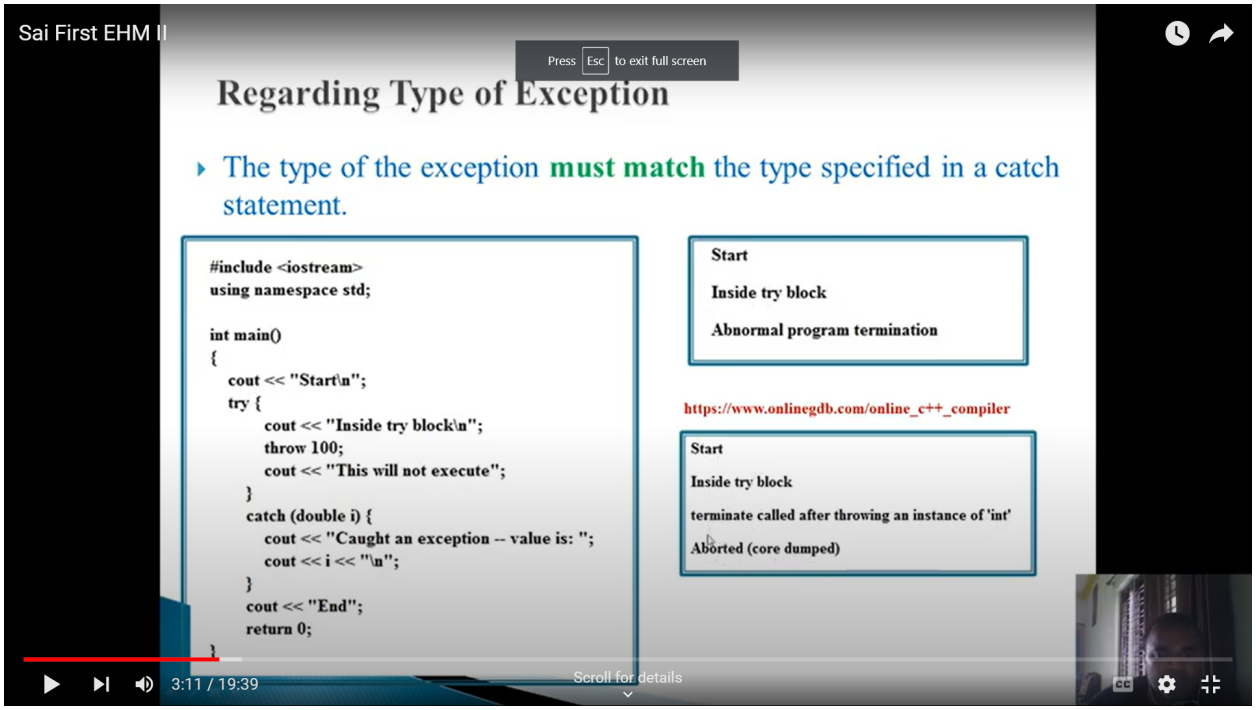
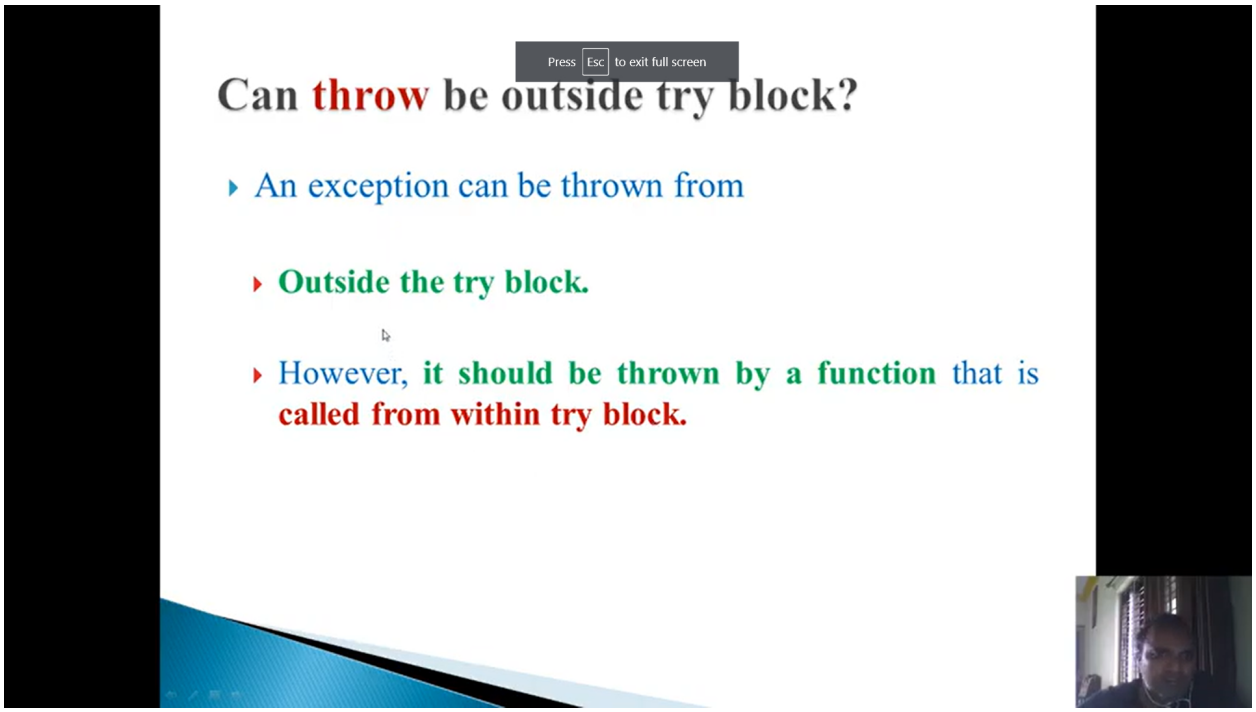
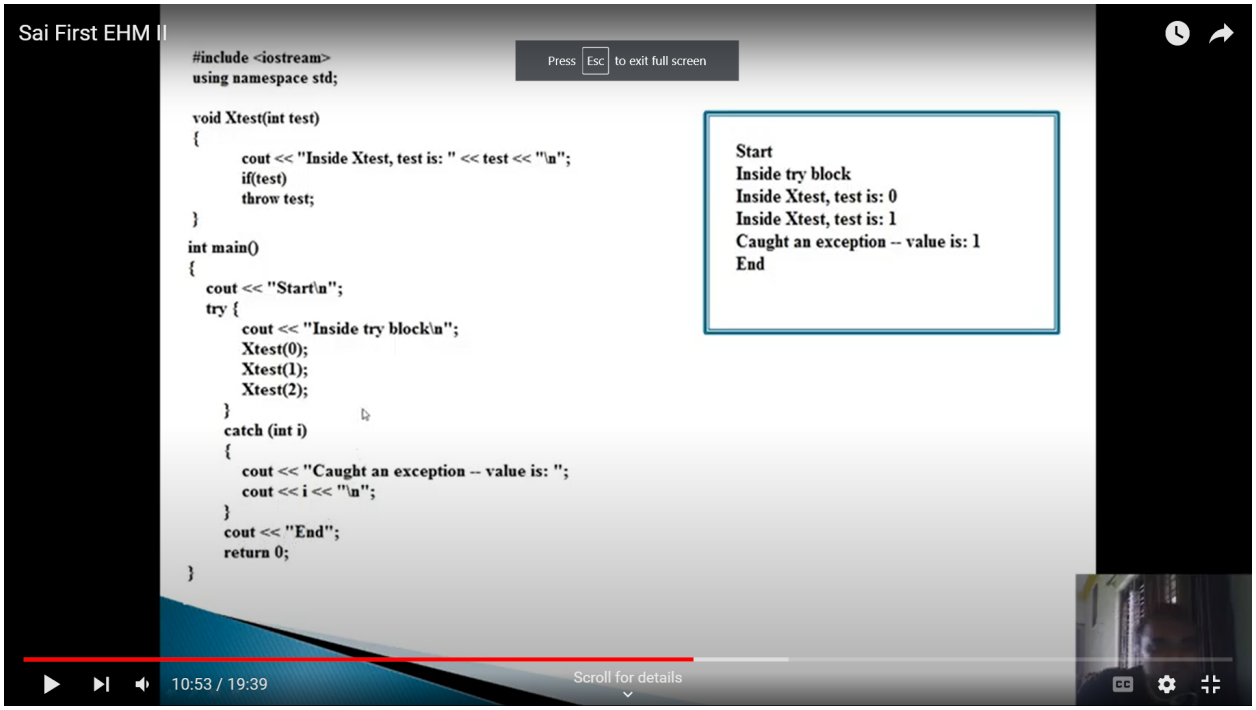
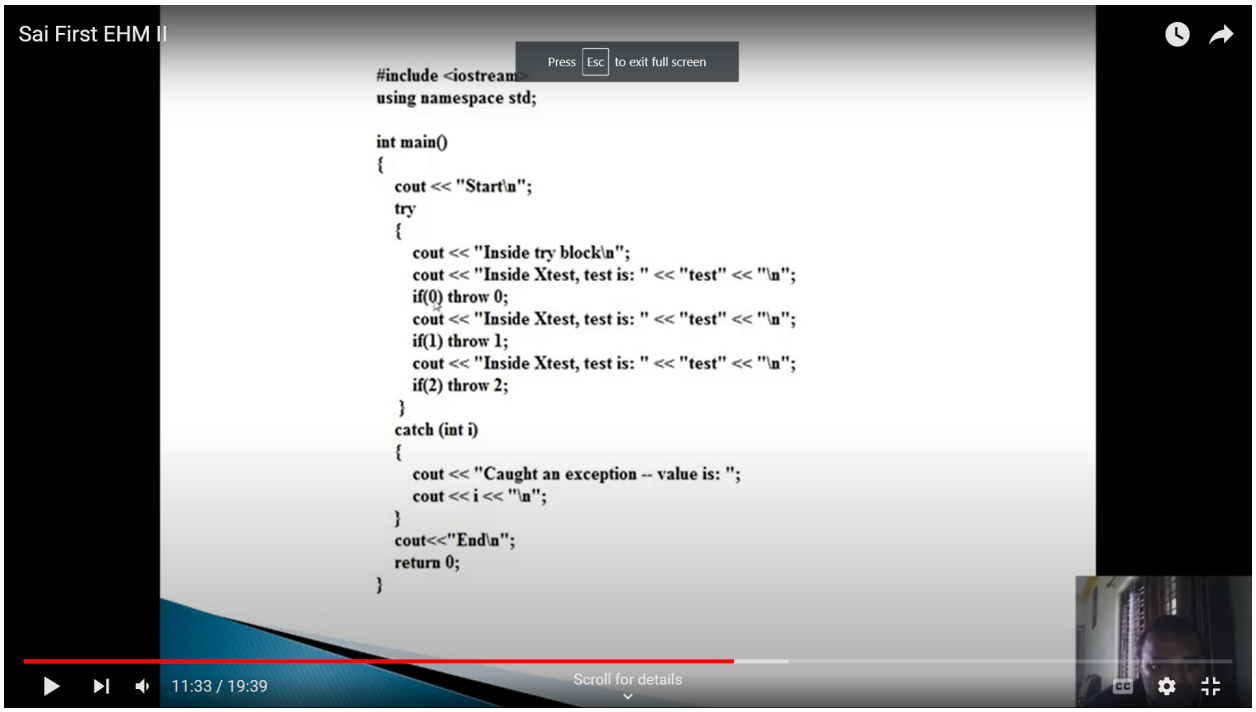
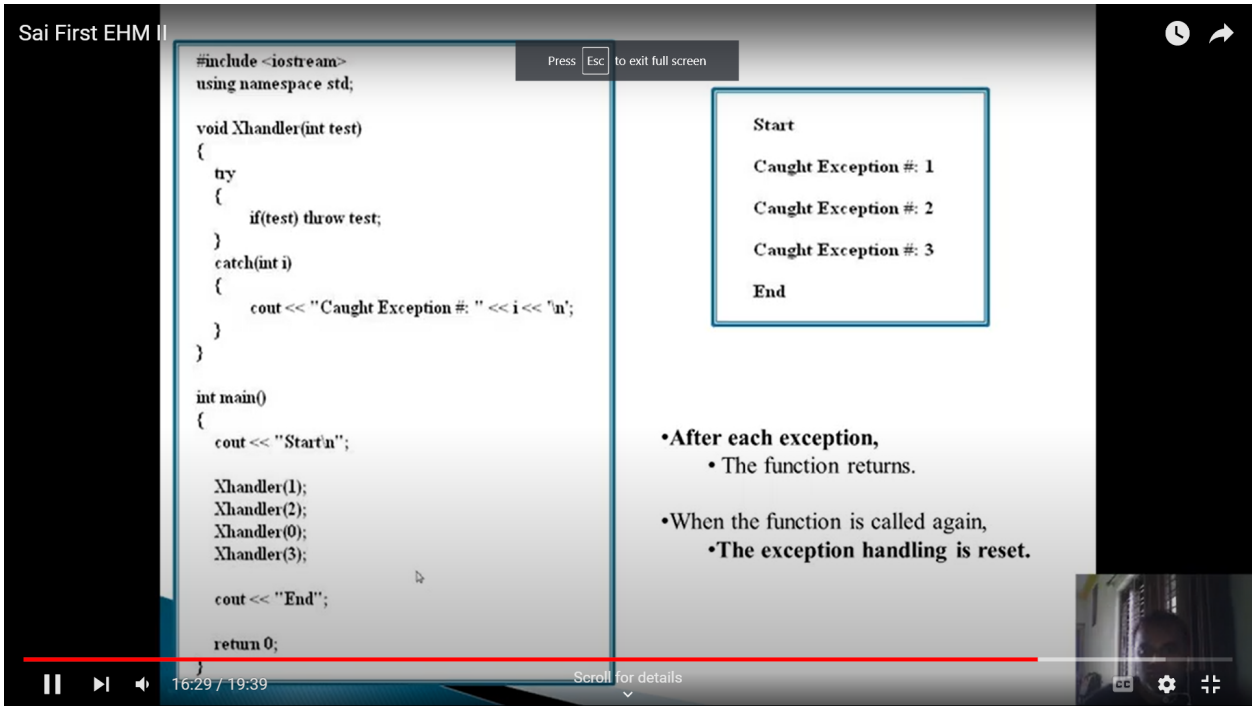
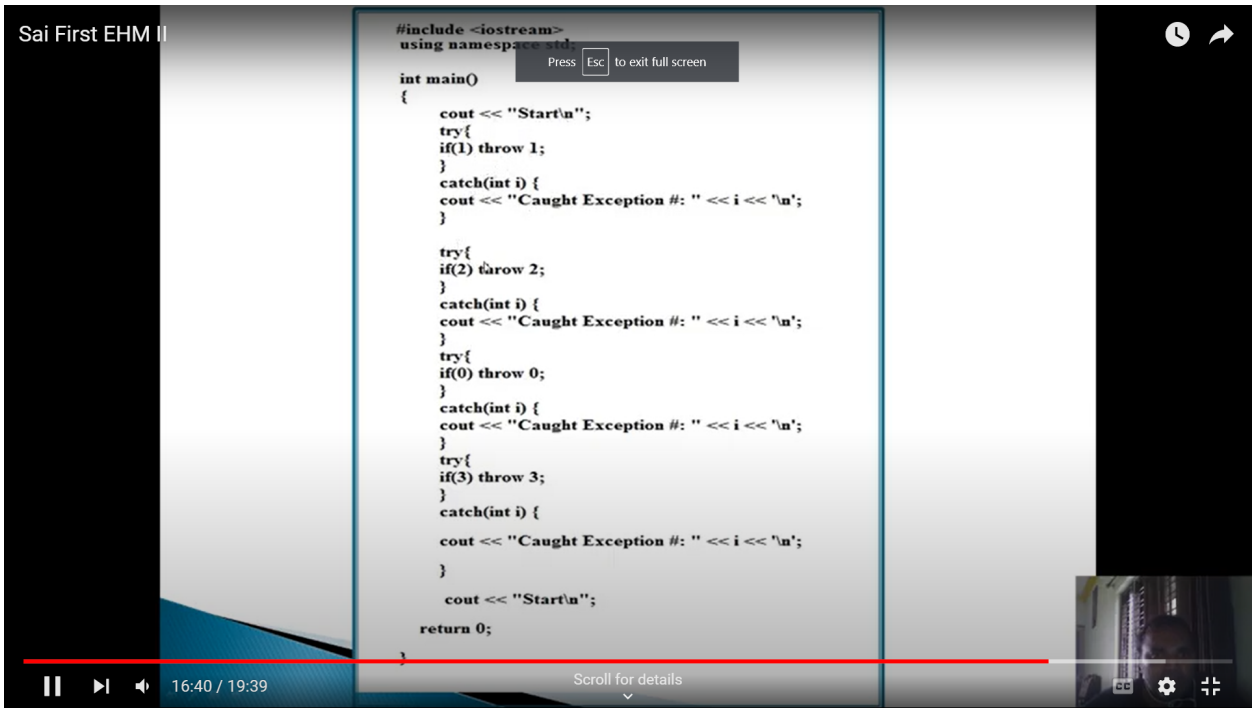

Sai First EHM II
Press Esc to exit full screen
Regarding Type of Exception
The type of the exception must match the type specified in a catch statement.
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try {
cout << "Inside try block\n";
throw 100;
cout << "This will not execute";
}
catch (double i) {
cout << "Caught an exception -- value is: ";
cout << i << "\n";
