Question 1:

/\*

Name: Eranus Thompson

Working with exceptions on a quadractic equation

\*/

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class DivisionByZero {

public:

string what() {

return message;

}

DivisionByZero() {

message = "attempt to divide by zero";

}

private:

string message;

};

class NegativeSqrRoot {

public:

string what() {

return message;

}

NegativeSqrRoot() {

message = "attempt to square root a negative number";

}

private:

string message;

};

class OneRepeatedRoot {

public:

string what() {

return message;

}

OneRepeatedRoot() {

message = "both roots are the same";

}

private:

string message;

};

void quadEquation(double a, double b, double c, double& x1, double& x2) {

/\*

pre condition : a != 0, ((b\*b) - 4 \* (a \* c) > 0, x1 != x2

post condition : throws a DivisionByZero exception if a == 0

throws a NegativeSqrRoot exception if

((b\*b) - 4 \* (a \* c) < 0

throws a OneRepeatedRoot exception if x1 == x2

\*/

if (a == 0) {

throw DivisionByZero();

}

else if (((b\*b) - 4 \* (a \* c)) < 0) {

throw NegativeSqrRoot();

}

else {

x1 = (-b + (sqrt((b\*b) - 4 \* (a \* c)))) / (2 \* a);

x2 = (-b - (sqrt((b\*b) - 4 \* (a \* c)))) / (2 \* a);

if (x1 == x2) {

throw OneRepeatedRoot();

}

}

}

int main() {

double a, b, c, x1 = 0, x2 = 0;

try {

cout << "Enter a value for A, B, and C: ";

cin >> a >> b >> c;

cout << endl;

quadEquation(a, b, c, x1, x2);

}

catch (DivisionByZero except) {

cout << except.what();

}

catch (NegativeSqrRoot except) {

cout << except.what();

}

catch (OneRepeatedRoot except) {

cout << except.what() << " \nroot1 = "

<< x1 << endl;

}

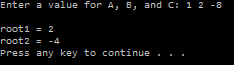
cout << endl;

return 0;

}

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose of the test | Input (enter a number) | Expected Output | Actual Output |
| Prints root1 and root2 | A = 1, B = 2, C = -8 | Root1 = 2  Root2 = -4 | Root1 = 2  Root2 = -4 |
| Throws a DivisionByZero exception | A = 0, B = 2, C = -8 | Attempt to divide by zero | Attempt to divide by zero |
| Throws a NegativeSqrRoot exception | A = 1, B = 4, C = 5 | Attempt to square root a negative number | Attempt to square root a negative number |
| Throws a OneRepeatedRoot exception | A = 1, B = -4, C = 4 | Both roots are the same | Both roots are the same |

Test Case 1:



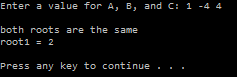
Test Case 2:



Test Case 3:



Test Case 4:



Question 2:

/\*

Name: Eranus Thompson

Working with exceptions

\*/

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class OddNumber {

public:

string what() {

return message;

}

OddNumber() {

message = "The integer is an Odd Number\n";

}

private:

string message;

};

class IsPower2 {

public:

string what() {

return message;

}

IsPower2() {

message = "The integer is a power of 2\n";

}

private:

string message;

};

class Negative {

public:

string what() {

return message;

}

Negative() {

message = "The integer is a Negative Number\n";

}

private:

string message;

};

void foo(int a);

void bar(int a);

void bell(int a);

bool isPower2(int a);

int main() {

try {

foo(1);

//foo(10);

//foo(-2);

//foo(16);

}

catch (Negative except) {

cout << except.what();

}

catch (OddNumber except) {

cout << except.what();

}

catch (IsPower2 except) {

cout << except.what();

}

return 0;

}

void foo(int a) {

if (a < 0) {

throw Negative();

}

else {

bar(a);

}

}

void bar(int a) {

if ((a % 2) != 0) {

throw OddNumber();

}

else {

bell(a);

}

}

void bell(int a) {

if (isPower2(a)) {

throw IsPower2();

}

else {

cout << "Ordinary Number\n";

}

}

bool isPower2(int a) {

if (a <= 0) {

return false;

}

return ((a > 0) && (a&(a - 1)) == 0);

}

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose of the test | Input (enter a number) | Expected Output | Actual Output |
| Negative exception | -2 | The integer is a Negative Number | The integer is a Negative Number |
| Odd Number Exception | 1 | The integer is an Odd Number | The integer is an Odd Number |
| Is Power 2 exception | 16 | The integer is a power of 2 | The integer is a power of 2 |
| Ordinary Number | 10 | Ordinary Number | Ordinary Number |

Test Case 1:

Test Case 2:

Test Case 3:

Test Case 4: