Alexander J King

Exception handling  
CPL  
4.27.19

1) Explain Exception Handling - What is an exception?

Many languages allow programs to trap input/output errors (including EOF) • An exception is any unusual event, either erroneous or not, detectable by either hardware or software, that may require special processing • The special processing that may be required after detection of an exception is called exception handling • The exception handling code unit is called an exception handler

2) Give advantages of a language with built-in Exception Handling

In a language with exception handling – Programs are allowed to trap some exceptions, thereby providing the possibility of fixing the problem and continuing

Where as in a language without exception handling – When an exception occurs, control goes to the operating system, where a message is displayed and the program is terminated

3) Compare the exception handlers of the following languages

a) C++

Added to C++ in 1990

Design is based on that of CLU, Ada, and ML

catch is the name of all handlers. it is an overloaded name, so the formal parameter of each must be unique. The formal parameter need not have a variable – It can be simply a type name to distinguish the handler it is in from others. The formal parameter can be used to transfer information to the handler The formal parameter can be an ellipsis, in which case it handles all exceptions not yet handled

An unhandled exception is propagated to the caller of the function in which it is raised. This propagation continues to the main function. If no handler is found, the default handler is called

b) Java ♥

Based on that of C++, but more in line with OOP philosophy. All exceptions are objects of classes that are descendants of the Throwable class.

The Java library includes two subclasses of Throwable : – Error Thrown by the Java interpreter for events such as heap overflow. Never handled by user programs – Exception. User-defined exceptions are usually subclasses of this. Has two predefined subclasses, IOException and RuntimeException (e.g., ArrayIndexOutOfBoundsException and NullPointerException

**Try** is exactly like c++ which has been utilized throughout my project.

Binding an exception to a handler is simpler in Java than it is in C++ – An exception is bound to the first handler with a parameter is the same class as the thrown object or an ancestor of it. An exception can be handled and rethrown by including a throw in the handler (a handler could also throw a different exception)

c) Visual Basic

Try catch.

4) What is event handling? Why would it be used?

An event is a notification that something specific has occurred, such as a mouse click on a graphical button. The event handler is a segment of code that is executed in response to an event

So for example I could have an even for on click. EX: When the user clicks a button it does an “event” that causes another thing to happen. This is common with “Next” buttons in an installation or a ‘red X’ to close a program.