```
#include <iostream>
using namespace std;

int main()
{
int x = -1;

// Some code
cout << "Before try \n";
try {
        cout << "Inside try \n";
        if (x < 0)
        {
                throw x;
                cout << "After throw (Never executed) \n";
        }
}
catch (int x ) {
        cout << "Exception Caught \n";
}

cout << "After catch (Will be executed) \n";
return 0;
}

#include <iostream>
using namespace std;

int main()
{
        try {
```

```
	throw 10;
	}
	catch (char *excp) {
		cout << "Caught " << excp;
	}
	catch (...) {
		cout << "Default Exception\n";
	}
	return 0;
}
```

```
#include <iostream>
using namespace std;

int main()
{
	try {
	throw 'a';
	}
	catch (int x) {
		cout << "Caught " << x;
	}
	catch (...) {
		cout << "Default Exception\n";
	}
	return 0;
}
```

```
#include <iostream>
using namespace std;
```

```
int main()
{
	try {
	throw 'a';
	}
	catch (int x) {
		cout << "Caught ";
	}
	return 0;
}

#include <iostream>
using namespace std;

// This function signature is fine by the compiler, but not recommended.
// Ideally, the function should specify all uncaught exceptions and function
// signature should be "void fun(int *ptr, int x) throw (int *, int)"
void fun(int *ptr, int x)
{
	if (ptr == NULL)
		throw ptr;
	if (x == 0)
		throw x;
	/* Some functionality */
}

int main()
{
	try {
	fun(NULL, 0);
```

```
	}
	catch(...) {
		cout << "Caught exception from fun()";
	}
	return 0;
}

#include <iostream>
using namespace std;

// Here we specify the exceptions that this function
// throws.
void fun(int *ptr, int x) throw (int *, int) // Dynamic Exception specification
{
	if (ptr == NULL)
		throw ptr;
	if (x == 0)
		throw x;
	/* Some functionality */
}

int main()
{
	try {
	fun(NULL, 0);
	}
	catch(...) {
		cout << "Caught exception from fun()";
	}
	return 0;
}
```

```
#include <iostream>
using namespace std;

int main()
{
	try {
		try {
			throw 20;
		}
		catch (int n) {
			cout << "Handle Partially ";
			throw; // Re-throwing an exception
		}
	}
	catch (int n) {
		cout << "Handle remaining ";
	}
	return 0;
}

#include <iostream>
using namespace std;

class Test {
public:
	Test() { cout << "Constructor of Test " << endl; }
	~Test() { cout << "Destructor of Test " << endl; }
};

int main()
{
```

```
	try {
		Test t1;
		throw 10;
	}
	catch (int i) {
		cout << "Caught " << i << endl;
	}
}


#include <iostream>
using namespace std;
int main()
{
	int att,per;
	cout<<"Enter att\n";
	cin>>att;
	cout<<"Enter per\n";
	cin>>per;
	if((att>=75) && (per>=40))
	{cout<<"Pass\n";
	}
	//else
	//cout<<"Fail\n";
	try
	{
		if(per>40)
		try
		{
		if(att<75)
		throw att;
```

```
                }
                catch(int att )
                {cout<<"Fail\n";
        throw;

                }
        }
                catch(...)
                {cout<<"Increase your att\n";



                }



}
```