Exceptions

An exception is an error that occurs while a program is running, causing the program to abruptly halt. You can use the try/except statement to gracefully handle exceptions.

```
# This program divides a number by another number.

def main():
    # Get two numbers.
    num1 = int(input('Enter a number: '))
    num2 = int(input('Enter another number: '))

    # Divide num1 by num2 and display the result.
    result = num1 / num2
    print(num1, 'divided by', num2, 'is', result)

# Call the main function.
main()
```

```
# This program divides a number by another number.

def main():
    # Get two numbers.
    num1 = int(input('Enter a number: '))
    num2 = int(input('Enter another number: '))

    # If num2 is not 0, divide num1 by num2
    # and display the result.
    if num2 != 0:
        result = num1 / num2
        print(num1, 'divided by', num2, 'is', result)
    else:
        print('Cannot divide by zero.')

# Call the main function.
main()
```

```
>>> def main():
        hours=int(input('how many hours did u work'))
        pay_rate=float(input('enter ur hourly pay rate'))
        gross_pay=hours*pay_rate
        print('gross pay:$',format(gross_pay,',.2f'),sep='')
>>> main()
how many hours did u work2
enter ur hourly pay rate3
gross pay:$6.00
```

OR How exception occur

```
>>> main()
how many hours did u work3
enter ur hourly pay rateforty
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
```

```
   main()
 File "<pyshell#5>", line 3, in main
   pay_rate=float(input('enter ur hourly pay rate'))
ValueError: could not convert string to float: 'forty'
>>>
```

Handling Exceptions
Use of Try Block:

```
try:
      statement
      statement
      etc.

except ExceptionName:
       statement
       statement
       etc.
>>> def main():
          try:
             hours=int(input('how many hours did u work'))
             pay_rate=float(input('enter ur hourly pay rate'))
             gross_pay=hours*pay_rate
             print('gross pay:$',format(gross_pay,',.2f'),sep='')
          except ValueError:
                   print('error: hours and hourly rate pay rate must be valid numbers')


>>> main()
how many hours did u work2
enter ur hourly pay rateforty
error: hours and hourly rate pay rate must be valid numbers
>>>
```

When try to read the file which doesn't exist

```
>>> def main():
          filename=input('enter the file name')
          try:
                   infile=open(filename,'r')
                   contents=infile.read()
                   print(contents)
                   infile.close()
          except IOError:
                   print('An error occured trying to read')
                   print('the file', filename)
```

```
>>> main()
enter the file namehoney.txt
An error occured trying to read
the file honey.txt
```

Handling Multiple Exceptions

```
# This program displays the total of the
# amounts in the sales_data.txt file.

def main():
    # Initialize an accumulator.
    total = 0.0

    try:
        # Open the sales_data.txt file.
        infile = open('sales_data.txt', 'r')

        # Read the values from the file and
        # accumulate them.
        for line in infile:
            amount = float(line)
            total += amount

        # Close the file.
        infile.close()

        # Print the total.
        print(format(total, ',.2f'))

    except IOError:
        print('An error occured trying to read the file.')

    except ValueError:
        print('Non-numeric data found in the file.')

    except:
        print('An error occured.')

# Call the main function.
main()
```

\

An Exception's default Error Message

```
# This program displays the total of the
# amounts in the sales_data.txt file.

def main():
    # Initialize an accumulator.
    total = 0.0

    try:
        # Open the sales_data.txt file.
        infile = open('sales_data.txt', 'r')

        # Read the values from the file and
        # accumulate them.
        for line in infile:
            amount = float(line)
            total += amount

        # Close the file.
        infile.close()

        # Print the total.
        print(format(total, ',.2f'))
    except Exception as err:
        print(err)

# Call the main function.
main()
```

The else clause

```
try:
  statement
  statement
  etc.
except ExceptionName:
  statement
  statement
  etc.
else:
  statement
  statement
  etc.
```

```
# This program displays the total of the
# amounts in the sales_data.txt file.

def main():
    # Initialize an accumulator.
    total = 0.0

    try:
        # Open the sales_data.txt file.
        infile = open('sales_data.txt', 'r')

        # Read the values from the file and
        # accumulate them.
        for line in infile:
            amount = float(line)
            total += amount

        # Close the file.
        infile.close()
    except Exception as err:
        print(err)
    else:
        # Print the total.
        print(format(total, ',.2f'))
```

Finally clause

```
try:
  statement
  statement
  etc.
except ExceptionName:
  statement
  statement
  etc.
finally:
  statement
  statement
  etc.
```