
```

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Multiple Inheritance Student Detail Pro6.exe
Student details are:
Name: Janvi
Age: 12
Percentage is: 80
Teachers details are:
Name: Manthan
Age: 15
Salary: 9000.98

-----
Process exited after 0.1474 seconds with return value 0
Press any key to continue . . . ■
```

7. Assume that the test results of a batch of students are stored in three different classes. Class Students are storing the roll number. Class Test stores the marks obtained in two subjects and class result contains the total marks obtained in the test. The class result can inherit the details of the marks obtained in the test and roll number of students. (Multilevel Inheritance)

Ans:-

Practical:-

```
Multilevel Inheritance Student Result pro7.cpp
1 /* Assume that the test results of a batch of students are stored in three different
2 classes. Class Students are storing the roll number. Class Test stores the marks
3 obtained in two subjects and class result contains the total marks obtained in
4 the test. The class result can inherit the details of the marks obtained in the
5 test and roll number of students. (Multilevel Inheritance) */
6
7 #include<iostream>
8 #include<stdio.h>
9 using namespace std;
10
11 class Student
12 {
13     int roll;
14     char name[25];
15 public:
16     void getdata()
17     {
18         cout<<"\n -----";
19         cout<<"\n Enter Roll No.      : ";
20         cin>>roll;
21         cout<<"\n Enter Student Name   : ";
22         cin>>name;
23     }
24     void putdata()
25     {
26         cout<<"\n -----";
27         cout<<"\n ***** Student Marklist *****";
28         cout<<"\n -----";
29         cout<<"\n Roll No.      : "<<roll;
30         cout<<"\n Student Name   : "<<name<<endl;
31     }
32 };
33 class StudentExam : public Student //Class StudentExam derived from Class Student
34 {
35     public:
36     int sub1, sub2, sub3, sub4, sub5, sub6;
37     float per;
38 public:
39     void accept_data()
40     {
41         getdata();
42         cout<<"\n Enter Marks for Subject 1 : ";
43         cin>>sub1;
44         cout<<"\n Enter Marks for Subject 2 : ";
45         cin>>sub2;
46         cout<<"\n Enter Marks for Subject 3 : ";
47         cin>>sub3;
48         cout<<"\n Enter Marks for Subject 4 : ";
49         cin>>sub4;
50         cout<<"\n Enter Marks for Subject 5 : ";
```

```

51         cin>>sub5;
52         cout<<"\n Enter Marks for Subject 6 : ";
53         cin>>sub6;
54     }
55     void display_data()
56     {
57         putdata();
58         cout<<"\n Marks of Subject 1 : "<<sub1;
59         cout<<"\n Marks of Subject 2 : "<<sub2;
60         cout<<"\n Marks of Subject 3 : "<<sub3;
61         cout<<"\n Marks of Subject 4 : "<<sub4;
62         cout<<"\n Marks of Subject 5 : "<<sub5;
63         cout<<"\n Marks of Subject 6 : "<<sub6;
64     }
65 };
66 class StudentResult : public StudentExam //Class StudentResult derived from Class StudentExam
67 {
68     public:
69     void calculate ()
70     {
71         per = (sub1+sub2+sub3+sub4+sub5+sub6)/6.0;
72         cout<<"\n\n Total Percentage : "<<per;
73         cout<<"\n ----- \n";
74     }
75 };
76 int main()
77 {
78     StudentResult str; //Object 'str' is created of derived Class StudentResult
79     int cnt, i;
80     cout<<"\n Enter No. of Students You Want? : ";
81     cin>>cnt;
82     for(i=0; i<cnt; i++)
83     {
84         str.accept_data();
85         str.display_data();
86         str.calculate();
87     }
88     return 0;
89 }
```

Activate Windows

Activate Windows

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Multilevel Inheritance Student Result pro7.exe
```

```
Enter No. of Students You Want? : 1

-----
Enter Roll No.      : 19
Enter Student Name : Janvi
Enter Marks for Subject 1 : 90
Enter Marks for Subject 2 : 89
Enter Marks for Subject 3 : 78
Enter Marks for Subject 4 : 67
Enter Marks for Subject 5 : 75
Enter Marks for Subject 6 : 93

