# EXPERIMENT NUMBER –Practical 8.1

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

TOPIC OF EXPERIMENT –

**WAP to perform exception handling for Divide by zero Exception.**

AIM OF THE EXPERIMENT

**Learn how to use exception handling C ++.**

FLOWCHART/ ALGORITHM

Start.

Step 1→ Creating a header file for input output stream and define the context.

# Step 2 → After that used using namespace std;

Step 3 → Creating the function division of return type double.

Step 4→ Defining the function with two integers as parameter to find the division of two numbers and finding the exception of division by zero (0).

Step 5 → Passing input at the time of calling (call by value) through try block.

Step 6 → Initializing a catch block to catch the thrown exception.

Step 7→- Printing the results.

STOP.

PROGRAM CODE

#include <iostream>

using namespace std;

double division(int a, int b)

{

if( b == 0 )

{

cout<<"==| smita shinde uid- 20BCS4643 |=="<<endl;

throw "Division by zero condition!";

}

return (a/b);

}

int main () {

int x = 100;

int y = 0;

double z = 0;

try {

z = division(x, y);

cout<< z <<endl;

} catch (const char\* msg) {

cerr<<msg<<endl; }

return 0;

}

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

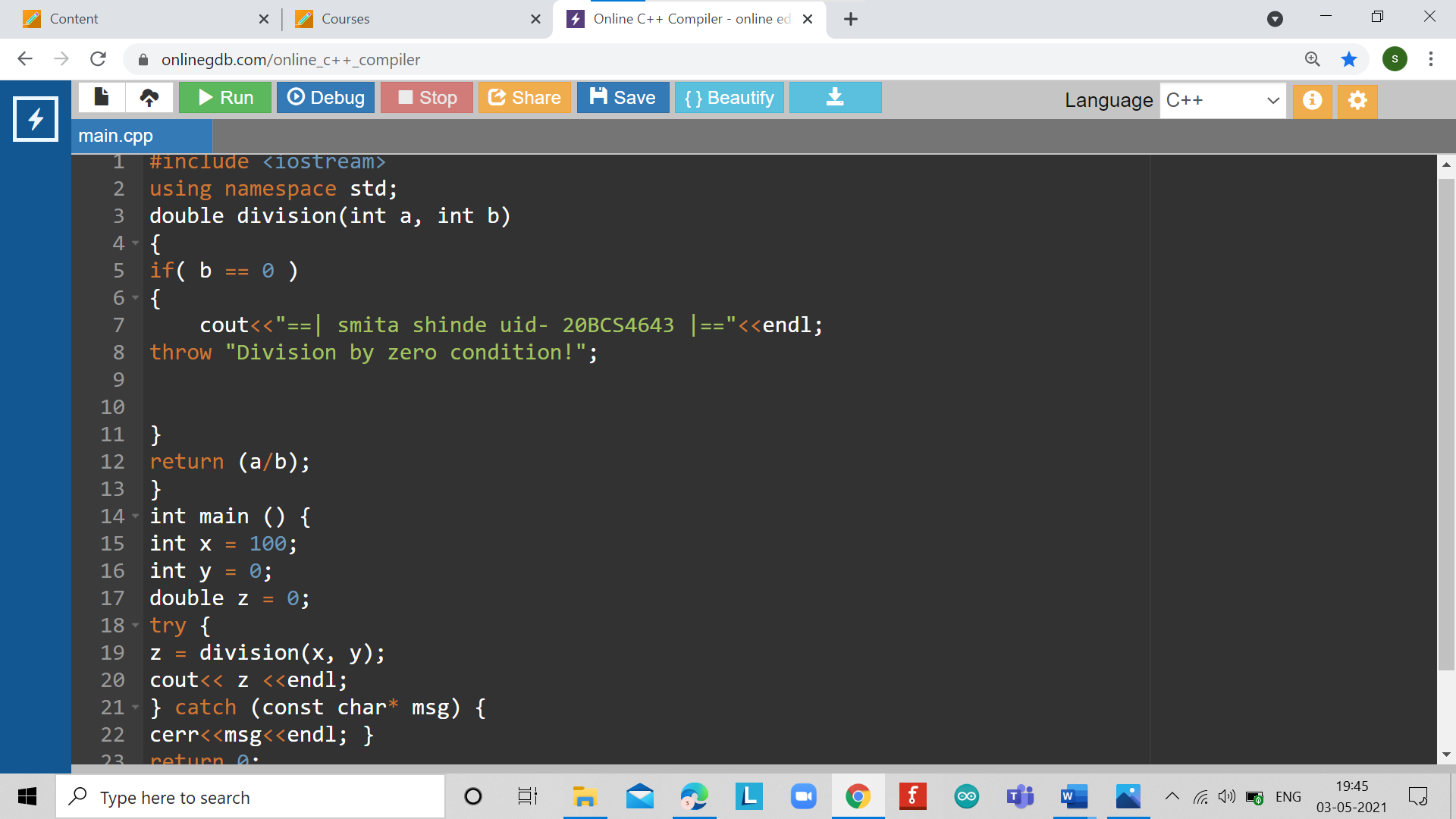
(Kindly jot down the compile time errors encountered)

