**Exception Handling**

1. Write a c++ program to demonstrate to use of the finally block for handling exceptions

#include <iostream>

using namespace std;

class FinalBlock {

public:

~FinalBlock() {

cout << "Finally block executed." << endl;

}

};

int main() {

try {

FinalBlock f;

throw 100;

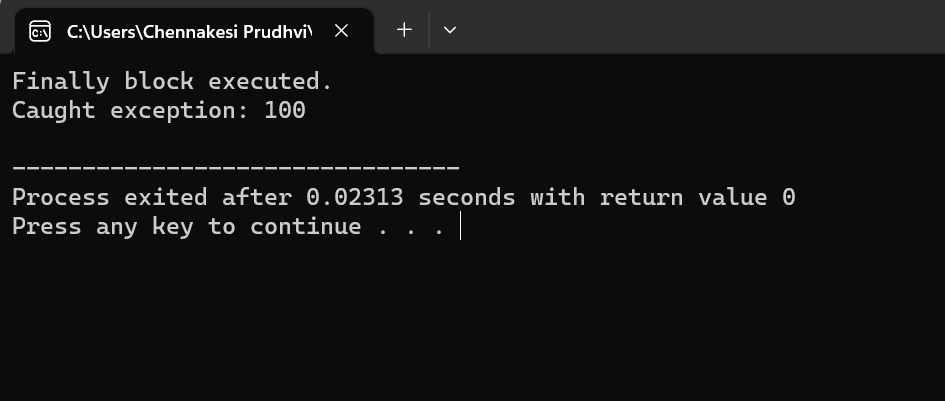
} catch (int e) {

cout << "Caught exception: " << e << endl;

}

return 0;

}



1. Write a c++ program to demonstrate to use of nested try-catch blocks for handling exceptions

#include <iostream>

using namespace std;

int main() {

try {

try {

throw "Inner exception";

} catch (const char\* msg) {

cout << "Caught inner exception: " << msg << endl;

throw; // Re-throw to outer block

}

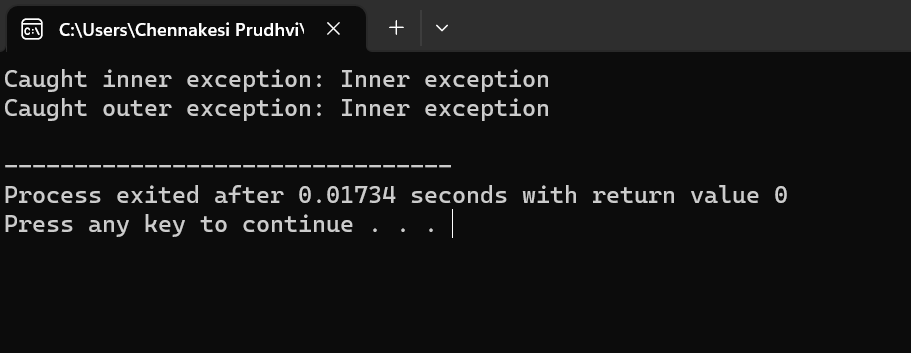
} catch (const char\* msg) {

cout << "Caught outer exception: " << msg << endl;

}

return 0;

}



1. Write a c++ program to demonstrate to use of user-defined exception for handling custom exception

#include <iostream>

using namespace std;

class MyException : public exception {

public:

const char\* what() const noexcept override {

return "Custom exception occurred!";

}

};

int main() {

try {

throw MyException();

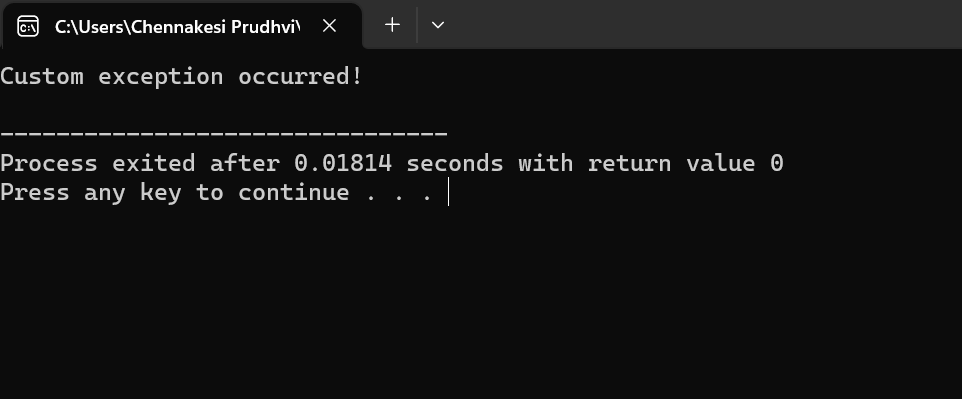
} catch (const MyException& e) {

cout << e.what() << endl;

