

assessment5

June 19, 2023

[]: #Q1

[]: 'here we are using exception class because as exception are basically like the
↳blueprint or foundation for all exception which contain all the instruction
↳which is need to be perform for and managing exception effectively
1.'by inheriting from the exception class custom exception it follow the same
↳interface and behaviour . consistency allows devloper to handle custom
↳exception in similar manner as the standered exception
2.'inherting the exception class enable our custom to be a part of our
↳exceptionn hierarchy . the exception hierarcy oraginzes exception into like
↳a structure like tree structure .it allows more specific exception handling
↳and catching exception at the different level
3. 'handling mechanisms class provide various method and property that
↳facilitate exception handling .this include the get message() which allows
↳used to get back the error message with the exception '

[]: #Q2

```
[3]: def print_exception_hierarchy(exception_class,indent=0):  
    print(' '*indent+exception_class.__name__)  
    for subclass in exception_class.__subclasses__():  
        print_exception_hierarchy(subclass, indent+4)  
print_exception_hierarchy(BaseException)
```

```
BaseException  
  Exception  
    TypeError  
      FloatOperation  
      MultipartConversionError  
    StopAsyncIteration  
    StopIteration  
    ImportError  
      ModuleNotFoundError  
      ZipImportError  
    OSError  
      ConnectionError  
        BrokenPipeError  
        ConnectionAbortedError
```

- ConnectionRefusedError
 - ConnectionResetError
 - RemoteDisconnected
 - BlockingIOError
 - ChildProcessError
 - FileExistsError
 - FileNotFoundError
 - IsADirectoryError
 - NotADirectoryError
 - InterruptedError
 - InterruptedSystemCall
 - PermissionError
 - ProcessLookupError
 - TimeoutError
 - UnsupportedOperation
 - itimer_error
 - herror
 - gaierror
 - SSLError
 - SSLCertVerificationError
 - SSLZeroReturnError
 - SSLWantWriteError
 - SSLWantReadError
 - SSLSyscallError
 - SSLEOFError
 - Error
 - SameFileError
 - SpecialFileError
 - ExecError
 - ReadError
 - URLError
 - HTTPError
 - ContentTooShortError
 - BadGzipFile
- EOFError
- IncompleteReadError
- RuntimeError
- RecursionError
 - NotImplementedError
 - ZMQVersionError
 - StdinNotImplementedError
 - _DeadlockError
 - BrokenBarrierError
 - BrokenExecutor
 - BrokenThreadPool
 - SendfileNotAvailableError
 - ExtractionError
 - VariableError

- NameError
 - UnboundLocalError
- AttributeError
 - FrozenInstanceError
- SyntaxError
 - IndentationError
 - TabError
- LookupError
 - IndexError
 - KeyError
 - NoSuchKernel
 - UnknownBackend
 - CodecRegistryError
- ValueError
 - UnicodeError
 - UnicodeEncodeError
 - UnicodeDecodeError
 - UnicodeTranslateError
 - UnsupportedOperation
 - JSONDecodeError
 - SSLCertVerificationError
 - Error
 - UnsupportedDigestmodError
 - IllegalMonthError
 - IllegalWeekdayError
 - ParserError
 - ClassNotFound
 - ClipboardEmpty
 - MessageDefect
 - NoBoundaryInMultipartDefect
 - StartBoundaryNotFoundDefect
 - CloseBoundaryNotFoundDefect
 - FirstHeaderLineIsContinuationDefect
 - MisplacedEnvelopeHeaderDefect
 - MissingHeaderBodySeparatorDefect
 - MultipartInvariantViolationDefect
 - InvalidMultipartContentTransferEncodingDefect
 - UndecodableBytesDefect
 - InvalidBase64PaddingDefect
 - InvalidBase64CharactersDefect
 - InvalidBase64LengthDefect
 - HeaderDefect
 - InvalidHeaderDefect
 - HeaderMissingRequiredValue
 - NonPrintableDefect
 - ObsoleteHeaderDefect
 - NonASCIIILocalPartDefect
 - InvalidDateDefect

