**Agnim Gupta**

**2028083**

**A23, CSSE**

**Question 1**

**WAP to find sort an integer array and a float array, using function template.**

#include<iostream>

using namespace std;

#define N 10

template <typename T>

void sort(T arr[], int SIZE)

{

    for (int i = 0; i < SIZE; i++)

    {

        for (int j = i+1; j < SIZE; j++)

        {

            if (arr[i] > arr[j])

            {

                T temp;

                temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

    }

}

int main()

{

    int int\_array[N];

    float float\_array[N];

    cout<<"Enter integer array elements:"<<endl;

    for (int i = 0; i < N; i++)

    {

        cin>>int\_array[i];

    }

    cout<<"Entner floating array elements:"<<endl;

    for (int i = 0; i < N; i++)

    {

        cin>>float\_array[i];

    }

    sort(int\_array,N);

    sort(float\_array, N);

    cout<<"After sorting they are :"<<endl;

    for (int i = 0; i < N; i++)

    {

        cout<<int\_array[i]<<", ";

    }

    cout<<endl;

    for (int i = 0; i < N; i++)

    {

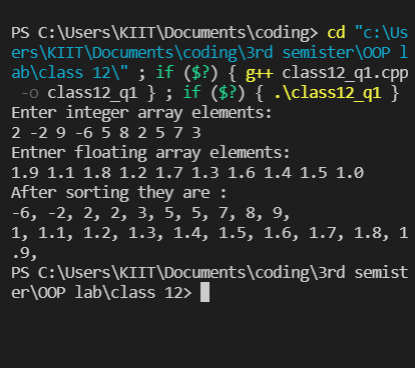
        cout<<float\_array[i]<<", ";

    }

    return 0;

}

**Output**

****

**Question 2**

**WAP to display data of two different types using function template with multiple arguments.**

#include<iostream>

using namespace std;

template<class T1, class T2>

class Test

{

        T1 a;

        T2 b;

    public:

        Test(T1 x, T2 y)

        {

            a = x;

            b = y;

        }

        void show()

        {

            cout << a << " and " << b << endl;

        }

};

int main()

{

    Test <float, int> test1 (9.99999, 998);

    Test <int, char> test2 (99, 't');

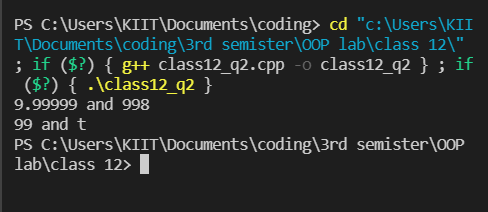
    test1.show();

    test2.show();

    return 0;

}

**Output**

****

**Question 3**

**Rewrite program i. using class template**

#include <iostream>

using namespace std;

const int N = 7;

template <class Type>

class Array{

    private:

        Type arr[N];

    public:

        void read(){

            for(int i = 0; i < N; i++){

                cin>>arr[i];

            }

        }

        void sortArr(){

            Type temp;

            int SIZE = sizeof(arr)/sizeof(Type);

            for(int i = 0; i < SIZE - 1; i++){

                for(int j = i + 1; j < SIZE; j++){

                    if(arr[i] < arr[j]){

                        temp = arr[i];

                        arr[i] = arr[j];

                        arr[j] = temp;

                    }

                }

            }

        }

        void display(){

            int SIZE = sizeof(arr)/sizeof(Type);

            for(int i = 0; i < SIZE; i++){

                cout<<arr[i]<<" ";

            }

            cout<<endl;

        }

};

int main(){

    Array <int> i\_obj;

    Array <float> f\_obj;

    cout<<"Enter integer array:";

    i\_obj.read();

    cout<<"Enter floating number array:";

    f\_obj.read();

    i\_obj.sortArr();

    f\_obj.sortArr();

    cout<<"Sorted integer array:"<<endl;

    i\_obj.display();

