**Task 1:**

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

void quadEquation(double a, double b, double c) {

string finderror;

if (a == 0){

finderror += " Cannot divide by zero";

}

else if (a < 0){

finderror += "A cannot be negative";

}

if (b\*b <= 4 \* a\*c) {

finderror += "B can not be smaller than 4AC";

}

if(!finderror.empty())

throw runtime\_error(finderror);

}

void quadEquation(double a, double b, double c);

int main() {

/\*

pre : a != 0, b\*b - 4\*a\*c >= 0

post: catch error before excuted

\*/

double a, b, c;

double input1, input2;

try{

cout << "Input values for A, B and C: " << endl;

cin >> a >> b >> c;

quadEquation(a, b, c);

input1 = (-b - sqrt((b \* b) - (4 \* a \* c))) / (2 \* a);

input2 = (-b + sqrt((b \* b) - (4 \* a \* c))) / (2 \* a);

cout << "Frist root: " << input1 << endl;

cout << "Second root: " << input2 << endl;

}

catch(exception & er)

{

cout << "Issue Found:\n" << er.what() << endl;

}

return 0;

}

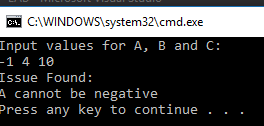
**Executable module instructions:**

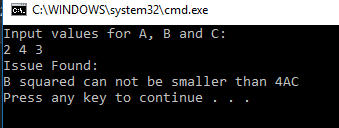
1. **Compile**
2. **Run**

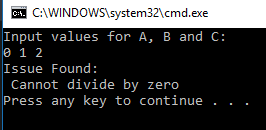
**Test data and expected results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Type** | **Input Data** | **Expected Output** |
| 1 | Negative | -1 4 10 | A cannot be negative |
| 2 | B < 4ac | 2 4 3 | B squared can not be smaller than 4ac |
| 3 | Zero | 0 1 2 | Cannot divide by zero |

**Output:**







**Task 2:**

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class Negative {

public:

string what() {

return message;

}

Negative() {

message = "Negative Number\n";

}

private:

string message;

};

class OddNumber {

public:

string what() {

return message;

}

OddNumber() {

message = "Odd Interger\n";

}

private:

string message;

};

class Power2 {

public:

string what() {

return message;

}

Power2() {

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

}

private:

string message;

};

void foo(int a);

void bar(int a);

void bell(int a);

bool Power2a(int a);

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 (Power2a(a)) {

throw Power2();

} else {

cout << "Ordinary Number\n";

}

}

bool Power2a(int a) {

if (a <= 0) {

return false;

}

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

}

int main() {

try {

foo(10);

//foo(-4);

//foo(69);

//foo(4);

} catch (Negative except) {

cout << except.what();

} catch (OddNumber except) {

cout << except.what();

} catch (Power2 except) {

cout << except.what();

}

return 0;

}**Executable module instructions:**

1. **Compile**
2. **Run**

**Test data and expected results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Type** | **Input Data** | **Expected Output** |
| 1 | Ordinary Number | 10 | Ordinary Number |
| 2 | Odd Integer | 69 | Odd Integer |
| 3 | Negative Integer | -4 | Negative integer |
| 4 | Power of 2 | 4 | The integer is a power of 2 |

**Output:**