- MacroToEdit
- InvalidFileException
- UnequalIterablesError
- InvalidVersion
- _InvalidELFFileHeader
- InvalidWheelFilename
- InvalidSdistFilename
- InvalidSpecifier
- InvalidMarker
- UndefinedComparison
- UndefinedEnvironmentName
- InvalidRequirement
 - RequirementParseError
- InvalidVersion
- AssertionError
- ArithmeticError
 - FloatingPointError
 - OverflowError
 - ZeroDivisionError
 - DivisionByZero
 - DivisionUndefined
- DecimalException
 - Clamped
 - Rounded
 - Underflow
 - Overflow
 - Inexact
 - Underflow
 - Overflow
 - Subnormal
 - Underflow
 - DivisionByZero
 - FloatOperation
 - InvalidOperation
 - ConversionSyntax
 - DivisionImpossible
 - DivisionUndefined
 - InvalidContext
- SystemError
 - CodecRegistryError
- ReferenceError
- MemoryError
- BufferError
- Warning
 - UserWarning
 - GetPassWarning
 - FormatterWarning
 - EncodingWarning

- DeprecationWarning
 - ProvisionalWarning
- PendingDeprecationWarning
- SyntaxWarning
- RuntimeWarning
 - ProactorSelectorThreadWarning
 - UnknownTimezoneWarning
 - PEP440Warning
- FutureWarning
 - ProvisionalCompleterWarning
- ImportWarning
- UnicodeWarning
- BytesWarning
- ResourceWarning
- DeprecatedTzFormatWarning
- PkgResourcesDeprecationWarning
- _OptionError
- _Error
- error
- Verbose
- Error
- SubprocessError
 - CalledProcessError
 - TimeoutExpired
- TokenError
- StopTokenizing
- ClassFoundException
- EndOfBlock
- TraitError
- Error
- Error
 - CancelledError
 - TimeoutError
 - InvalidStateError
- _GiveupOnSendfile
- error
- Incomplete
- TimeoutError
- InvalidStateError
- LimitOverrunError
- QueueEmpty
- QueueFull
- Empty
- Full
- ArgumentError
- ZMQBaseError
 - ZMQError
 - ContextTerminated

- Again
 - InterruptedSystemCall
- ZMQBindError
- NotDone
- PickleError
 - PicklingError
 - UnpicklingError
- _Stop
- ArgumentError
- ArgumentTypeError
- ConfigError
 - ConfigLoaderError
 - ArgumentError
 - ConfigFileNotFound
- ConfigurableError
 - MultipleInstanceError
- ApplicationError
- error
- TimeoutError
- error
- ReturnValueIgnoredError
- KeyReuseError
- UnknownKeyError
- LeakedCallbackError
- BadYieldError
- ReturnValueIgnoredError
- Return
- InvalidPortNumber
- error
- LZMAError
- RegistryError
- _GiveupOnFastCopy
- Error
 - NoSectionError
 - DuplicateSectionError
 - DuplicateOptionError
 - NoOptionError
 - InterpolationError
 - InterpolationMissingOptionError
 - InterpolationSyntaxError
 - InterpolationDepthError
 - ParsingError
 - MissingSectionHeaderError
- NoIPAddresses
- BadZipFile
- LargeZipFile
- BadEntryPoint
- NoSuchEntryPoint

- DuplicateKernelError
- ErrorDuringImport
- NotOneValueFound
- CannotEval
- OptionError
- BdbQuit
- Restart
- ExceptionPexpect
 - EOF
 - TIMEOUT
- PtyProcessError
- FindCmdError
- HomeDirError
- ProfileDirError
- IPythonCoreError
 - TryNext
 - UsageError
 - StdinNotImplementedError
- InputRejected
- GetoptError
- ErrorToken
- PrefilterError
- AliasError
 - InvalidAliasError
- Error
 - InterfaceError
 - DatabaseError
 - InternalError
 - OperationalError
 - ProgrammingError
 - IntegrityError
 - DataError
 - NotSupportedError
- Warning
- SpaceInInput
- DOMException
 - IndexSizeErr
 - DomstringSizeErr
 - HierarchyRequestErr
 - WrongDocumentErr
 - InvalidCharacterErr
 - NoDataAllowedErr
 - NoModificationAllowedErr
 - NotFoundErr
 - NotSupportedErr
 - InuseAttributeErr
 - InvalidStateErr
 - SyntaxErr