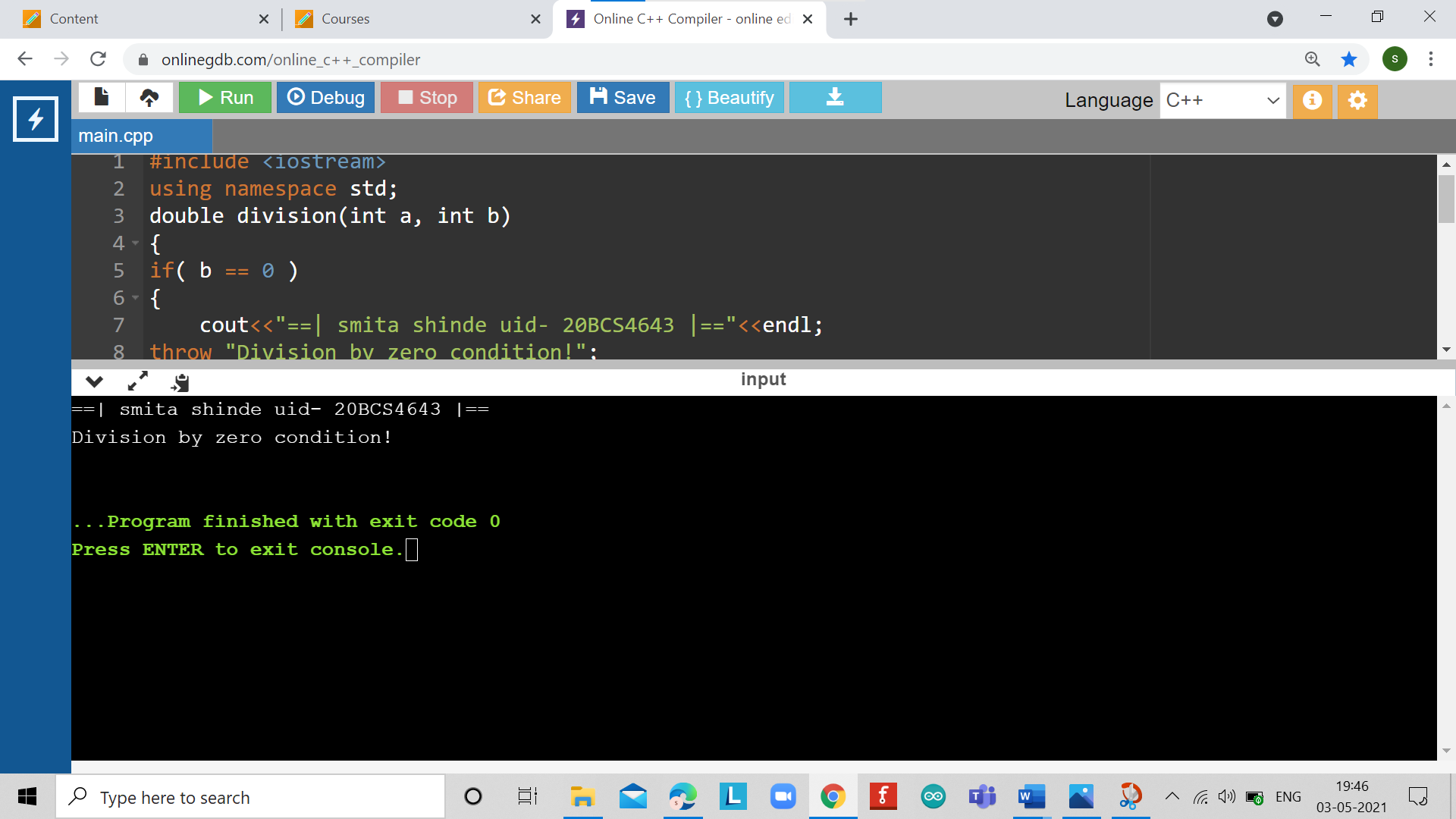
No error occurred while execution of the program.