-----
***** Student Marklist *****
-----
Roll No.      : 19
Student Name : Janvi
Marks of Subject 1 : 90
Marks of Subject 2 : 89
Marks of Subject 3 : 78
Marks of Subject 4 : 67
Marks of Subject 5 : 75
Marks of Subject 6 : 93
Total Percentage : 82

-----
Process exited after 24.69 seconds with return value 0
Press any key to continue . . .
```

8. Write a program to Mathematic operation like Addition, Subtraction, Multiplication, Division Of two number using different parameters and Function Overloading

Ans:-

Practical:-

Mathematical pro8 Operation Using Function Overloading.cpp Multilevel Inheritance Student Result pro7.cpp

```
1 /* Write a program to Mathematic operation like Addition, Subtraction,
2 Multiplication, Division Of two number using different parameters and
3 Function Overloading */
4
5 #include <iostream>
6 using namespace std;
7
8 int main(){
9     /* Variable declaration */
10    int x, y;
11    int sum, difference, product, modulo;
12    float quotient;
13
14    // Taking input from user and storing it
15    // in x and y
16    cout << "Enter First Number\n";
17    cin >> x;
18    cout << "Enter Second Number\n";
19    cin >> y;
20
21    // Adding two numbers
22    sum = x + y;
23    // Subtracting two numbers
24    difference = x - y;
25    // Multiplying two numbers
26    product = x * y;
27
28    // Dividing two numbers by typecasting one operand to float
29    quotient = (float)x / y;
30    // returns remainder of after an integer division
31    modulo = x % y;
32
33    cout << "\nSum = " << sum;
34    cout << "\nDifference = " << difference;
35    cout << "\nMultiplication = " << product;
36    cout << "\nDivision = " << quotient;
37    cout << "\nRemainder = " << modulo;
38
39    return 0;
}
```

Activate Windows

Activate Windows

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Mathematical pro8 Oparation Using Functio Overloading.exe
Enter First Number
87
Enter Second Number
67

Sum = 154
Difference = 20
Multiplication = 5829
Division = 1.29851
Remainder = 20
-----
Process exited after 11.21 seconds with return value 0
Press any key to continue . . .
```

9. Write a Program of Two 1D Matrix Addition using Operator Overloading

Ans:-

Practical:-

```
Mathematical pro8 Operation Using Function Overloading.cpp Multilevel Inheritance Student Result pro7.cpp Matrix Addition using Operator Overloading 9.cpp
1 //Write a Program of Two 1D Matrix Addition using Operator Overloading
2
3 #include<iostream>
4 using namespace std;
5
6 class Matrix
7 {
8     int a[3][3];
9 public:
10    void accept();
11    void display();
12    void operator +(Matrix x);
13 };
14 void Matrix::accept()
15 {
16     cout<<"\n Enter Matrix Element (3 X 3) : \n";
17     for(int i=0; i<3; i++)
18     {
19         for(int j=0; j<3; j++)
20         {
21             cout<<" ";
22             cin>>a[i][j];
23         }
24     }
25 }
26 void Matrix::display()
27 {
28     for(int i=0; i<3; i++)
29     {
30         cout<<" ";
31         for(int j=0; j<3; j++)
32         {
33             cout<<a[i][j]<<"\t";
34         }
35         cout<<"\n";
36     }
37 }
38 void Matrix::operator +(Matrix x)
39 {
40     int mat[3][3];
41     for(int i=0; i<3; i++)
42     {
43         for(int j=0; j<3; j++)
44         {
45             mat[i][j]=a[i][j]+x.a[i][j];
46         }
47     }
48     cout<<"\n Addition of Matrix : \n\n";
49     for(int i=0; i<3; i++)
50     {
51     }
```

Activate Windows

```

51     cout<< " ";
52     for(int j=0; j<3; j++)
53     {
54         cout<<mat[i][j]<<"\t";
55     }
56     cout<<"\n";
57 }
58
59 int main()
60 {
61     Matrix m,n;
62     m.accept();      // Accepting Rows
63     n.accept();      // Accepting Columns
64     cout<<"\n First Matrix : \n\n";
65     m.display();    // Displaying First Matrix
66     cout<<"\n Second Matrix : \n\n";
67     n.display();    // Displaying Second Matrix
68     m+n;           // Addition of Two Matrices. Overloaded '+' Operator
69     return 0;
70 }
71

```

Activate Window:

Output:-

C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Matrix Addition using Operator Overloading 9.exe

```

Enter Matrix Element (3 X 3) :
1 2 34 5 6 7 88 90 45

Enter Matrix Element (3 X 3) :
34 56 78 90 2 3 4 5 43

First Matrix :

1      2      34
5      6      7
88     90     45

Second Matrix :

34      56      78
90      2       3
4       5       43

Addition of Matrix :

35      58      112
95      8       10
92      95     88

-----
Process exited after 31.77 seconds with return value 0
Press any key to continue . . .

