Practical-12

Q. Program to create a matrix class with matrix functions , overloaded operators and performing exception handling.

Code:-

```
//Harsh Bamotra AC-1216
//Program to perform matrix class with overloading operators and exception handling.

#include <iostream>
using namespace std;

/*************************************** Creating Matrix class ***************************************/
class matrix
  {
        private:                                        //defining  private members
        int row , col;

        public:                                         //defining public members
        int arr[10][10];
        void setData(int n1 , int n2)                   //defining function to initialize the private members
                {
                        row=n1;
                        col=n2;
                }

        void create_arr()                               //defining function to create an array
                {
                        for(int i=0 ; i<row ; i++)
                                {
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        cout << "Enter the elements at index [" << i << "][" << j << "]::";
                                                        cin >> arr[i][j];
                                                }
                                }
                }

        void display_arr()                              //defining function to print the array
                {
                        for(int i=0 ; i<row ; i++)
                                {
                                        cout << endl;
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        cout << arr[i][j] << " ";
                                                }
                                }
                }
```

```
    void trans();                                   //defining function for transpose of the matrix

/************************************** Overloading operators ************************************/

    matrix operator +(matrix m)
        {                                           //overloading + operator for adding two matrix
            matrix temp;                            //defining temp matrix class
            temp.row=m.row;
            temp.col=m.col;
            if(row==m.row && col==m.col)
                {
                for(int i=0 ; i<row ; i++)
                    {
                        for(int j=0 ; j<col ; j++)
                            {
                                temp.arr[i][j]=arr[i][j]+m.arr[i][j];
                            }                                   //adding the elements
                    }                                   //and initializing them in the temp
                }
            else
                {
                cout << "Error !! The order of the matrix are not same. ";
                }

            return temp;                            //returning the result
        }



    matrix operator -(matrix m)
        {                                           //overloading – operator for subtracting two matrix
            matrix temp;                            //defining temp matrix class to store the sum
            temp.row=m.row;
            temp.col=m.col;
            if(row==m.row && col==m.col)
                {
                for(int i=0 ; i<row ; i++)
                    {
                        for(int j=0 ; j<col ; j++)
                            {
                                temp.arr[i][j]=arr[i][j]-m.arr[i][j];
                            }                                   //subtracting the elements
                    }                                   //and initializing them in temp
                }
            else
                {
                cout << "Error !! The order of the matrix are not same. ";
                }

            return temp;                            //returning the result
        }
```

```
        matrix operator *(matrix m)                                     //overloading * operator to multiply two matrix
                {
                        matrix temp;                                    //defining temp matric class
                        temp.row=row;
                        temp.col=m.col;
                        for(int i=0 ; i<row ; i++)
                                {
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        temp.arr[i][j]=0;    //initializing the elements of temp to 0
                                                }
                                }
                        if(col==m.row)
                                {
                                        for(int i=0 ; i<row ; i++)
                                                {
                                                        for(int j=0 ; j<m.col ; j++)
                                                                {
                                                                        for(int k=0 ; k<col ; k++)
                                                                                {
                                                                                temp.arr[i][j]+=arr[i][k]*m.arr[k][j];
                                                                                }
                                                                }      //multiplying the matrix
                                                }
                                }
                        else
                                {
                                        cout << "Error !! the column of the fisrt matrix not equal to the row of second.";
                                }
                        return temp;                                    //returning the result matrix
                }

   };



void matrix :: trans()                          // defining function trans for transposing matrix
        {
                int arr1[10][10];
                for(int i=0 ; i<row ; i++)
                        {
                                for(int j=0 ; j<col ; j++)
                                        {
                                                arr1[i][j]=arr[j][i];        //transposing the matrix
                                        }
                        }

                cout << "The matrix after transpose::";
                for(int i=0 ; i<row ; i++)
                        {
                                cout << endl;
                                for(int j=0 ; j<col ; j++)
                                        {
                                                cout << arr1[i][j] << " ";
                                        }                                               //printing the transposed matrix
                        }
        }
```

```
int main()
  {
        int r1 , r2 , c1 , c2 , x , y;                                  //defining variables
        matrix m1 , m2 , m3 ;                                           //defining matrix class object
        cout << "*****Enter the details of first matrix*****" << endl << endl;
        cout << "Enter the number of row::";
        cin >> r1;                                                      //taking number of rows from the user
        cout << "Enter the number of columns::";
        cin >> c1;                                                      //taking number of columns from the user
        m1.setData(r1 , c1);
        m1.create_arr();                                                //initializing members and creating matrix
        cout << "The matrix you entered::" << endl;
        m1.display_arr();                                               //printing the matrix

        cout << endl << endl << "*****Enter the details of second matrix*****" << endl << endl;
        cout << "Enter the number of row::";
        cin >> r2;                                                      //taking number of rows from the user
        cout << "Enter the number of columns::";
        cin >> c2;                                                      //taking number of columns from the user
        m2.setData(r2 , c2);
        m2.create_arr();                                                //initializing members and creating matrix
        cout << "The matrix you entered::" << endl;
        m2.display_arr();                                               //printing the matrix

        cout << endl << "**********************************************************************" << endl;
        cout << "1.Sum" << endl << "2.Product" << endl << "3.Transpose" << endl << "4.Subracting" << endl ;
        cout << "**********************************************************************" << endl;
        cout << "Enter your choice(1 , 2 , 3 or 4)::";
        cin >> x;                                                       //printing the menu and taking user's choice
        switch (x)                                                      //defining switch case
        {
        case 1:
                try                                                     //handling exception
                {
                        if(r1==r2 && c1==c2)
                                {
                                        m3=m1+m2;           //adding and printing the matrix
                                        cout << "The sum of the matrix::" << endl;
                                        m3.display_arr();
                                }
                        else
                                {
                                        throw r1;
                                }
                }

                catch(int err)                                          //printing error message
                        {
                                cout << "Error !! The order of the matrix must be same to perform sum.";
                        }
                break;

        case 2:
                try
                {                                               //handling exception
                        if(c1==r2)
                                {
                                        m3=m1*m2;                               //multiplying matrix
```

```
                                        cout << "The product of the matrix::" << endl;
                                        m3.display_arr();                                       //printing the result
                                }
                        else
                                {
                                        throw r1;
                                }
                }
                catch(int err)                                                  //printing the error message
                        {
                                cout << "Error !! The column of the first matrix must be same to the row of the second.";
                        }
                break;

        case 3:
                cout << "**********************************************************************" << endl;
                cout << "Which matrix you want to transpose (1 or 2)::";
                cin >> y;
                if(y==1)
                        {
                                m1.trans();
                        }
                else if(y==2)                                                   //transposing and printing matrix
                        {
                                m2.trans();
                        }
                else
                        {
                                cout << "Wrong Input!!" << endl ;
                        }
                break;

        case 4:
                try                                                                     //handling exception
                {
                        if(r1==r2 && c1==c2)
                                {
                                        m3=m1-m2;                                       subtracting and printing the matrix
                                        cout << "The subtraction of the matrix::";
                                        m3.display_arr();
                                }
                        else
                                {
                                        throw r1;
                                }
                }
                catch(int err)                                  //printing the error message
                        {
                                cout << "Error !! The order of the matrix must be same to perform subtraction.";
                        }
                break;

        default:                                                //exiting message
                cout << "Wrong input !!!";
    }

       return 0 ;
}
```

