**SUBMITTED BY :**

**NAME : ZISHNENDU SARKER**

**ROLL NO.: 2K19/CO/450**

**SUBJECT : OOP LAB**

**LAB ASSIGNMENT NO. : 04**

**SUBMITTED TO :**

**DIKSHA RUHELA MAM**

**Program - 10**

1. **Write a program containing a possible exception. Use a try block to throw it and a catch block to handle it promptly.**

#include<iostream>

#include<cmath>

#define pi 3.1416

using namespace std;

void power\_factor(float a){

if(a>1 || a<-1)

throw(a);

else{

cout<<"Voltage(V)is lagging from current(I) by "

<<acos(a)\*180/pi<<" degree\n";

}

}

int main(){

float a;

try{

cout<<" Enter power factor ";

cin>>a;

power\_factor(a);

}

catch(float b){

cout<<" Caught an exception \n";

}

return 0;

}

1. **Write a program that illustrates the application of multiple catch statements.**

#include<iostream>

#define size 5

using namespace std;

void multiple\_catch(int n){

float v[size];

try{

if(n>size)

throw(n);

else if(n<=0)

throw(0.01);

else{

cout<<"Enter "<<n<<" elements one by one \n";

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

cin>>v[i];

cout<<"Now contents of v["<<n<<"]: \n";

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

cout<<v[i]<<" ";

}

cout<<endl;

}

catch(int m){

cout<<"Array size must be less than or equal "

<<size<<" \n";

}

catch(double c){

cout<<"Array size must be positive except 0 \n";

}

}

int main(){

int s;

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

cin>>s;

multiple\_catch(s);

return 0;

}

1. **Write a program which uses the catch(…) handler.**

#include<iostream>

#define size 5

using namespace std;

void multiple\_catch(int n){

float v[size];

try{

if(n>size)

throw(n);

else if(n<=0)

throw(0.01);

else{

cout<<"Enter "<<n<<" elements one by one \n";

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

cin>>v[i];

cout<<"Now contents of v["<<n<<"]: \n";

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

cout<<v[i]<<" ";

}

cout<<endl;

}

catch(...){

cout<<"Exception is detected \n";

}

}

int main(){

int s;

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

cin>>s;

multiple\_catch(s);

return 0;

}

1. **Write a program that demonstrates how certain exception types are not allowed to be thrown.**

#include<iostream>

using namespace std;

void empty() throw(){

cout<<"In empty()\n";

}

void with\_type( float x) throw(int){

if(x==1)

throw(1);

else if(x==1.1)

throw (2.1);

}

int main(){

try{

empty();

with\_type(1);

}

catch (int n){

cout<<"Caught an int = "<<n;

}

catch(float){

cout<<"Caught a float ";

}

cout<<endl;

return 0;

}

1. **Write a program to demonstrate the concept of re-throwing an exception.**

#include<iostream>

using namespace std;

void division(int a,int b){

try{

if(b==0)

throw b;

else

cout<<" a/b = "<<(float)a/b<<"\n";

}

catch(int){

cout<<" Caught an exception as first throwing \n";

throw;

}

}

int main(){

int a,b;

cout<<" Enter the value of a & b : ";

cin>>a>>b;

try{

division(a,b);

}

catch(int){

cout<<" Caught an exception as rethrowing \n";

}

return 0;

}

1. **Write a program with the following:**
2. **A function to read two double type integers from the keyboard.**
3. **A function to calculate the division of these two numbers.**
4. **A try block to throw an exception when a wrong type of data is keyed in.**
5. **A try block to detect and throw an exception if the condition “divide by zero” occurs.**
6. **Appropriate catch block to handle the exception thrown.**

#include<iostream>

using namespace std;

class santo{

double a;

double b;

public:

void input(double x,double y){a=x;b=y;}

void division();

};

void santo::division(){

try{

if(b==0)

throw b;

else

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

}

catch(int){

cout<<"An exception is caught \n";

}

}

int main(){

double d;

int m;

santo hasibul;

double m1,n1;

cout<<"Enter two number : ";

cin>>m>>n1;

try{

if(sizeof(d)!=sizeof(m) || sizeof(d)!=sizeof(n1))

throw n1;

else{

hasibul.input(m,n1);

hasibul.division();

}

}

catch(double){

cout<<"Caught an exception"<<endl;

}

return 0;

}

1. **Write a main program that calls a deeply nested function containing an exception. Incorporate necessary exception handling mechanisms.**

#include<iostream>

using namespace std;

long int square(int i){

return i\*i;

}

long int sum(int n){

long int s;

s=0;

for(int i=1;i<=n;i++)

s+=square(i);

return s;

}

void display(int m){

try{

if(m<0)

throw m;

else

cout<<sum(m)<<"\n";

}

catch(int n){

cout<<"Caught an exception"<<endl;

}

}

int main(){

int n;

cout<<"Enter a positive number ";

cin>>n;

display(n);

return 0;

}