```

10. Write a program to concatenate the two strings using Operator Overloading

Ans:-

Practical:-

```
Two Strings using Operator Overloading 10.cpp
1 //Write a program to concatenate the two strings using Operator Overloading
2
3 #include <iostream>
4 #include <string.h>
5
6 using namespace std;
7
8 // Class to implement operator overloading
9 // function for concatenating the strings
10 class AddString {
11
12 public:
13     // Classes object of string
14     char s1[25], s2[25];
15
16     // Parameterized Constructor
17     AddString(char str1[], char str2[])
18     {
19         // Initialize the string to class object
20         strcpy(this->s1, str1);
21         strcpy(this->s2, str2);
22     }
23
24
25     // Overload Operator+ to concat the string
26     void operator+()
27     {
28         cout << "\nConcatenation: " << strcat(s1, s2);
29     }
30
31     // Driver Code
32     int main()
33     {
34         // Declaring two strings
35         char str1[] = "Janvi ";
36         char str2[] = "Panchal";
37
38         // Declaring and initializing the class
39         // with above two strings
40         AddString a1(str1, str2);
41
42         // Call operator function
43         +a1;
44         return 0;
45     }

```

Activate Windows

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Two Strings using Operator Overloading 10.exe
```

```
Concatenation: Janvi Panchal
-----
Process exited after 0.1205 seconds with return value 0
Press any key to continue . . .
```

11. Write a program to calculate the area of circle, rectangle and triangle using Function Overloading

Rectangle: Area * breadth

Triangle: $\frac{1}{2} * \text{Area} * \text{breadth}$

Circle: Pi * Area * Area

Ans:-

Practical:-

```
circle, rectangle and triangle using Pro11 Overloading.cpp
1  /* Write a program to calculate the area of circle, rectangle and triangle using
2  Function Overloading */
3
4  #include<iostream>
5  using namespace std;
6  int area(int);
7  int area(int,int);
8  float area(float);
9  float area(float,float);
10 int main()
11 {
12     int s,l,b;
13     float r,bs,ht;
14     cout<<"Enter side of a square:";
15     cin>>s;
16     cout<<"Enter length and breadth of rectangle:";
17     cin>>l>>b;
18     cout<<"Enter radius of circle:";
19     cin>>r;
20     cout<<"Enter base and height of triangle:";
21     cin>>bs>>ht;
22     cout<<"\nArea of square is "<<area(s);
23     cout<<"\nArea of rectangle is "<<area(l,b);
24     cout<<"\nArea of circle is "<<area(r);
25     cout<<"\nArea of triangle is "<<area(bs,ht);
26 }
27 int area(int s)
28 {
29     return(s*s);
30 }
31 int area(int l,int b)
32 {
33     return(l*b);
34 }
35 float area(float r)
36 {
37     return(3.14*r*r);
38 }
39 float area(float bs,float ht)
40 {
41     return((bs*ht)/2);
42 }
```

Activate Windows

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\circle, rectangle and triangle using Pro11 Overloading.exe
Enter side of a square:12
Enter length and breadth of rectangle:67
70
Enter radius of circle:56
Enter base and height of triangle:90
54
Area of square is 144
Area of rectangle is 4690
Area of circle is 9847.04
Area of triangle is 2430
-----
Process exited after 47.23 seconds with return value 0
Press any key to continue . . .
```

12. Write a program to swap the two numbers using friend function without using third variable

Ans:-

Practical:-

```
Swap Value Without using third variable 12.cpp
1 /*Write a program to swap the two numbers using friend function without
2 using third variable */
3
4 #include <iostream>
5 using namespace std;
6 int main()
7 {
8     int a=5, b=10;
9     cout<<"Before swap a= "<<a<< " b= "<<b<<endl;
10    a=a*b; //a=50 (5*10)
11    b=a/b; //b=5 (50/10)
12    a=a/b; //a=10 (50/5)
13    cout<<"After swap a= "<<a<< " b= "<<b<<endl;
14    return 0;
15 }
```