Output:-

1. Sum

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1
********************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
********************************************************************************
Enter your choice(1 , 2 , 3 or 4)::1
The sum of the matrix::

2 2
2 2
C:\Users\harsh\Desktop>
```

2. Product

Command Prompt

```
Enter the number of columns::3
Enter the elements at index [0][0]::2
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
Enter the elements at index [1][2]::1
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::1
Enter the elements at index [2][2]::1
The matrix you entered::

2 2 1
1 1 1
1 1 1

*****Enter the details of second matrix*****

Enter the number of row::3
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
The matrix you entered::

1 1
1 2
1 2
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::2
The product of the matrix::

5 8
3 5
3 5
C:\Users\harsh\Desktop>
```

3. Transpose

Command Prompt

```
Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::3
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [1][2]::3
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
Enter the elements at index [2][2]::3
The matrix you entered::

1 2 3
1 2 3
1 2 3

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
The matrix you entered::

1 2
1 2
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::3
**********************************************************************
Which matrix you want to transpose (1 or 2)::1
The matrix after transpose::
1 1 1
2 2 2
3 3 3
C:\Users\harsh\Desktop>
```

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
The matrix you entered::

1 2
1 2

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
The matrix you entered::

1 2
1 2
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::3
**********************************************************************
Which matrix you want to transpose (1 or 2)::2
The matrix after transpose::
1 1
2 2
C:\Users\harsh\Desktop>
```

4. Subtracting

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::4
The subtraction of the matrix::
0 0
0 0
C:\Users\harsh\Desktop>
```

5. Exception handling

Command Prompt

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\harsh>cd desktop

C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::5
Wrong input !!!
C:\Users\harsh\Desktop>
```

