**Agnim Gupta**

**2028083**

**A23, CSSE**

**Question 1**

//1. Write a program that throws and catches an integer exception.

// Handle the exception and print its value.

#include <iostream>

using namespace std;

int main()

{

    int n{0};

    cout<<"n: ";

    cin>>n;

    try

    {

        if(n<5)

        {

            throw n;

        }

    }

    catch(int x)

    {

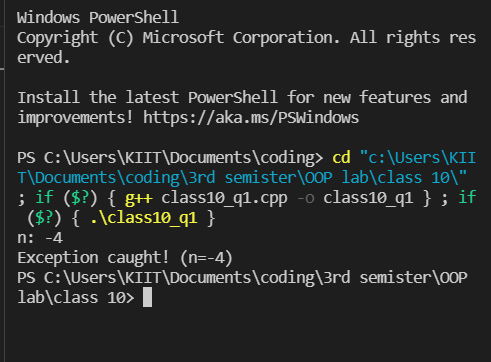
        printf("Exception caught! (n=%d)\n", x);

    }

    return 0;

}

**Output**

****

**Question 2**

//2. Write a program that can throw integer and double exceptions in the same try block.

//Implement the exception handling blocks for both exceptions.

#include <iostream>

using namespace std;

int main()

{

    int a;

    double b;

    cin >> a;

    cin >> b;

    try

    {

        cout << "In try block " << endl;

        if (a != 0)

            cout << 5 / a;

        else

            throw (a);

        if (b != 0.0)

            cout << 5.0 / b;

        else

            throw (b);

    }

    catch (int a)

    {

        cout << "exception caught of int data type" << endl;

    }

    catch (double b)

    {

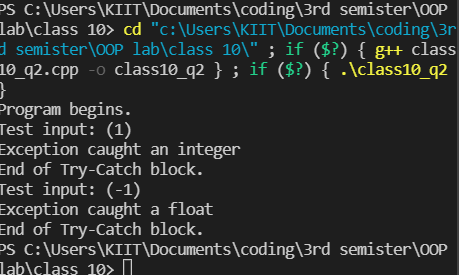
        cout << "exception caught of double data type" << endl;

    }

    return 0;

}

**Output**

****

**Question 3**

//3. Write a program to sort an array of integers using function pointer in descending

//order and resort this array in ascending order using virtual function.

#include <iostream>

using namespace std;

class base

{

public:

    void disOrder(int \*arr, int size)

    {

        cout << "Descending Order of base class\n";

        int a;

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

        {

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

            {

                if (arr[i] < arr[j])

                {

                    a = arr[i];

                    arr[i] = arr[j];

                    arr[j] = a;

                }

            }

        }

    }

    virtual void asOrder(int \*arr, int size)

    {

        cout << "Virtual function for Ascending Order of base class\n";

    }

};

class derived : public base

{

public:

    void asOrder(int \*arr, int size)

    {

        cout << "Ascending Order of derived class\n";

        int a;

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

        {

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

            {

                if (arr[i] > arr[j])

                {

                    a = arr[i];

                    arr[i] = arr[j];

                    arr[j] = a;

                }

            }

        }

    }

    void disOrder(int \*arr, int size)

    {

        cout << "Descending Order of derived class\n";

    }

};

int main()

{

    base \*a;

    int arr[5] = {1, 2, 3, 4, 5};

    derived b;

    a = &b;

    a->disOrder(arr, 5);

    for (int i = 0; i < 5; i++)

        cout << arr[i];

    cout << "\n";

    a->asOrder(arr, 5);

    for (int i = 0; i < 5; i++)

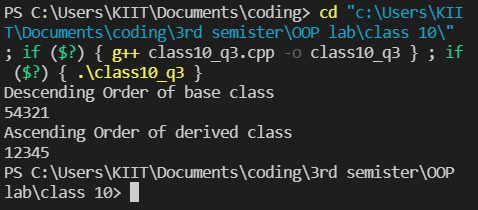
        cout << arr[i];

    cout << "\n";

    return 0;

}

**Output**

****

**Question 4**

//4. Write a program to accept 10 integers in an array . Check all numbers in the array.

//When any negative number is found , throw an exception.

#include <iostream>

using namespace std;

int main()

{

    int input[10], n;

    try

    {

        cout << "How many elements do you want to input? ";

        cin >> n;

        string s = "Array out of bounds Exception Caught";

        if(n > 10)

        {

            throw (s);

        }

        else

        {

            cout << "Input elements\n";

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

            {

                cin >> input[i];

            }

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

            {

                if (input[i] < 0)

                {

                    throw (input[i]);

                }

            }

        }

    }

    catch (int i)

    {

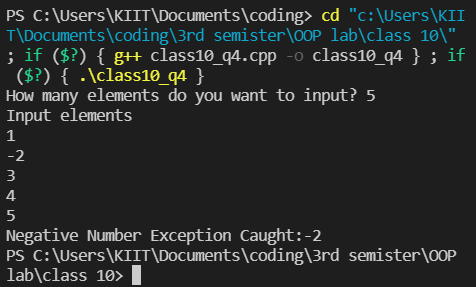
        cout << "Negative Number Exception Caught:"<<i;

    }

    return 0;

}

**Output**

****