Python Programming

KNOWLEDGE RESOURCE CENTRE

Errors

- Compile time errors
- Logical errors
- Runtime errors

Exception

- An exception is an event which occurs during the execution of a program that disrupts the normal flow of the program's execution
- Exception is a runtime error.
- Process of responding to the exception raised is called exception handling

Errors

Error	Description
SyntaxError	Raised by the parser when a syntax error occurs
IndentationError	Raised when there is an incorrect indentation
IndexError	Raised when the index of a sequence is out of bound
KeyError	Raised when a non existent key is accessed
NameError	Raised when a non existing identifier is referred
TypeError	Raised when a wrong type of parameter is sent to a function
ValueError	Raised when parameter has an invalid value
ZeroDivisionError	Raised when division is done by zero
ModuleNotFoundError	Raised when the imported module is not found
UnboundLocalError	Raised when a reference is made to a local variable in a function or method, but no value has been bound to that variable
AttributeError	Raised when an object tries to access a member which is not available
FileNotFoundError	Raised if file is not found

Exception

- try
  - Keyword used to keep the code segment under check
- except
  - Segment to handle the exception after catching it
- else
  - Run this when no exception exists
- finally
  - No matter what , run this code if or not an exception occurs

Syntax

```
try:
        You do your operations here; ......................
except ExceptionI as [variable]:
        If there is ExceptionI, then execute this block.
[except ExceptionII as [variable]]:
        If there is ExceptionII, then execute this block.
......................
[else:
        If there is no exception then execute this block.
]
```

Exception

- A single try statement can have multiple except statements.
- You can also provide a generic except clause, which handles any exception.
- After the except clause(s), you can include an else-clause. The code in the else-block executes if the code in the try: block does not raise an exception.
- The else-block is a good place for code that does not need the try: block's protection.

try with multiple except

```
try:
        You do your operations here; ......................
except Exception1:
        statements
except Exception2:
        statements

……………..
 else:
        If there is no exception then execute this block.
```

try with multiple except

```
try:
        You do your operations here; ......................
except (Exception1,Exception2,Exception3….) as variable:
        statements
```

try-finally Clause

- finally block can be used along with a try block.
- The finally block is a place to put any code that must execute, whether the try-block raised an exception or not.
- The syntax of the try-finally is

```
try:
   You do your operations here;
finally:
   This would always be executed.
```

- You cannot use else clause as well along with a finally clause.

Raising your own exception

- The raise statement allows the programmer to force a specific exception to occur.
- raise <error>(“Type your message here”)
- eg

```
try:
    raise NameError("Hi there")  # Raise Error
except NameError:
    print ("An exception")
raise
```

User defined Exceptions

- To create your own exceptions we need to write a class that inherits from the super Exception class

Object Oriented Programming

- OOPs is a method of programming that models process or things in the world as a class or object
- Class is a blue print for objects.
- Classes can contain methods and attributes
- Object is an instance of a class

- class <classname>:

  methods

- <varname>=<classname>()

__init__

- Python classes have a special method named __init__ which gets called every time an instance of the class is created.
- def __init__(self,first,a):

```
    self.name=first
    self.age=a
```

self refers to the current instance

self

- Self is a reference variable which always point to the current object
- First argument to constructor is always self
- First argument to the instance method is always self
- Self is not a keyword

Instance methods

- instance methods should have self as the first argument of the method

Class Attributes & methods

- class attributes are given outside all the methods in the class
- In the methods of the class, class attribute is referred using the class name
- class methods will take class as argument and refer to class attributes using cls.
- @classmethod should be given with class methods

Inheritance

- Inheriting the properties from base class to child class is called inheritance
- In Python inheritance works by passing the parent class as an argument to the definition of a child class
- super keyword can be used to call the super class __init__() method

- Method Overloading
- Method overriding

Regular Expressions

- A regular expression is a special sequence of characters which is used for pattern matching.
- Python provides a module re for regular expressions

Functions in re

- findall() returns list of all matches
  - re.findall(pattern,source_string)
    - returns the list of all matches if the pattern exists in the string
    - returns an empty list if no match exists
- search() returns match object if there are any matches
  - re.search(pattern,source_string)
    - returns a match object if match exists(first occurrence)
    - returns None if match doesn't exist

Functions in re

- match object
  - start() gives the position of occurrence of the match
  - span()returns a tuple which contains the start and end of the match
  - string returns the actual string used in pattern matching

Functions in re

- split() will split the string with the given pattern
  - re.split(pattern,sourcestring,[maxsplit])
- sub() will substitute a new string for an old string
  - re.sub(old pattern, new pattern,sourcestring,[no of occurrences])

Meta characters in Regex

- [] - returns a match if the string contains the pattern/characters specified in []
- ^ - returns a match if the string starts with the given pattern/characters
- $ - returns a match if the string ends with the given pattern/characters
- . - Matches any character except newline

Meta characters in Regex

- * - returns a match if there is 0 or more occurrences of the pattern
- + - returns a match if there is 1 or more occurrences of the pattern
- {} - returns a match if there is a specified number of occurrences of a pattern

Sets

- [abc] – returns a match if the given string contains any one of specified character
- [a-z]– returns a match if the given string contains characters within the specified range
- [^abc]- returns a match if the given string contains any other character other than that specified
- [5678]
- [0-9]
- [0-9][0-9]

Thank you