PROGRAMS’ EXPLANATION (in brief)

1. We start our program with preprocessor (#) and header file (< iostream >) we have many types of header files but in this program, we used iostream. #include< iostream >, Int main () is a function which work as a container of statements. All the statements are enclosed within the pair of braces { }. “using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things with in “std” namespace.
2. In this program we have initialized a function named division of return type double which take two Integer as arguments and check whether the denominator is equals to zero or not. If it is equal to zero then it will throw an exception. Then in the main function we have call that function with in a try block with call by value method of function calling. If the exception thrown it is caught by the catch block otherwise terminate without entering the catch block.

OUTPUT





# EXPERIMENT NUMBER –Practical 8.2

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

TOPIC OF EXPERIMENT –

WAP to implement the exception handling with the functionality of testing the throw restrictions**.**

AIM OF THE EXPERIMENT

**Learn how to use exception handling C ++.**

FLOWCHART/ ALGORITHM

Start.

Step 1→ Creating a header file for input output stream and define the context.

# Step 2 → After that used using namespace std;

Step 3 → Creating the function handle of return type as void.

Step 4 → Defining the function with one integer as parameter and use if conditions to throw the different type of exceptions as int, char and of type double.

Step 5 → Defining one int type of catch block and one all catch block in the handle function.

Step 6 → Passing different type of inputs at the time of calling (call by value) through main function.

Step 7 →getting an output on screen.

Stop.

PROGRAM CODE

#include <iostream>

#include <conio.h>

using namespace std;

void handle(int test)

{

try {

if (test==0)

throw test; // throw int

if (test==1)

throw 'a'; // throw char

if (test==2)

throw 123.23; // throw double

}

catch(int i) { // catch an int exception

cout<< "Caught " <<i<< "\n"; }

catch(...) { // catch all other exceptions

cout<< "Caught one!\n"; }

}

int main( ) {

cout<<"== | smita shinde uid-20BCS4643 |=="<<endl;

cout<< "start\n";

handle(0);

handle(1);

handle(2);

cout<< "end";

return 0;

