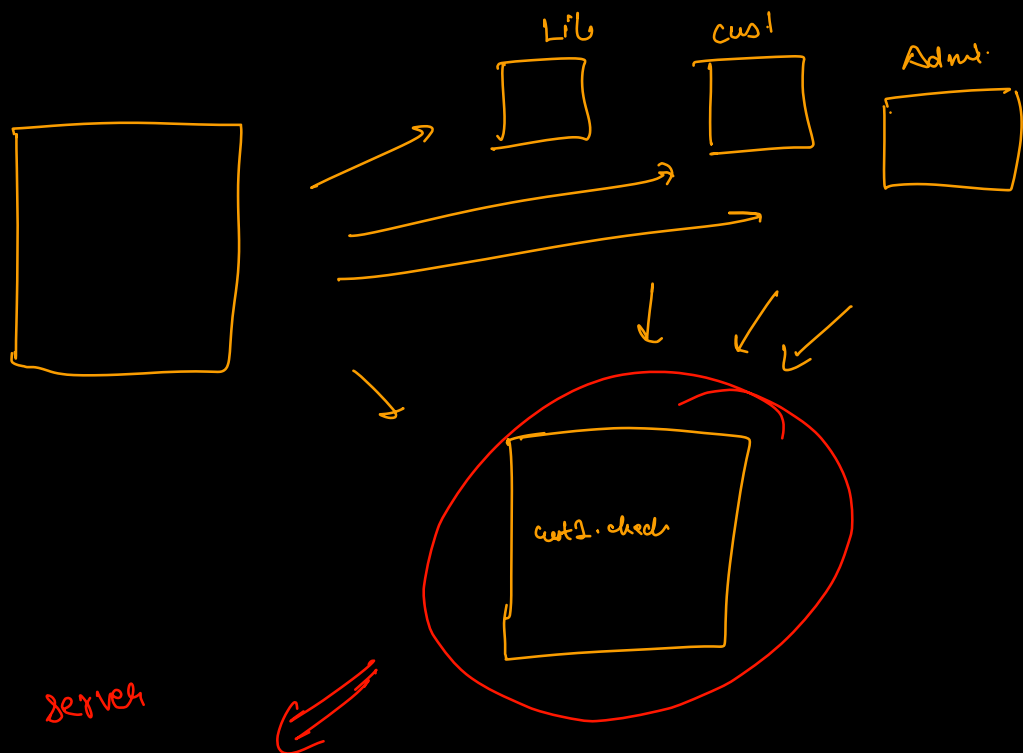


Destructor in Class

`--del-- (self):`

gets called whenever an object is deleted
or your program end.



