

ASSIGNMENT 7

MATRIX OPERATION

Problem Statement :-

Write a program in C++ using function template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.

Learning Objectives :-

- i. Templates
- Etc.,

Theory :-

Templates :-

Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type.

A template is a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using template concept.

You can use templates to define functions as well as classes.

Function Template :-

The general form of a template function definition is shown here :

```
template <class type> return_type function_name(parameter list)
{
    // body of function
}
```

Here, type is a placeholder name for a data type used by the function. This name can be used within the function definition.

```
template<typename type>
Type min(Type tA, Type tB)
{
    return ( tA > tB ) ? tA : tB;
```

If the template function makes use of multiple template type parameter, they can be separated by commas.

```
template<typename T1, typename T2>
//template function here
```

```
int nValue = min(3,7);           //returns 3
double dValue = min(6.34,18.52); //returns 6.34
```

Class Template :-

we can define function templates, we can also define class templates. The general form of a generic class declaration is shown here:

```
template<class Type>
class class_name
{
    //body of the class
};
```

Here, type is the placeholder type name, which will be specified when a class is instantiated. You can define more than one generic data type by using a comma-separated list.

```
template<class T>
class matrix
{
    T m[10][10];
    int row, col;
    public :
        //member functions
};
```

Related Mathematics :-

//Input :-

$$M_i = \{ m, r, c \}$$

The set M_i represents matrix in which m is actual matrix and r and c are rows and columns respectively.

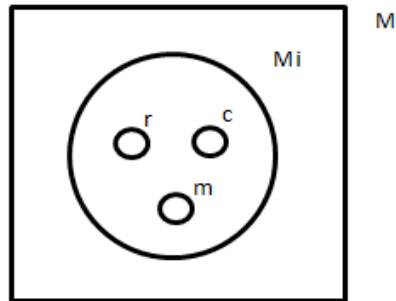
//Output :-

$$M = \{ M_1, M_2, M_3, \dots, M_n \}$$

Where, M is a set of n matrices.

$$A = (M, +, -, *)$$

Where, A is a algebraic system in which the operations $+$, $-$, $*$ are performed on set M .



Algorithm :-

1. START
2. Create a class matrix using template T
3. Declare data member T m[10][10]
4. Write get_value() to accept the values in the matrix having n rows and m columns
5. Write add() to add matrices of type T.
6. Write sub() to subtract matrices of type T.
7. Write mul() to multiply matrices of type T.
8. Write tra() to transpose matrices of type T.
9. Write display() to display matrix of type T.
10. Create objects of type T in the main()
11. Accept the values of rows and columns for matrix A and B.
12. Further program is based on selection logic.
13. Display Menu :-
 - i. Addition
 - ii. Subtraction
 - iii. Multiplication
 - iv. Transpose
 - v. Exit
14. If first option is selected, then perform add() to add matrices.
15. If second option is selected, then perform sub() to subtract matrices.
16. If third option is selected, then perform mul() to multiply matrices.
17. If fourth option is selected, then perform tra() for transpose operation.
18. If fifth option is selected, then exit from loop.
19. STOP

Conclusion :-

Using function template, we can perform matrix operations.