}
cout << "End";
return 0;
}
Start
Inside try block
Abnormal program termination
https://www.onlinegdb.com/online_c++_compiler
Start
Inside try block
terminate called after throwing an instance of 'int'
Aborted (core dumped)
3:11 / 19:39
Scroll for details

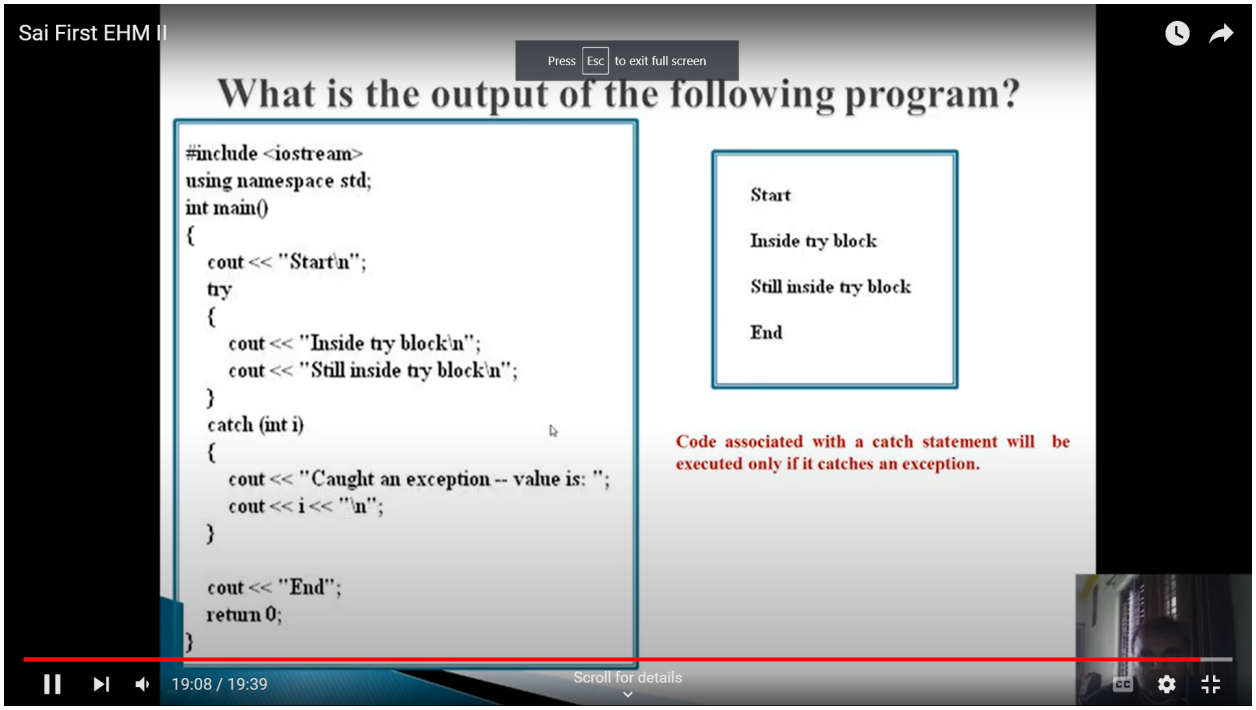

Press Esc to exit full screen
Can throw be outside try block?
An exception can be thrown from
Outside the try block.
However, it should be thrown by a function that is called from within try block.

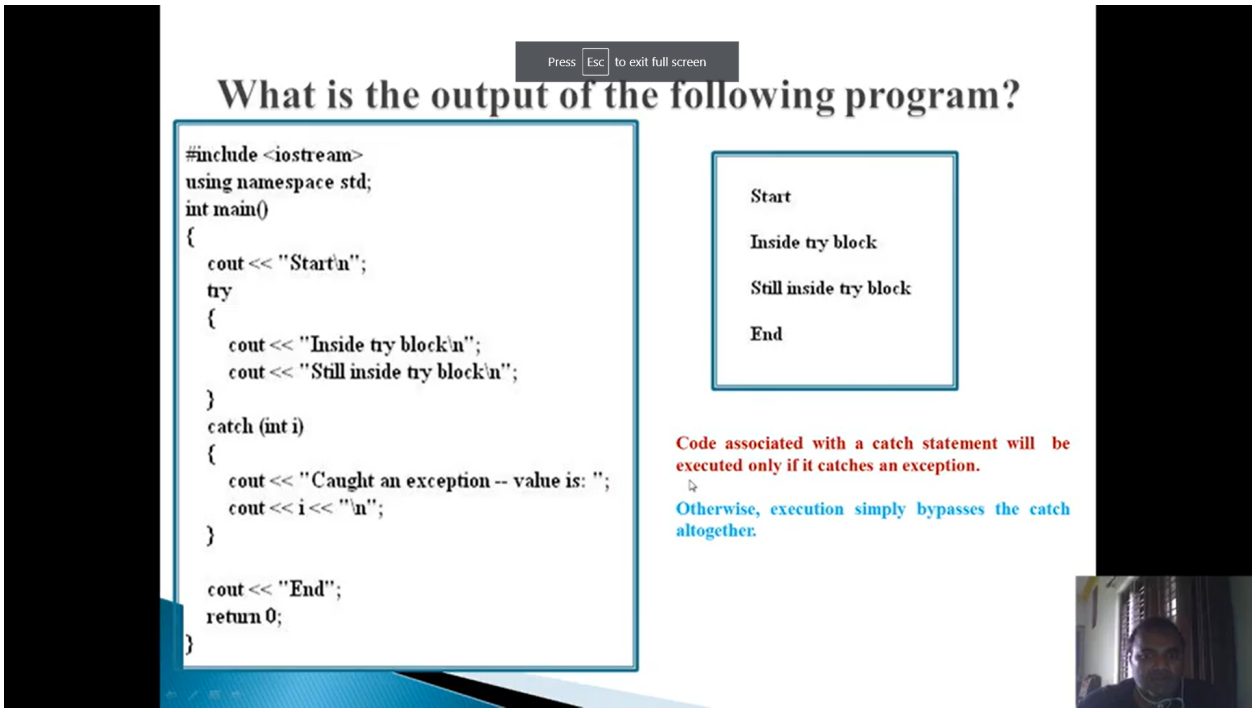

Sai First EHM II
Press Esc to exit full screen
#include <iostream>