}

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

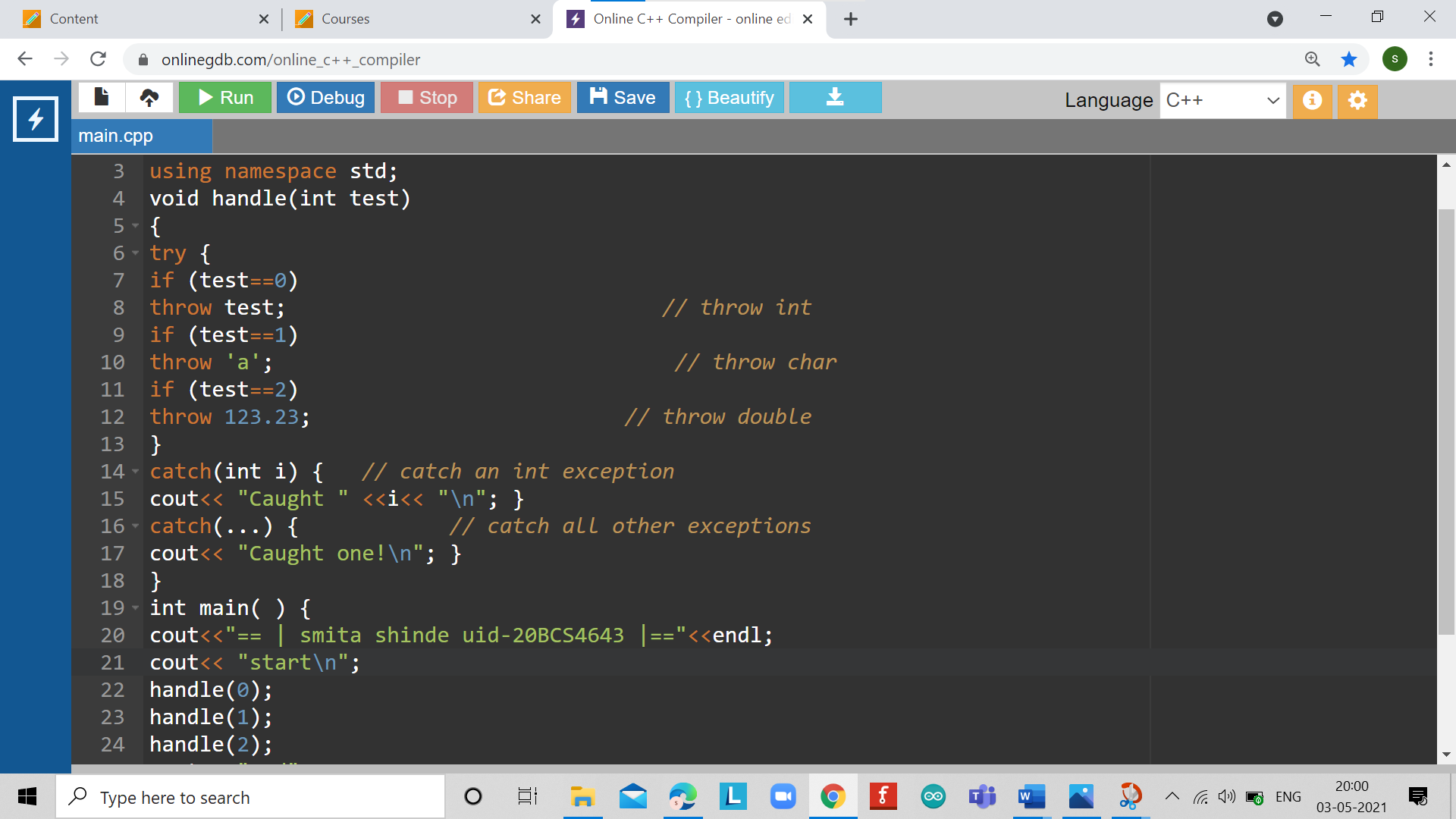
(Kindly jot down the compile time errors encountered)

