

BASICS OF EXCEPTION HANDLING

One of the newer features of C++ is exception (or error) handling. You use the 'try', 'catch', and 'throw' keywords for this purpose. When programming, you tell the compiler to 'try' a section of code. If an error occurs in the section (or in any functions in the section), you 'throw' the exception. The exception is then handled like a hot potato and is passed up the ladder until it is caught. If an exception is not caught, the program terminates. Here's the basic syntax*:

```
try

    f1();
    f2();
    f3();

catch(const char *s)    // catch string error

    // ...

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

    cout << "Caution!"; // partially respond
    throw;               // pass the exception

void f2(void)

    // ...

    throw "Input Error!";
```

You may wish to 'throw' a class object rather than a string or error code. An example of this can be shown as follows:

```
class MyClass

    int *values;
    int maxSize;
public:
    class HotPotatoe ; // exception class
    void aMethod( void );
    AnObject;

void MyClass::aMethod( void )

    // ...

    if( error )
        throw HotPotatoe();

void f( void )

    try
```

```

    AnObject.aMethod();

catch( MyClass::HotPotatoe )

    // handle error

```

SAMPLE USAGE OF EXCEPTION HANDLING

The following code snippet provides an example of a compiler-specific implementation of exception handling. Future versions of C++ compilers will likely incorporate the standard exception handling syntax described above.

```

// exceptions.cp

/*
    Symantec THINK C 7.0 specific code.
*/

#include <iostream.h>
#include <Exceptions.h>

class HotPotatoe TCL_EXCEPTION_CLASS

public:
    TCL_DECLARE_CLASS
    HotPotatoe(double A, double B) a=A; b=B;
    double  a,b;
;
TCL_DEFINE_CLASS_M0(HotPotatoe);

static void f(double x)

    double max = 3.14159;

    if( x > max )
        throw_( HotPotatoe(max,x) );

void main( void )

    TCL_BREAK_ON_CATCH(false);

    try_

        cout << "Example One:" << endl;
        f(5.0);

    catch_reference_( HotPotatoe, oo )

        cout << "limit: " << oo.a << "    value: " << oo.b
        << endl << endl;

    end_try_

    try_

        cout << "Example Two:" << endl;

```

```
try_  
    f(5.0);  
catch_all_  
    cout << "exception caught..." << endl;  
    cout << "rethrow exception..." << endl;  
    throw_same_();  
end_try_  
catch_all_  
    cout << "rethrown exception caught." << endl;  
end_try_  
// end exceptions.cp
```

* Exception handling is new enough that my current compiler doesn't support the 'try', 'catch', and 'throw' keywords (at least not directly), so I don't have any *real* example code segments verified and compiled.