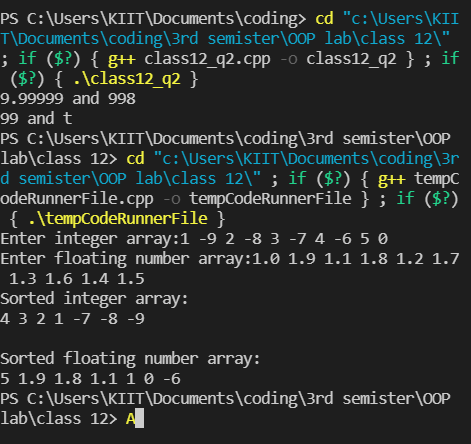
    cout<<endl<<"Sorted floating number array:"<<endl;

    f\_obj.display();

    return 0;

}

**Output**

****

**Question 4**

**Rewrite program ii. using class template**

#include<iostream>

using namespace std;

template<class T1, class T2>

class Test

{

        T1 a;

        T2 b;

    public:

        Test(T1 x, T2 y)

        {

            a = x;

            b = y;

        }

        void show()

        {

            cout << a << " and " << b << endl;

        }

};

int main()

{

    Test <float, int> test1 (9.9999, 997);

    Test <int, char> test2 (5599, 't');

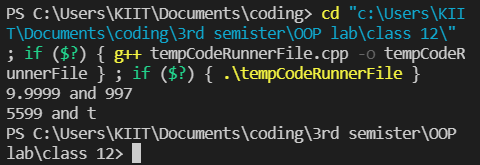
    test1.show();

    test2.show();

    return 0;

}

**Output**

****

**Question 5**

**Write a C++ program using function template to read two matrices of different data types such as integers and floating point values and find the sum of the matrices of integers and floating point numbers separately, and display the total sums of these arrays individually.**

#include <iostream>

#include <iomanip>

int const size = 3;

using namespace std;

template < class M, class S>

    void add(M a[][size], S b[][size])

    {

        S c[size][size];

        for (int i = 0; i < size; i++)

            for (int j = 0; j < size; j++)

            {

                c[i][j] = a[i][j] + b[i][j];

            }

        for (int i = 0; i < size; i++)

        {

            for (int j = 0; j < size; j++)

            {

                cout << "\t" << c[i][j];

            }

            cout << endl;

        }

    }

int main()

{

    int x[size][size], y[size][size];

    float g[size][size], h[size][size];

    int ch;

    cout << endl << "Enter values for Integer Matrix (Only Integers): " << endl;

    for (int i = 0; i < size; i++)

    {

        for (int j = 0; j < size; j++)

        {

            cin >> x[i][j];

        }

    }

    cout << endl << "Enter values for Float Matrix (Only Float): " << endl;

    for (int i = 0; i < size; i++)

    {

        for (int j = 0; j < size; j++)

        {

            cin >> g[i][j];

        }

    }

    cout << endl << endl << "The Integer Matrix by the user:" << endl;

    for (int i = 0; i < size; i++)

    {

        for (int j = 0; j < size; j++)

        {

            cout << x[i][j] << "  ";

        }

        cout << endl;

    }

    cout << endl << endl << "The Float Matrix by the user:" << endl;

    for (int i = 0; i < size; i++)

    {

        for (int j = 0; j < size; j++)

        {

            cout << g[i][j] << "  ";

        }

        cout << endl;

    }

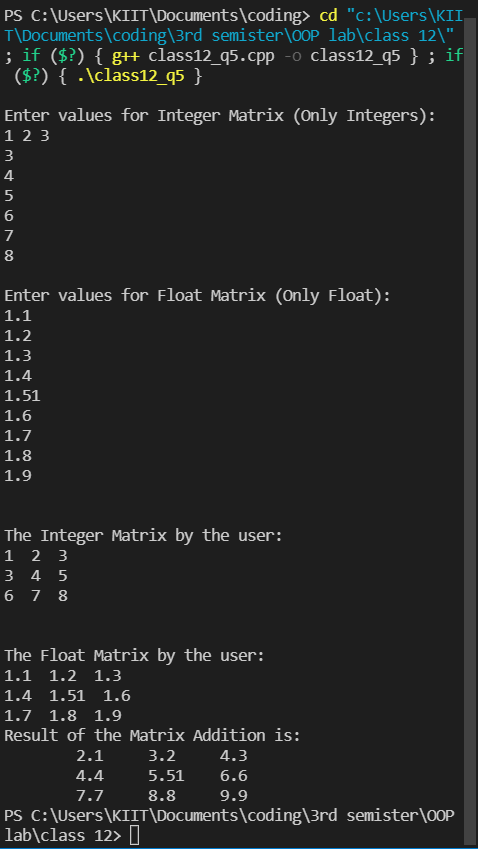
    cout << "Result of the Matrix Addition is: " << endl;