Exception in main menu

Command Prompt

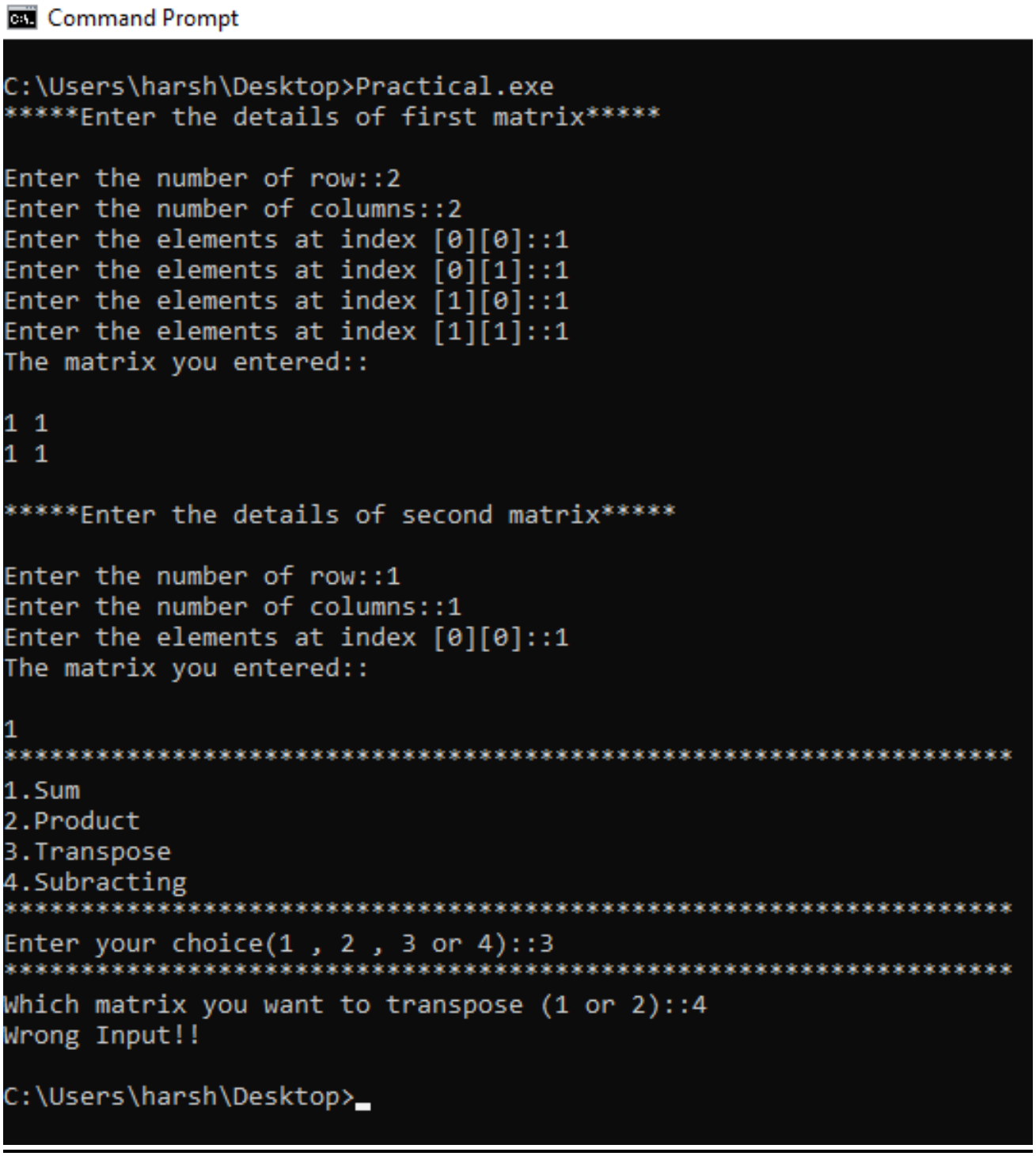
```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::1
Enter the elements at index [0][0]::1
The matrix you entered::

1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::1
Error !! The order of the matrix must be same to perform sum.
C:\Users\harsh\Desktop>
```

Exception in sum

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
The matrix you entered::

1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::2
Error !! The column of the first matrix must be same to the row of the second.
C:\Users\harsh\Desktop>
```

Exception in case of product



Exception in case of transpose

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
The matrix you entered::

1 1
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::4
Error !! The order of the matrix must be same to perform subtraction.
C:\Users\harsh\Desktop>
```

Exception in case of subraction