INF 502 – SOFTWARE DEVELOPMENT METHODOLOGIES

Python: advanced concepts

NAU NORTHERN ARIZONA UNIVERSITY

Errors and exceptions

- *Syntax Errors* occur when the *grammar* of a Python statement is incorrect

```
>>> x = 2
>>> if (x > 0)
  File "<stdin>", line 1
    if (x > 0)
              ^
SyntaxError: invalid syntax
>>>
```

- Exceptions are errors that happen in execution time, even with correct Syntax

```
>>> list1 = (1,2,3,4,5)
>>> list1[8]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: tuple index out of range
>>>
```

Errors and Exceptions

Error	Description/example
IOError	e.g. file not found
IndexError	an attempt to access a sequence (such as a list with an index out of range)
TypeError	an operation or function applied to an object of inappropriate type
NameError	a variable name is not recognised
ValueError	an operation or function receives an argument of the right type but an inappropriate value
ZeroDivisionError	a specific type of ValueError raised when an attempt is made to divide by zero.

Exception Hunting

Knowing the potential exceptions is key… Experience (bad ones) help you

Exceptions can be ‘caught’ and handled in Python script:

try … catch … finally

```
try:
    # <do something that might fail>
except (<exception1>, <exception2>, ...):
    # <something went wrong: deal with it>
else:
   # <what to do if no exception>
finally:
    # <statements here are always executed>
```

Exception Hunting

Knowing the potential exceptions is key… Experience (bad ones) help you

Exceptions can be ‘caught’ and handled in Python script:

try … catch … finally

```
import os
filename = "other.csv"
try:    #try to remove the file "other.csv"
    os.remove(filename)
except FileNotFoundError: #if a FileNotFoundError happens when it tries...
    print("There is no file named", filename)  #Log the problem
```

Exception Hunting

```
print ("Are you authorized to drink in AZ?")
try:
  age = input ("Type your age: ")
  age = int(age)

except (ValueError):
  print ("The value typed is not an integer")

else:
  if (age < 21):
    print ("You cannot drink")
  else:
    print ("You can drink")
    age = int(age)
```

Raising exceptions

```
def canDrink(age):
  if (age < 0):
    #a new ValueError will be raised to who called canDrink
    my_error = ValueError("{0} is not a valid age ". format(age))
    raise my_error
  result = True if (age >= 21) else False
  return result
```

Python modules

Some Python Modules

Longer programs can be split up into separate files

Each file, with functions and variables, can be considered a *module*

Modules are *imported* using the **import** statement

The Standard Python Library consists of some modules (and packages of modules)

Some Python Modules

math	Mathematical functions
sys	System-specific parameters and functions
os	Miscellaneous operating system interfaces
random	Generate (pseudo-)random numbers
zlib	A compression library (compatible with gzip)
argparse	Command-line argument parsing
urllib	Open and access resources across the internet by URL
datetime	Basic date and time types
re	Regular expressions

MATH module

sqrt(x)	exp(x)	log(x)	log(x, base)
log10(x)	sin(x), etc.	asin(x), etc.	sinh(x), etc.
hypot(x, y)	pi	e	

OS module

- Miscellaneous operating system interfaces
  - path: Manipulate file and directory names
  - environ: A mapping object representing the string environment. *e.g.* os.environ[‘HOME’] is the pathname of your home directory (on some systems)
  - remove: Delete a file
  - rename: Rename a file
  - stat: get information about file read/write permissions, last time of modification, etc.
  - listdir : Return a list of the entries in a directory

OS module

```
>>> import os
>>> HOME = os.environ['HOME']
>>> print(os.listdir(HOME))
['.bash_history', '.bash_profile', 'Desktop', 'Documents, 'Downloads', 'Library',
'research', 'teaching', ... ]
```

We can create our own modules

```
# File myModule.py

def myFunction():
    print("This function belongs to my Module")

def anotherFunction():
    print("This is another function in my module")
```

```
# File modularization.py

import myModule

myModule.myFunction()
myModule.anotherFunction()
```

The end

NAU NORTHERN ARIZONA UNIVERSITY