    add(x, g);

    return 0;

}

**Output**

****

**Question 6**

**WAP to throw and handle ‘division by zero’ exception.**

#include <iostream>

using namespace std;

float Division(float numerator, float denominator)

{

   if (denominator == 0)

   {

      throw runtime\_error("Math error: Attempted to divide by Zero\n");

   }

   return (numerator / denominator);

}

int main()

{

   float numerator, denominator,quotient;

   cout << "Enter the numerator: ";

   cin >> numerator;

   cout << "Enter the denominator: ";

   cin >> denominator;

   try

   {

      quotient = Division(numerator, denominator);

      cout << "The quotient is " <<quotient << endl;

   }

   catch (runtime\_error &e)

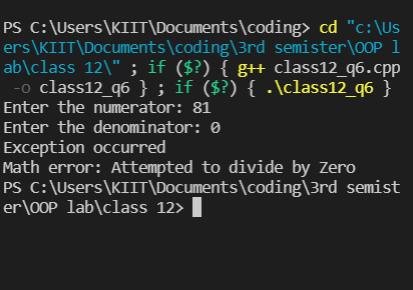
   {

      cout << "Exception occurred" << endl << e.what();

   }

}

**Output**

****

**Question 7**

**WAP to throw and handle’ array out of bound’ exception.**

#include<iostream>

using namespace std;

int main(){

int n;

cout<<"enter size"<<endl;

cin>>n;

int arr[n];

cout<<"enter the values"<<endl;

try

{

    for(int i=0;i<=n;i++)

    {

        if(i<n)

        cin>>arr[i];

        else

        throw i;

    }

}

catch (int a)

{

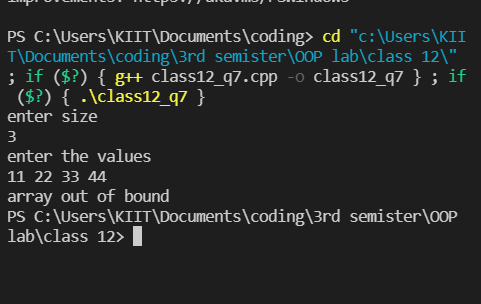
    cout<<"array out of bound"<<endl;

}

return 0;

}

**Output**

****

**Question 8**

**WAP to demonstrate multiple catch and catch all**

#include<iostream>

using namespace std;

void test(int a)

{

    try{

    if(a==1)

      throw a;

    else if (a==0)

      throw 'a';

    else if (a==-1)

      throw 1.0;

    cout << "End of try block";

    }

    catch(char c)

    {

    cout<<"Caught an Character \n";

    }

    catch(int m)

    {

    cout<<"Caught an integer \n";

    }

    catch(double d)

    {

    cout<<"Caught a float \n";

    }

    cout<<"End of Try -Catch block";

}

int main()

{

test(1);

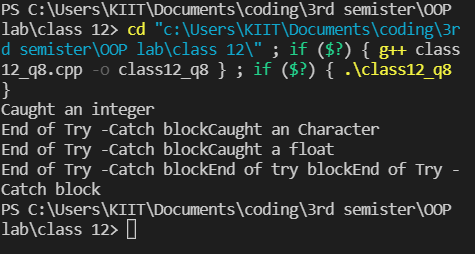
test(0);

test(-1);

test(2);

}

**Output**

****