}

return 0;

}



1. Write a c++ program to demonstrate to use of the standard class for handling exceptions

#include <iostream>

#include <stdexcept>

using namespace std;

int main() {

try {

throw out\_of\_range("Out of range exception");

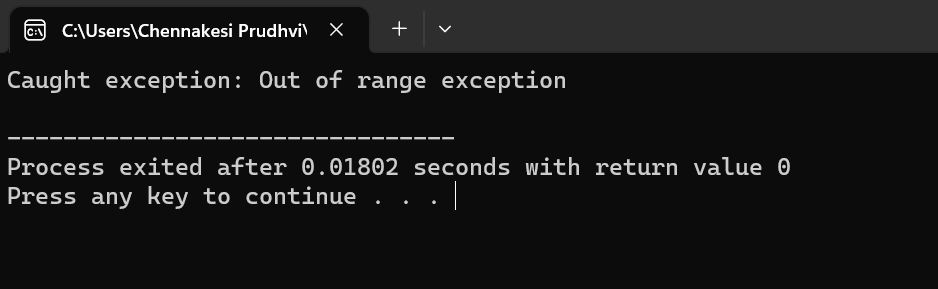
} catch (const exception& e) {

cout << "Caught exception: " << e.what() << endl;

}

return 0;

}



1. Write a c++ program to demonstrate to use of the keyword to throw an exception

#include <iostream>

using namespace std;

void checkNumber(int num) {

if (num < 0) {

throw "Negative number not allowed!";

}

}

int main() {

try {

checkNumber(-5);

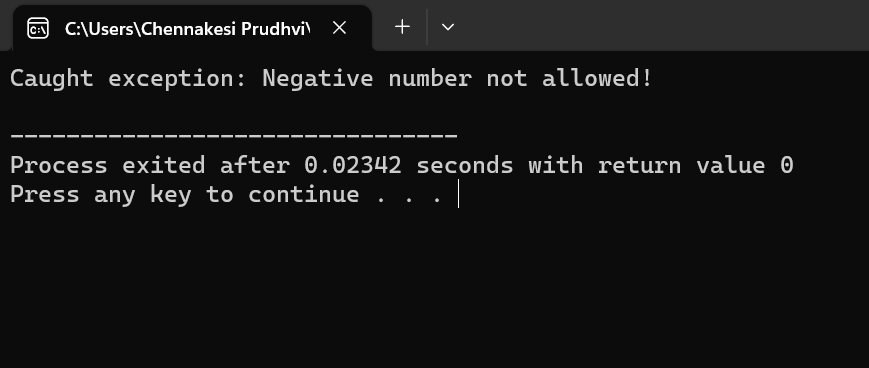
} catch (const char\* msg) {

cout << "Caught exception: " << msg << endl;

}

return 0;

}



1. Write a c++ program to demonstrate to use of multiple catch blocks for handling different types of exceptions

#include <iostream>

using namespace std;

int main() {

try {

throw 20.5;

} catch (int e) {

cout << "Caught integer exception: " << e << endl;

} catch (double e) {

cout << "Caught double exception: " << e << endl;

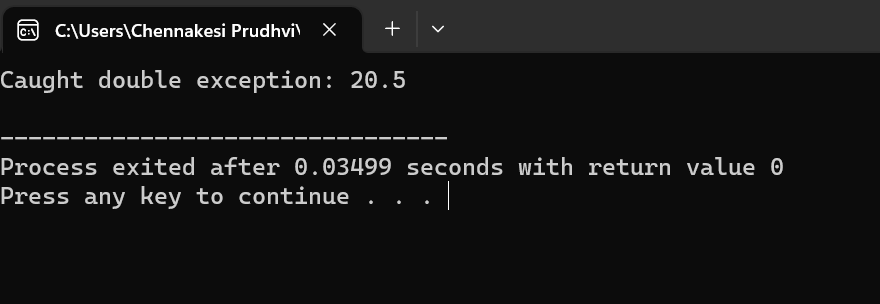
} catch (...) {

cout << "Caught unknown exception" << endl;

}

return 0;

}



1. Write a c++ program to demonstrate to use of try-catch blocks for handling exceptions

#include <iostream>

using namespace std;

int main() {

try {

throw 10;

} catch (int e) {

cout << "Caught exception: " << e << endl;

}

return 0;

}