using namespace std;
void Xtest(int test)
{
cout << "Inside Xtest, test is: " << test << "\n";
if(test)
throw test;
}
int main()
{
cout << "Start\n";
try {
cout << "Inside try block\n";
Xtest(0);
Xtest(1);
Xtest(2);
}
catch (int i)
{
cout << "Caught an exception -- value is: ";
cout << i << "\n";
}
cout << "End";
return 0;
}
Start
Inside try block
Inside Xtest, test is: 0
Inside Xtest, test is: 1
Caught an exception -- value is: 1
End
10:53 / 19:39
Scroll for details


Sai First EHM II
Press Esc to exit full screen
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try
{
cout << "Inside try block\n";
cout << "Inside Xtest, test is: " << "test" << "\n";
if(0) throw 0;
cout << "Inside Xtest, test is: " << "test" << "\n";
if(1) throw 1;
cout << "Inside Xtest, test is: " << "test" << "\n";
if(2) throw 2;
}
catch (int i)
{
cout << "Caught an exception -- value is: ";
cout << i << "\n";
}
cout<<"End\n";
return 0;
}
11:33 / 19:39
Scroll for details


Sai First EHM II
Press Esc to exit full screen
#include <iostream>
using namespace std;
void Xhandler(int test)
{
try
{
if(test) throw test;
}
catch(int i)
{
cout << "Caught Exception #: " << i << '\n';
}
}
int main()
{
cout << "Start\n";
Xhandler(1);
