Notes on Chapter 7 - Exceptions and Assertions

Swarup Tripathy *

February 2022

A curated list of important points for my reference.

1. an exception is usually defined as 'something that does not conform to the norm' and therefore somewhat rare. There is nothing rare about exceptions in Python. They are everywhere.
2. Most commonly ocurring types of exceptions are
   - TypeError
   - IndexError
   - NameError
   - ValueError
3. Given the following code

```
def readVal(valType, requestMsg, errorMsg):
while True:
    val = input(requestMsg+' ')
    try:
        return valType(val)
    except ValueError:
        print(val,errorMsg)

readVal(int, 'enter an integer:','is not an integer')
```

   The function readVal is **Polymorphic**, i.e., it works for arguments of many different types. Such functions are easy to write in Python, since types are First Class Objects.

   What are **First Class Objects?** First Class Objects in Python

*John V Guttag

A first class object is an entity that can be dynamically created, destroyed, passed to a function, returned as a value, and have all the rights as other variables in the programming language have. Depending on the language this can imply

- being expressible as an anonymous literal value
- being storable in variables
- being storable in data structures
- having an intrinsic identity (independent of any given name)
- being comparable for equality with other entities
- being passable as a parameter to a procedure/function
- being returnable as the result of a procedure/function
- being constructible at runtime
- being printable
- being readable
- being transmissible among distributed processes
- being storable outside running processes

4. Assertions are a useful defensive programming tool. They can be used to confirm that the arguments to a function are of appropriate types. They are also a useful debugging tool.