- InvalidModificationErr
- NamespaceErr
- InvalidAccessErr
- ValidationError
- ValidationError
- EditReadOnlyBuffer
- _Retry
- InvalidLayoutError
- HeightIsUnknownError
- ParserSyntaxError
- InternalParseError
- _PositionUpdatingFinished
- SimpleGetItemNotFound
- UncaughtAttributeError
- HasNoContext
- ParamIssue
- _JediError
 - InternalError
 - WrongVersion
 - RefactoringError
- OnErrorLeaf
- InvalidPythonEnvironment
- MessageError
 - MessageParseError
 - HeaderParseError
 - BoundaryError
 - MultipartConversionError
 - CharsetError
- Error
- HTTPException
 - NotConnected
 - InvalidURL
 - UnknownProtocol
 - UnknownTransferEncoding
 - UnimplementedFileMode
 - IncompleteRead
 - ImproperConnectionState
 - CannotSendRequest
 - CannotSendHeader
 - ResponseNotReady
 - BadStatusLine
 - RemoteDisconnected
 - LineTooLong
- InteractivelyDefined
- KillEmbedded
- Error
 - NoSuchProcess
 - ZombieProcess


```

        AccessDenied
        TimeoutExpired
    _Ipv6UnsupportedError
    QueueEmpty
    QueueFull
    DebuggerInitializationError
    ExpatError
    Error
        ProtocolError
        ResponseError
        Fault
    ParseBaseException
        ParseException
        ParseFatalException
            ParseSyntaxException
    RecursiveGrammarException
    ResolutionError
        VersionConflict
            ContextualVersionConflict
        DistributionNotFound
        UnknownExtra
    _Error
    UnableToResolveVariableException
    InvalidTypeInArgsException
GeneratorExit
SystemExit
KeyboardInterrupt
CancelledError
AbortThread

```

```
[ ]: #Q3
```

```
[ ]: 'Arithmetic error are the error in the mathematical error according to the
    ↳logic it is right but according to the programming language it is error '
```

```
[15]: import logging

def perform_division(dividend, divisor):
    try:
        result = dividend / divisor
        return result
    except ZeroDivisionError as e:
        logging.error("Error: Division by zero is not allowed.")
    except Exception as e:
        logging.error("An error occurred during division: %s", str(e))
logging.basicConfig(filename='exception.log', level=logging.ERROR)
```

```
dividend = 1000
divisor = 0
result = perform_division(dividend, divisor)
```

```
[16]: import logging
import sys

logging.basicConfig(filename='exception.log',level=logging.ERROR)

try:
    result= sys.maxsize+1
except OverflowError as e:
    logging.error('Error: this calculation exceeds the maximum limit.')
```

[]: #Q4

[]: 'lookuperror occurs only when the class exception fails when a key or index_

- ↳lookup fails in various data structure it may be list or dict . it is a_
- ↳subclass of exception that provides a common base for exception related to_
- ↳failure '

```
[20]: import logging
logging.basicConfig(filename='key.txt', level=logging.
    ↳ERROR,format='%(message)s-%(levelname)s')

my_dict={'a':1,'b':2,'c':3}
try:
    value=my_dict['d']
except KeyError as e:
    logging.error('Error:key not found in the dict')
```

```
[21]: import logging
logging.basicConfig(filename='list.txt',level=logging.ERROR)
my_list=[1,2,3]
try:
    value=my_list[5]
except IndexError as e:
    logging.error('Error:index is out of range')
```

[]: #Q5

[]: 'import error occurs when the particular function is not available in the_

- ↳inbuilt function of python when it fails to locate a inbuilt module it_
- ↳occurs . it is a sub class of exception it occurs failure in the importing_
- ↳of module'

```
[23]: import logging
logging.basicConfig(filename='logging.txt',level=logging.ERROR)
try:
    import non_existent_module
except ImportError as e:
    logging.error("Error: Failed to import the module.")
try:
    import non_existent_module
except ModuleNotFoundError as e:
    logging.error('Error:module not found')
```

[]: 'in module not found error where the programme as written a diff module name_
 ↳variable but calling different module so the module not found error occurs '

[]: #Q6

[]: 'be sepcific in exception handling'
 'use multiple blocks if u anticipate multiple type of error use multiple '
 'finally to ensure the code to be get cleanup after exceuting .this get after_
 ↳try and except block'
 'avoid except block without specifing the perfect exception becz it hide the_
 ↳error and makes more difficult in debugging'
 ' make the try block as small as possible as it help to handle the exception_
 ↳more easily'