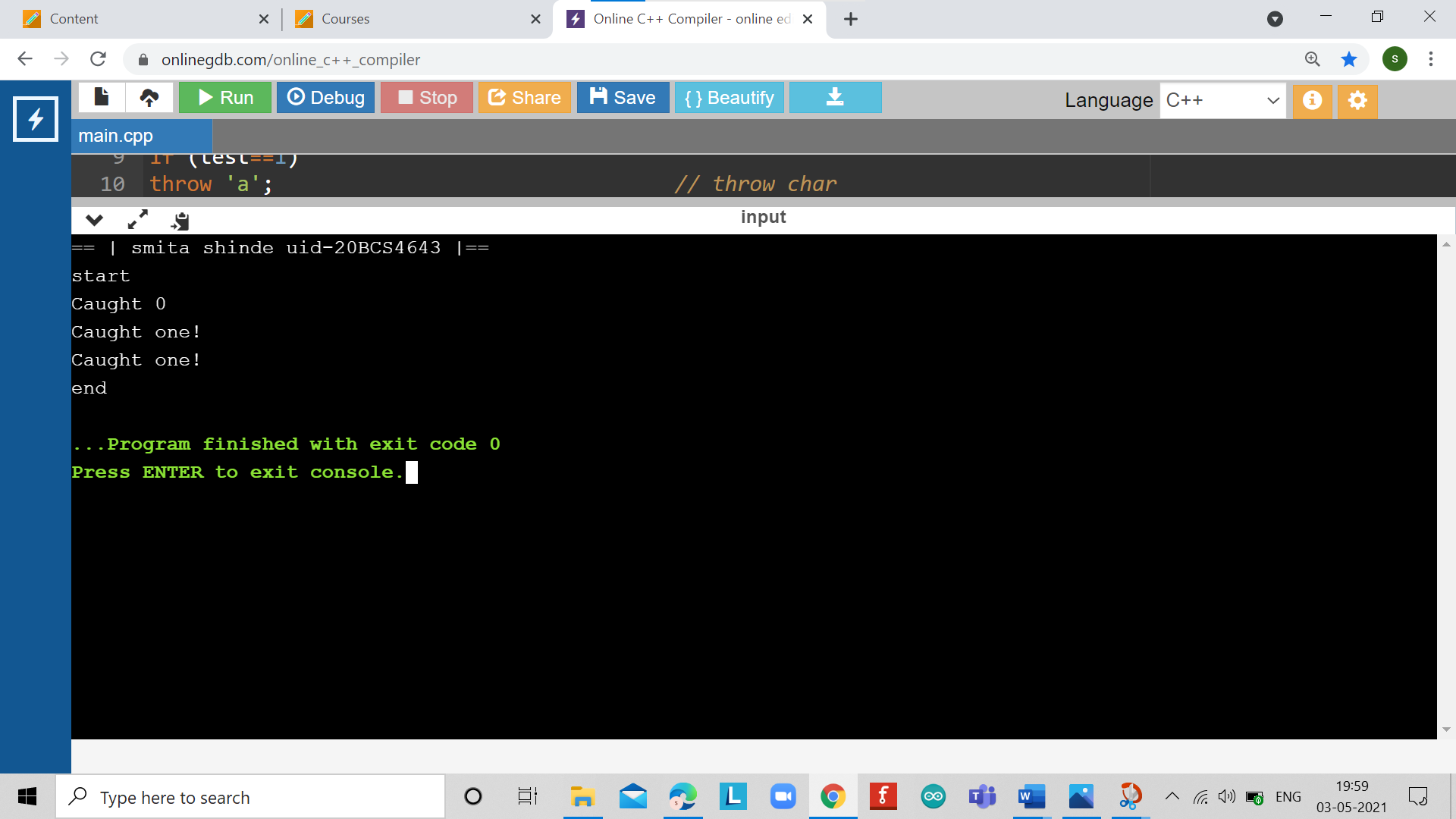
No error occurred while execution of the program.

PROGRAMS’ EXPLANATION (in brief)

1. We start our program with preprocessor (#) and header file (< iostream >) we have many types of header files but in this program, we used iostream. #include< iostream >, Int main () is a function which work as a container of statements. All the statements are enclosed within the pair of braces { }. “using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things with in “std” namespace.
2. In this program we have initialized a function named handle of return type void which take one Integer as arguments and used if statements to throw different types of exceptions and initialized different type of catch blocks in the same function one of type int and one all catch block. Then in the main function we have call that function by using call by value method of function calling. Now in the handle method parsed argument is checked with different if statements to throw different kind of exceptions.

OUTPUT





LEARNING OUTCOMES

|  |
| --- |
| * Identify situations where computational methods would be useful. |
| * Approach the programming tasks using techniques learnt and write pseudo-code. |
| * Choose the right data representation formats based on the requirements of the problem. |
| * Use the comparisons and limitations of the various programming constructs and choose the right one for the task. |

EVALUATION COLUMN (To be filled by concerned faculty only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Parameters** | **Maximum**  **Marks** | **Marks**  **Obtained** |
| 1. | Worksheet Completion including writing learning objective/ Outcome | 10 |  |
| 2. | Post Lab Quiz Result | 5 |  |
| 3. | Student engagement in Simulation/ Performance/ Pre Lab Questions | 5 |  |
| 4. | Total Marks | 20 |  |