Program-1

#include <iostream>

using namespace std;

template <typename T> void SortArray(T \*arr, int n=10){

// Bubble Sorting

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

for (int j=i; j<n; j++){

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

T temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

template <typename T> void Display(T \*arr, int n=10){

cout << "\n--- Displaying ---";

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

cout << arr[i] << " ";

}

}

int main()

{

int arr1[] = {2,4,3,9,1,6,8,7,5,0};

float arr2[] = {1.2, 8.4, 8.43, 8.39, 1.1};

cout <<"----- Before Soring -----";

Display(arr1);

Display(arr2, 5);

SortArray(arr1);

SortArray(arr2, 5);

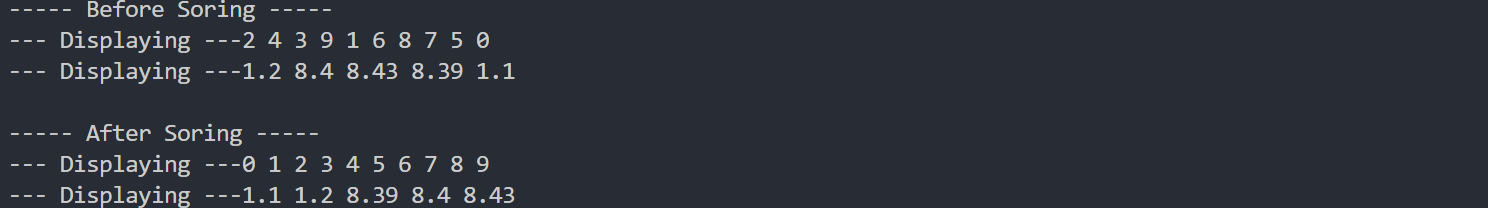
cout <<"\n\n----- After Soring -----";

Display(arr1);

Display(arr2, 5);

return 0;

}



Program-2

#include <iostream>

#include <vector>

using namespace std;

template <typename T1, typename T2> void Display(T1 a, T2 b){

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

}

int main (){

int i = 8;

float f = 13.56;

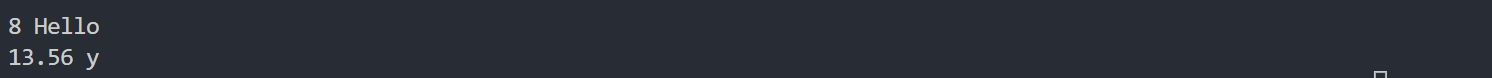
char c = 'y';

string s = "Hello";

Display(i, s);

Display(f, c);

}



Program-3

#include <iostream>

using namespace std;

// Class template

template <class T>

class MyClass {

private:

int size;

T arr[];

public:

MyClass(int n, T a[]){

size = n;

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

arr[i] = a[i];

}

}

void SortArray(){

// Bubble Sorting

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

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

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

T temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

void Display(){

cout <<endl;

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

cout << arr[i] << " ";

}

}

};

int main()

{

int arr1[10] = {3,1,9,2,8,11,7,6,4,5};

MyClass<int> obj1(10, arr1);

float arr2[4] = {7.4, 7.2, 7.9, 7.6};

MyClass<float> obj2(4, arr2);

cout <<"----- Before Soring -----";

obj1.Display();

obj2.Display();

obj1.SortArray();

obj2.SortArray();

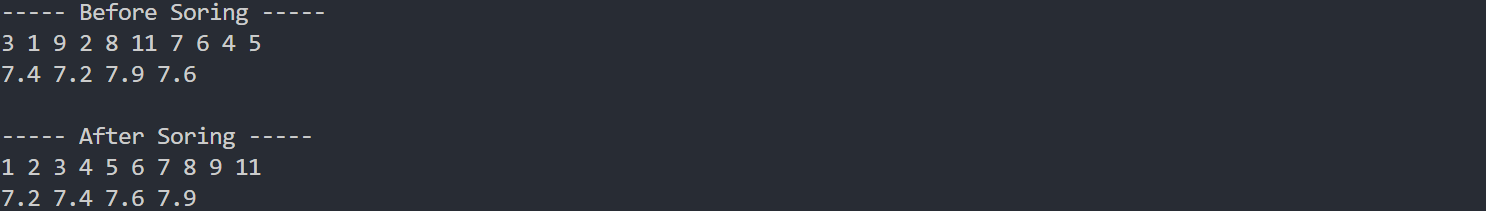
cout <<"\n\n----- After Soring -----";

obj1.Display();

obj2.Display();

return 0;

}



Program-4

#include <iostream>

using namespace std;

void func(){

float a, b;

cout << "\nEnter value for a and b : ";

cin >> a >> b;

cout << a << "/" << b << " = ";

try{

// if condition used to get a custom output otherwise div by 0

// gives inf because new C++ compiler already handles the exception

if (b==0){

cout << "[Exception occurred]";

return;

}

cout << a/b ;

}

// unused code because new C++ compliers handle the exception by itself

catch (exception e) {

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

}

}

int main(){

func();

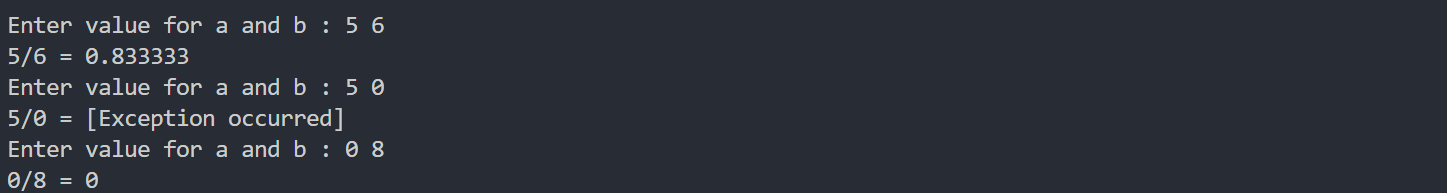
func();

func();

return 0;

}

Output



Program-5

#include <iostream>

using namespace std;

int main() {

int arr[4] = {1,3,5,2};

for (int i=-2; i<8; i++){

try{

cout << arr[i] << " ";

}

catch (exception e){

cout << "[Array out of bounds EXCEPTION]";

e.what();

break;

}

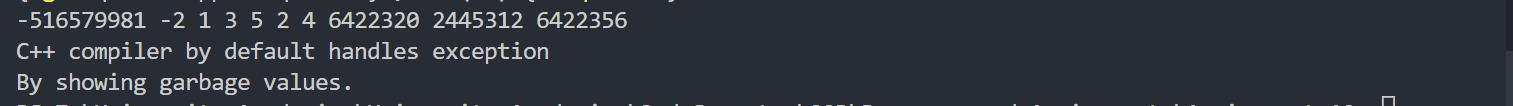
}

cout << "\nC++ compiler by default handles exception";

cout << "\nBy showing garbage values.";

}

**Output**

****