Practical 6

Name: Saurabh Asnare (SYCOA 276)

```
#include <iostream>
using namespace std;
#define N 4
int flag = 0;
// # Matrix Class
class Matrix
private:
  int rows, col;
  int arr[N][N];
public:
  void acceptMatrix(int);
  void displayMatrix(int);
  void Addition(Matrix, Matrix);
  void Subtraction(Matrix, Matrix);
  void Multiplication(Matrix, Matrix);
  Matrix()
     cout << "Matrix class object
created.." << endl;
     rows = 0;
     col = 0;
  ~Matrix()
     rows = 0;
```

```
col = 0;
   }
};
// *** Function Definations ***
void Matrix ::acceptMatrix(int k)
  int i, j;
  flag = 2;
  cout << "\n# Matrix-" << k << " : " <<
endl;
  cout << "Enter rows: ";</pre>
  cin >> rows:
  cout << "Enter columns: ";</pre>
  cin >> col;
  try
     if (rows == 0 \parallel col == 0 \parallel rows > N \parallel
col > N)
     {
        throw 1;
        cout << "\n^* Enter matrix-" << k
<< " : " << endl;
        for (i = 0; i < rows; i++)
           for (j = 0; j < col; j++)
```

```
cin >> arr[i][j];
          }
        }
     }
  catch (int e)
     cout << "\n* Size Overflow.." <<
endl;
     flag = 1;
  }
}
void Matrix ::displayMatrix(int k)
{
  int i, j;
  cout << "\n# Matrix-" << k << " is: " <<
endl;
  if (flag == 1 || flag == 0)
  {
     cout << "* Please Enter matrix
first..." << endl;
  }
  else
     for (i = 0; i < rows; i++)
     {
       for (j = 0; j < col; j++)
          cout << arr[i][j] << " ";
        }
```

```
cout << endl;
     }
  }
void Matrix ::Addition(Matrix m1, Matrix
m2)
  // Resultant matrix is
  int i, j;
  if (flag == 1 \parallel \text{flag} == 0)
  {
     cout << "\n* Enter Matrix first" <<
endl;
   }
  else
     try
        if (m1.rows != m2.rows | m1.col !=
m2.col)
           throw 1;
           cout << "\n# Resultant matrix is:</pre>
" << endl;
           for (i = 0; i < m1.rows; i++)
             for (j = 0; j < m1.col; j++)
                arr[i][j] = m1.arr[i][j] +
m2.arr[i][j];
```

```
cout << arr[i][j] << " ";
                                                                throw 1;
             }
             cout << endl;
                                                                cout << "\n# Resultant matrix is:</pre>
          }
                                                     " << endl;
        }
                                                                for (i = 0; i < m1.rows; i++)
     }
                                                                  for (j = 0; j < m1.col; j++)
     catch (int e)
                                                                   {
                                                                     arr[i][j] = m1.arr[i][j] -
       cout << "\n* Matrix order should
                                                     m2.arr[i][j];
be same.." << endl;
                                                                     cout << arr[i][j] << " ";
     }
                                                                   }
  }
                                                                  cout << endl;
}
void Matrix ::Subtraction(Matrix m1,
Matrix m2)
                                                          catch (int e)
{
  // Resultant matrix is
                                                             cout << "\n* Order should be
  int i, j;
                                                     same.." << endl;
  if (flag == 1 || flag == 0)
                                                           }
                                                        }
     cout << "\n* Enter Matrix first" <<
endl;
                                                     }
  }
  else
                                                     void Matrix :: Multiplication(Matrix m1,
                                                     Matrix m2)
     try
                                                        int i, j, k, sum;
                                                        if (flag == 1 \parallel flag == 0)
       if (m1.rows != m2.rows | m1.col !=
m2.col)
                                                        {
```

```
cout << "\n* Enter Matrix first" <<
                                                                   cout << arr[i][j] << " ";
endl;
                                                                 }
  }
                                                                 cout << endl;
  else
                                                            }
     try
                                                         catch (int e)
       if (m1.col != m2.rows)
                                                           cout << "\n* You can't multiply.."
          throw 1;
                                                    << endl;
                                                         }
                                                      }
          for (i = 0; i < m1.rows; i++)
                                                    }
          {
             for (j = 0; j < m2.col; j++)
                                                    // # Main Function
                                                    int main()
               sum = 0;
                                                    {
               for (k = 0; k < m1.col; k++)
                                                      cout << endl;
                                                      Matrix m1, m2, m;
                 sum = sum +
                                                      cout << "\n # Matrix Arithmetic #\n";</pre>
(m1.arr[i][k] * m2.arr[k][j]);
                                                      int ch;
                                                      do
               arr[i][j] = sum;
                                                         cout << "\n# Select from below</pre>
                                                    choices: \n1.Accept Matrix \n2.Display
                                                    Matrix \n3.Addition \n4.Subtraction
          printf("\n# Resultant matrix is:
                                                    \n5.Multiplication \n6.Exit \nEnter your
n";
                                                    choice: ";
          for (i = 0; i < m1.rows; i++)
                                                         cin >> ch;
                                                         switch (ch)
            for (j = 0; j < m2.col; j++)
             {
                                                         case 1:
```

```
m1.acceptMatrix(1);
                                                        }
       m2.acceptMatrix(2);
                                                     } while (1);
       break;
                                                     return 0;
                                                   }
     case 2:
                                                  Output:
       m1.displayMatrix(1);
       m2.displayMatrix(2);
                                                   Matrix class object created..
       break;
                                                  Matrix class object created..
                                                  Matrix class object created..
     case 3:
       m.Addition(m1, m2);
                                                   # Matrix Arithmetic #
       break;
                                                  # Select from below choices:
     case 4:
                                                   1.Accept Matrix
       m.Subtraction(m1, m2);
                                                   2.Display Matrix
       break;
                                                   3.Addition
                                                  4.Subtraction
     case 5:
                                                   5. Multiplication
       m.Multiplication(m1, m2);
                                                   6.Exit
       break;
                                                  Enter your choice: 1
     case 6:
                                                  # Matrix-1:
       cout << "\n* You are exit!\n";</pre>
                                                  Enter rows: 2
       cout << endl;
                                                  Enter columns: 2
       exit(0);
       break;
                                                   * Enter matrix-1:
                                                   1
     default:
                                                  2
       cout << "\n* Invalid choice!" <<
                                                   3
endl;
                                                   4
       break;
```

	6.Exit
# Matrix-2 :	Enter your choice: 4
Enter rows: 2	
Enter columns: 2	# Resultant matrix is:
	-4 -4
* Enter matrix-2 :	-4 -4
5	
6	# Select from below choices:
7	1.Accept Matrix
8	2.Display Matrix
	3.Addition
# Select from below choices:	4.Subtraction
1.Accept Matrix	5.Multiplication
2.Display Matrix	6.Exit
3.Addition	Enter your choice: 5
4.Subtraction	
5.Multiplication	# Resultant matrix is:
6.Exit	19 22
Enter your choice: 3	43 50
# Resultant matrix is:	
68	
10 12	
# Select from below choices:	
1.Accept Matrix	
2.Display Matrix	
3.Addition	
4.Subtraction	

5.Multiplication