Exceptions

- What are Exceptions
- Try Except Syntax
- How it Works
- Multiple Except Statements
- Raising Exceptions
- Custom Exceptions
- Complete try – except – else – finally syntax

What are Exceptions

- Exceptions are errors raised during the execution of the program
- Exceptions are not syntax errors
- Exceptions can be handled in a program, which otherwise result in termination of the program

trainer.cpp@gmail.com

Some Examples

- 1/0
  ZeroDivisionError
- [1,2,3] ** 2
  TypeError
- x*x
  NameError
- x = 1
  x.y
  AttributeError
- L = [1,2,3]
  L[4]
  IndexError

Try Except Syntax

- **try:**
  <code that might throw exception>
  except <optional Exception name or tuple>:
  exception handling code

```
try:
    value = int(input())
except ValueError:
    print("Can't you enter an integer")
```

```
try:
    value = int(input())
except (ValueError, KeyboardInterrupt):
    print("Stop Messing!!")
```

Working

- When the code inside **try** clause executes:

  If there is an exception, code below the point of exception is skipped and the code belonging to **except** gets executed.

  If however, there is no exception, the code of **except** clause is not executed.

- Still, if the **except** clause(s), does not specify the exception thrown, the exception propagates till either it is finally caught somewhere, or the program terminates.

trainer.cpp@gmail.com

Multiple except Clauses and Exception object

- Multiple Except Clauses

```
try:
    statements                          # code with possibly exception conditions
except <exception name>:                # run for this specific exception
    statements
except (<tuple of exception names>):    # run for any of these
    statements
```

- Exceptions Object

```
try:
    statements                          # code with possibly exception conditions
except <exception name> as <variable>: # store the exception in variable
    statements
```

trainer.cpp@gmail.com

Complete Exception Syntax

```
try:
    statements                                  # code with possibly exception conditions
except <exception name>:                        # run for this specific exception
    statements
except (<tuple of exception names>):            # run for any of these
    statements
except <exception name> as <variable>:          # store the exception in variable
    statements
except:                                         # run for all remaining exceptions
    statements
else:                                           # else: run when no exceptions
    statements
finally:                                        # finally: run irrespective of exception
    statements
```

trainer.cpp@gmail.com

Else and Finally options

- **Else**:
  - Gets executed only in case there is no exception
  - Must always be preceded by at least an except clause
- **Finally**:
  - Always gets executed
  - Even if one of the except handlers itself raises some exception
  - No exception occurred anywhere

trainer.cpp@gmail.com

Understanding Empty Except

- ```
  try:
      exit()
  except :                    # catch all exceptions including one used for system errors
      print("Caught")
  ```

- ```
  try:
      exit()                  # also try the input function
  except Exception:           # catch all possible exceptions except exit(),
      print("Caught")         # keyboard interrupt .. (Python 3.X)
  ```

trainer.cpp@gmail.com

Raising Exceptions and Re-raising

- The **raise** keyword is used to raise exceptions.
- Syntax:

  raise <Name of Exception/ Exception Object>

- *except <Exception>:*

  raise #re raises the exception caught

trainer.cpp@gmail.com

Custom Exceptions and Exception hierarchy

- All user defined Exceptions should inherit from Exception Class
- *class MyException(Exception):*

  pass
- When creating an exception hierarchy, the except clauses should appear in the order from the **derived to base** class.

trainer.cpp@gmail.com

Assert statement and Debug Mode

- assert <Condition>, <some assertion message>

  assert raises an AssertionError exception, when the condition is False.
- __debug__ constant if set to True, only then assertions are raised
- -O option runs in non-debug mode

trainer.cpp@gmail.com

Exception Hierarchy

- **BaseException** : Parent of all exception classes in Python
- **Exception**: Inherits from BaseException. Parent of all exceptions except some system exceptions (SystemExit, KeyboardInterrupt…)
- All Exception classes should inherit from *Exception* and not *BaseException*

trainer.cpp@gmail.com