Xhandler(2);
Xhandler(0);
Xhandler(3);
cout << "End";
return 0;
}
Start
Caught Exception #: 1
Caught Exception #: 2
Caught Exception #: 3
End
•After each exception,
• The function returns.
•When the function is called again,
•The exception handling is reset.
16:29 / 19:39
Scroll for details


Sai First EHM II
Press Esc to exit full screen
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try{
if(1) throw 1;
}
catch(int i) {
cout << "Caught Exception #: " << i << '\n';
}
try{
if(2) throw 2;
}
catch(int i) {
cout << "Caught Exception #: " << i << '\n';
}
try{
if(0) throw 0;
}
catch(int i) {
cout << "Caught Exception #: " << i << '\n';
}
try{
if(3) throw 3;
}
catch(int i) {
cout << "Caught Exception #: " << i << '\n';
}
cout << "Start\n";
return 0;
}
16:40 / 19:39
Scroll for details


Sai First EHM II
Press Esc to exit full screen
What is the output of the following program?
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try
{
cout << "Inside try block\n";
cout << "Still inside try block\n";
}
catch (int i)
{
cout << "Caught an exception -- value is: ";
cout << i << "\n";
}
cout << "End";
return 0;
}
Start
Inside try block
Still inside try block
End
Code associated with a catch statement will be executed only if it catches an exception.
19:08 / 19:39
Scroll for details


Press Esc to exit full screen
What is the output of the following program?
#include <iostream>
using namespace std;
int main()
{
cout << "Start\n";
try
{
cout << "Inside try block\n";
cout << "Still inside try block\n";
}
catch (int i)
{
cout << "Caught an exception -- value is: ";
cout << i << "\n";
}
cout << "End";
return 0;
}
Start
Inside try block
Still inside try block
End
Code associated with a catch statement will be executed only if it catches an exception.
Otherwise, execution simply bypasses the catch altogether.