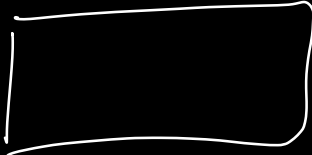
wh

Exception Handling

WhatsApp \Rightarrow 1gb

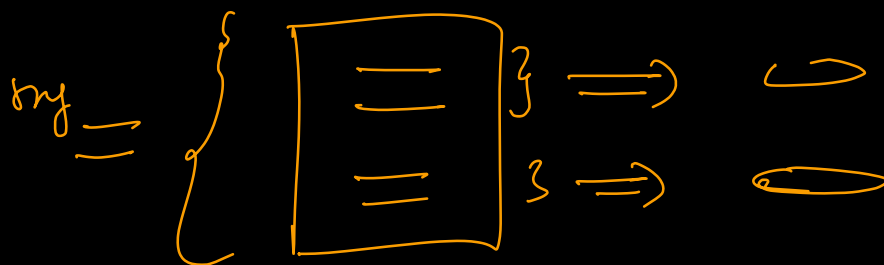
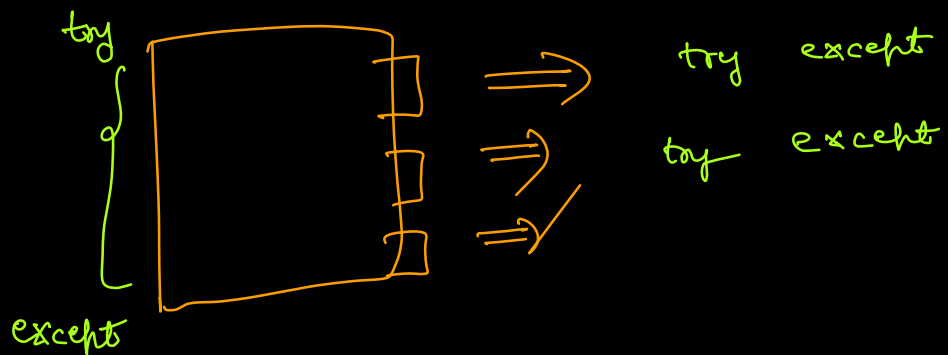
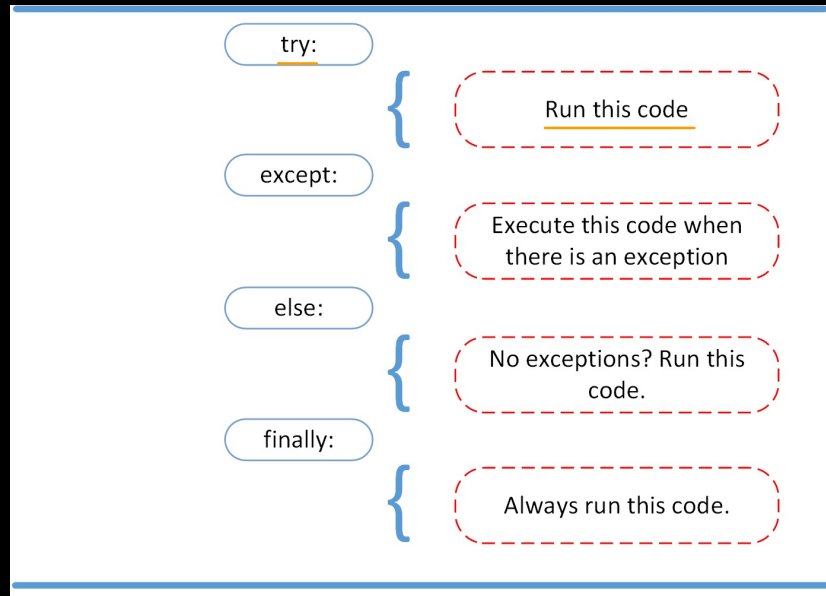


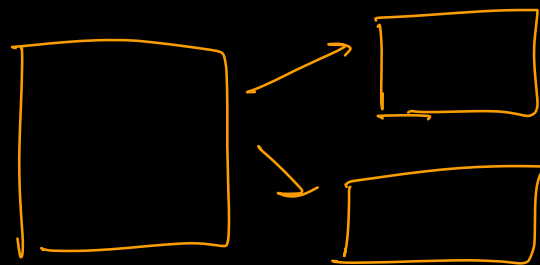
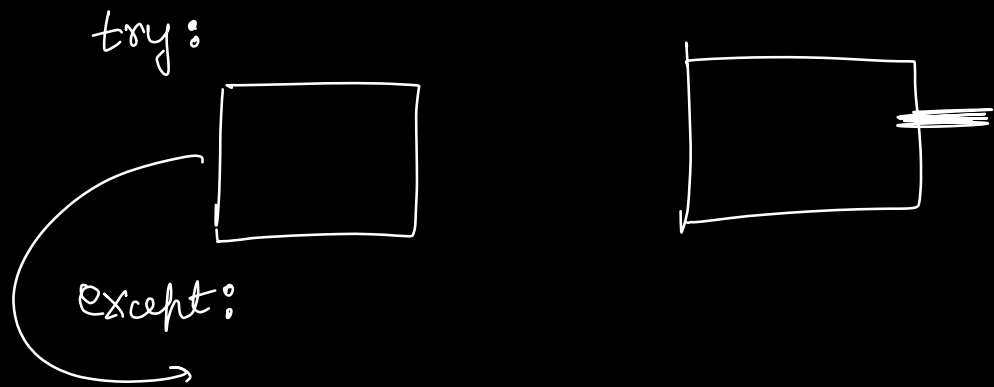
1000