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Swap Value Without using third variable 12.exe
Before swap a= 5 b= 10
After swap a= 10 b= 5

-----
Process exited after 0.1271 seconds with return value 0
Press any key to continue . . . ■
```

13. Write a program to find the max number from given two numbers using friend function

Ans:-

Practical:-

```
Swap Value Without using third variable 12.cpp Find Max Number using Friend Function 13.cpp
1  /* Write a program to find the max number from given two numbers using
2   friend function */
3
4  #include<iostream>
5  using namespace std;
6
7  class a;
8
9  class b
10 {
11     int number;
12     public:
13     b(int x)
14     {
15         number=x;
16     }
17     void friend greatest(a a1,b b1);
18 };
19
20 class a
21 {
22     int number;
23     public:
24     a(int x)
25     {
26         number=x;
27     }
28     void friend greatest(a a1,b b1);
29 };
30
31 void greatest(a a1,b b1)
32 {
33     if(a1.number>b1.number)
34     {
35         cout<<"\n Number in class A is greatest i.e. "<<a1.number;
36     }
37     else if(a1.number<b1.number)
38     {
39         cout<<"\n Number in class B is greatest i.e. "<<b1.number;
40     }
41     else
42     {
43         cout<<"\n Both are equal i.e. "<<a1.number;
44     }
45 }
```

Activate Windows

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Find Max Number using Friend Function 13.exe

Program to find greatest of two numbers in two different classes using friend function

Enter number for class A - 44

Enter number for class B - 87

Number in class B is greatest i.e. 87

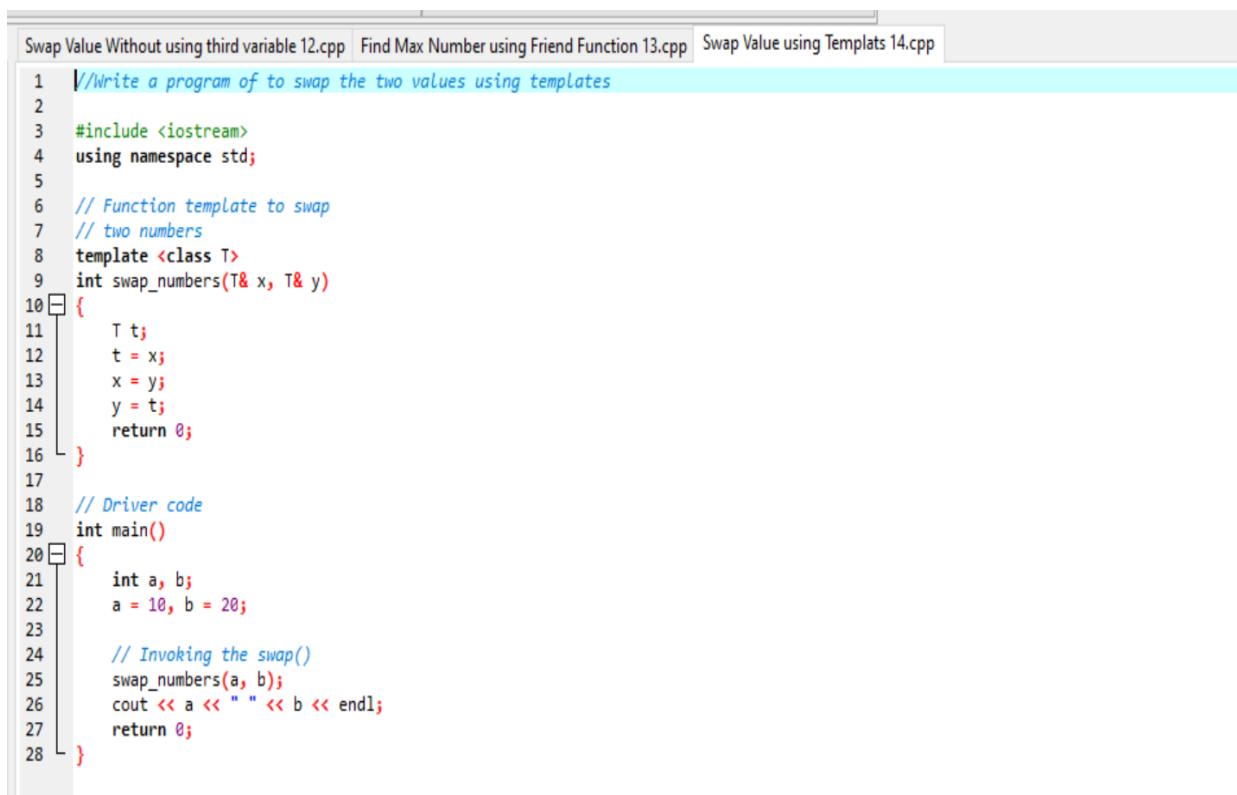
-----
Process exited after 13.34 seconds with return value 0
Press any key to continue . . .
```

