In [1]:

```
try:
    f = open('testfile','w')
    f.write('Test write this')
except IOError:
    # This will only check for an IOError exception and then execute this prin
t statement
   print "Error: Could not find file or read data"
else:
   print "Content written successfully"
   f.close()
```

```
Content written successfully
```

In [2]:

```
try:
    f = open('testfile','r')
    f.write('Test write this')
except IOError:
    # This will only check for an IOError exception and then execute this prin
t statement
   print "Error: Could not find file or read data"
else:
   print "Content written successfully"
   f.close()
```

```
Error: Could not find file or read data
```

In [3]:

```
try:
    f = open('testfile','r')
    f.write('Test write this')
except:
    # This will check for any exception and then execute this print statement
   print "Error: Could not find file or read data"
else:
   print "Content written successfully"
   f.close()
```

```
Error: Could not find file or read data
```

In [4]:

```
try:
   f = open("testfile", "w")
   f.write("Test write statement")
finally:
   print "Always execute finally code blocks"
```

```
Always execute finally code blocks
```

In [5]:

```
def askint():
        try:
            val = int(raw_input("Please enter an integer: "))
        except:
            print "Looks like you did not enter an integer!"

        finally:
            print "Finally, I executed!"
        print val
askint()
```

```
Please enter an integer: 5.3
Looks like you did not enter an integer!
Finally, I executed!

---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-5-93f04ee89e42> in <module>()
      8             print "Finally, I executed!"
      9         print val
---> 10 askint()

<ipython-input-5-93f04ee89e42> in askint()
      7         finally:
      8             print "Finally, I executed!"
----> 9         print val
     10 askint()

UnboundLocalError: local variable 'val' referenced before assignment
```

In [6]:

```
def askint():
        try:
            val = int(raw_input("Please enter an integer: "))
        except:
            print "Looks like you did not enter an integer!"

        finally:
            print "Finally, I executed!"
        print val
askint()
```

```
Please enter an integer: 5
Finally, I executed!
5
```

In [7]:

```
def askint():
        try:
            val = int(raw_input("Please enter an integer: "))
        except:
            print "Looks like you did not enter an integer!"
            val = int(raw_input("Try again-Please enter an integer: "))
        finally:
            print "Finally, I executed!"
        print val
askint()
```

```
Please enter an integer: 5
Finally, I executed!
5
```

In [8]:

```
def askint():
        try:
            val = int(raw_input("Please enter an integer: "))
        except:
            print "Looks like you did not enter an integer!"
            val = int(raw_input("Try again-Please enter an integer: "))
        finally:
            print "Finally, I executed!"
        print val
askint()
```

```
Please enter an integer: 5.3
Looks like you did not enter an integer!
Try again-Please enter an integer: 5
Finally, I executed!
5
```

In [10]:

```
def askint():
        try:
            val = int(raw_input("Please enter an integer: "))
        except:
            print "Looks like you did not enter an integer!"
            val = int(raw_input("Try again-Please enter an integer: "))
        finally:
            print "Finally, I executed!"
        print val
askint()
```

```
Please enter an integer: 5.3
Looks like you did not enter an integer!
Try again-Please enter an integer: 5.3
Finally, I executed!

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-10-29fb32a22113> in <module>()
      8             print "Finally, I executed!"
      9         print val
---> 10 askint()

<ipython-input-10-29fb32a22113> in askint()
      4         except:
      5             print "Looks like you did not enter an integer!"
----> 6             val = int(raw_input("Try again-Please enter an integer: "
))
      7         finally:
      8             print "Finally, I executed!"

ValueError: invalid literal for int() with base 10: '5.3'
```