

Comparative Programming

Chapter 14

Three major language features:-

- 1) Inheritance - Adds reliability (changes are only made to the parent class and are inherited by all the child classes). Also you can add members and methods in child class. Parents can selectively make child classes inherit certain members. Child can override information from the parent.
- 2) Abstract Data Types -
- 3) Polymorphism - Different functions can get called based on the data type. These functions have the same name.

Relation between parent and child

Exception Handling - Gracefully handle situations in which you don't have control. Programs ceases to operate if there's no exception handling.

Give the user the illusion of control of the program.
Take account of all worst case scenarios.

What we are trying to do? Reducing compounding the problem.

Manual Exception Handling can be done but requires much work from the programmer.

An exception is raised when its associate event occurs.
hooking to see if an error occurred.

Try...catch mechanism is used.

Try...Catch

Catch is just an overload.

Design Issues

Order in which we look for these exceptions.

Where are we going to return back in the program when an exception occurs.

Whether we can explicitly call an Exception Handler.

Throw - Run/Throw Exception - It calls the Exception Handler.

Whether the Handler was able to fix the Handler.

* Graceful Termination.

Rollback

Pickup the pieces and move on.

No manual intervention since that might produce unnecessary human errors in something like a database.

In C, we use -

```
try {  
}
```

```
    catch (formal parameter) {
```

```
        -- handler code
```

```
}
```

```
    catch (formal parameter) {
```

```
}
```

No control over some process - use a try catch

Most languages use some form of throw... catch.

Finally statement

No matter what our catch is execute a particular block of code.
It increases the reliability.

Example - File is closed no matter what.

Number for Helpdesk

Event Handler v/s Exception

Exception is abnormal scenario.

Event is normal scenario

We give up control of the process when we use the Handler.

In GUI, we have control over the timing where sets of instructions are processed.

Give the user the control

Event is trying to take control but give up on the timings.

Exceptions are uncalled for. It just encapsulates the instructions we want to protect. It can be done later in the program / in later stages of coding, you can add them.