MODULE: 4.2 (C, C++ TEMPLATES)

1. Write a program of to swap the two values using Templates

Ans:-

Practical:-



The screenshot shows a code editor window with three tabs at the top: "Swap Value Without using third variable 12.cpp", "Find Max Number using Friend Function 13.cpp", and "Swap Value using Templates 14.cpp". The "Swap Value using Templates 14.cpp" tab is active. The code in the editor is as follows:

```
Swap Value Without using third variable 12.cpp Find Max Number using Friend Function 13.cpp Swap Value using Templates 14.cpp

1 //Write a program of to swap the two values using templates
2
3 #include <iostream>
4 using namespace std;
5
6 // Function template to swap
7 // two numbers
8 template <class T>
9 int swap_numbers(T& x, T& y)
10 {
11     T t;
12     t = x;
13     x = y;
14     y = t;
15     return 0;
16 }
17
18 // Driver code
19 int main()
20 {
21     int a, b;
22     a = 10, b = 20;
23
24     // Invoking the swap()
25     swap_numbers(a, b);
26     cout << a << " " << b << endl;
27     return 0;
28 }
```

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Swap Value using Templates 14.exe
20 10
-----
Process exited after 0.04846 seconds with return value 0
Press any key to continue . . .
```

2. Write a program of to sort the array using templates.

Ans:-

Practical:-

```
Swap Value Without using third variable 12.cpp | Find Max Number using Friend Function 13.cpp | Swap Value using Templates 14.cpp | Sort List using Templates 15.cpp
1 //Write a program of to sort the array using templates.
2
3 #include<iostream>
4 using namespace std;
5 #define n 10 //Global Variable Dicleretion
6
7 template <class T>
8 void sort(T arr[], int SIZE){
9     for (int i = 0; i < SIZE; i++)
10    {
11        for (int j = i+1; j < SIZE; j++)
12        {
13            if (arr[i] > arr[j])
14            {
15                T temp;
16                temp = arr[i];
17                arr[i] = arr[j];
18                arr[j] = temp;
19            }
20        }
21    }
22
23
24 int main()
25 {
26     int int_array[n];
27     float float_array[n];
28     cout<<"Enter integer array elements:"<<endl;
29     for (int i = 0; i < 5; i++)
30    {
31         cin>>int_array[i];
32     }
33     cout<<"Enter floating array elements:"<<endl;
34     for (int i = 0; i < 5; i++)
35    {
36         cin>>float_array[i];
37     }
38     sort(int_array,n);
39     sort(float_array, n);
40     cout<<"After sorting they are :"<<endl;
41     for (int i = 0; i < n; i++)
42    {
43         cout<<int_array[i]<<", ";
44     }
45     cout<<endl;
46     for (int i = 0; i < n; i++)
47    {
48         cout<<float_array[i]<<", ";
49     }
50
51 }
```

Output:-

```
C:\Users\Lenovo\Desktop\C\CPP\Module 4.2\Sort List using Templates 15.exe
Entner integer array elements:
34
56
78
90
1
3
5
7
8
90
Entner floating array elements:
7
66
55
7
13
67
89
83
55
43
After sorting they are :
1, 3, 5, 7, 8, 34, 56, 78, 90, 90,
7, 7, 13, 43, 55, 55, 66, 67, 83, 89,
-----
Process exited after 28.39 seconds with return value 0
Press any key to continue . . .
```