Insta/AB \Rightarrow  4gb

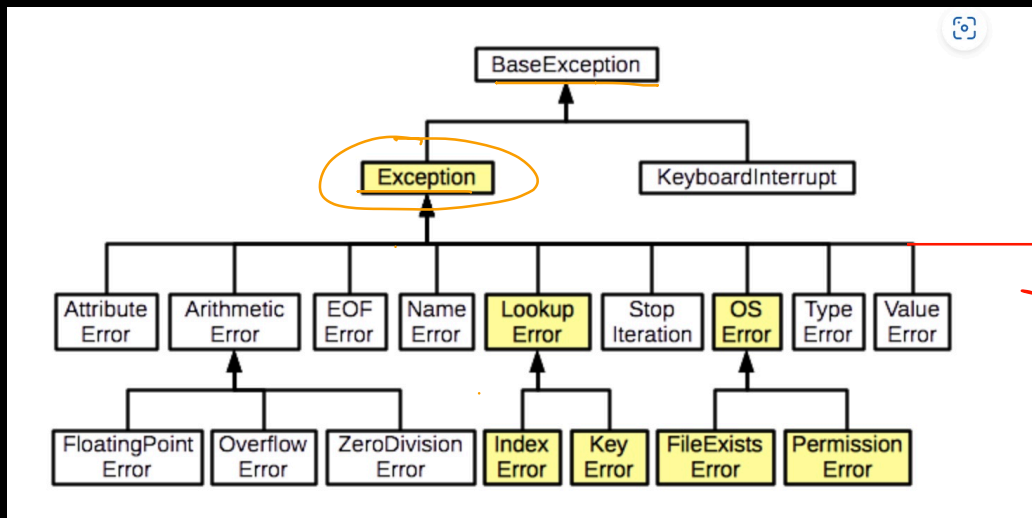
1. We have to avoid shutting down the full code
2. We should handle error gracefully.

In Python, a [runtime error](#) occurs when the program is executing and encounters an unexpected condition that prevents it from continuing. Runtime errors are also known as exceptions and can occur for various reasons such as division by zero, attempting to access an index that is out of range, or calling a function that does not exist.





handling
should not
be some



When python faces an error, it creates an exception object.

We can catch specific exception and handle error based on that.

Exception	Cause of Error
<code>AssertionError</code>	Raised when an <code>assert</code> statement fails.
<code>AttributeError</code>	Raised when attribute assignment or reference fails.
<code>EOFError</code>	Raised when the <code>input()</code> function hits end-of-file condition.
<code>FloatingPointError</code>	Raised when a floating point operation fails.
<code>GeneratorExit</code>	Raised when a generator's <code>close()</code> method is called.
<code>ImportError</code>	Raised when the imported module is not found.
<code>IndexError</code>	Raised when the index of a sequence is out of range.
<code>KeyError</code>	Raised when a key is not found in a dictionary.
<code>KeyboardInterrupt</code>	Raised when the user hits the interrupt key (<code>Ctrl+C</code> or <code>Delete</code>).
<code>MemoryError</code>	Raised when an operation runs out of memory.
<code>NameError</code>	Raised when a variable is not found in local or global scope.
<code>NotImplementedError</code>	Raised by abstract methods.
<code>OSError</code>	Raised when system operation causes system related error.
<code>OverflowError</code>	Raised when the result of an arithmetic operation is too large to be represented.
<code>ReferenceError</code>	Raised when a weak reference proxy is used to access a garbage collected referent.
<code>RuntimeError</code>	Raised when an error does not fall under any other category.
<code>StopIteration</code>	Raised by <code>next()</code> function to indicate that there is no further item to be returned by iterator.
<code>SyntaxError</code>	Raised by parser when syntax error is encountered.
<code>IndentationError</code>	Raised when there is incorrect indentation.
<code>TabError</code>	Raised when indentation consists of inconsistent tabs and spaces.
<code>SystemError</code>	Raised when interpreter detects internal error.
<code>SystemExit</code>	Raised by <code>sys.exit()</code> function.
<code>TypeError</code>	Raised when a function or operation is applied to an object of incorrect type.
<code>UnboundLocalError</code>	Raised when a reference is made to a local variable in a function or method, but no value has been bound to that variable.
<code>UnicodeError</code>	Raised when a Unicode-related encoding or decoding error occurs.
<code>UnicodeEncodeError</code>	Raised when a Unicode-related error occurs during encoding.
<code>UnicodeDecodeError</code>	Raised when a Unicode-related error occurs during decoding.
<code>UnicodeTranslateError</code>	Raised when a Unicode-related error occurs during translating.
<code>ValueError</code>	Raised when a function gets an argument of correct type but improper value.
<code>ZeroDivisionError</code>	Raised when the second operand of division or modulo operation is zero.

