Programming in python

BBM103 Introduction to Programming Lab 1
Week 9

Fall 2016
Exceptions Revisited

- Built-in Exceptions
- User-defined Exceptions
Syntax

Here is a simple syntax of try....except...else blocks

```python
try:
    You do your operations here;

except ExceptionI:
    If there is ExceptionI, then execute this block.

except ExceptionII:
    If there is ExceptionII, then execute this block.

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

```python
def divide(x, y):
    try:
        result = x / y
    except ZeroDivisionError:
        print("division by zero!")
    else:
        print("result is", result)
    finally:
        print("executing finally clause")

print("Example 1")
number1=int(input("please enter first number"))
number2=int(input("please enter second number"))
divide(number1,number2)
```

Output:

Example 1
please enter first number 10
please enter second number 0
division by zero!
executing finally clause
```python
import sys

_2x_metni = """"You use one of the 2.x versions of Python. For execute this program, one of the 3.x versions of Python should be installed your computer."""

_3x_metni = "Welcome to program"

try:
    if sys.version_info.major < 3:
        print(_2x_metni)
    else:
        print(_3x_metni)
except AttributeError:
    print(_2x_metni)

while True:
    value = input('Enter an integer: ')
    try:
        value = int(value)
        print('The square of the number you entered is', value**2)
    except ValueError:
        print(value, 'is not an integer')
        # to exit the while loop if s is not an integer
        break
```
Example: `raise`

```python
def f(x):
    return g(x) + 1

def g(x):
    if x < 0: raise ValueError("I can't cope with a negative number here.")
    else: return 5

try:
    print(f(-6))
except ValueError:
    print("That value was invalid.")
```

Output:
That value was invalid.

Example: `raise`

```python
import random

number1 = int(input("please enter a number: "))

if number1 < 0:
    raise Exception("This program can not handle negative numbers")

number2 = int(random.randint(0,100))

try:
    print(number1,"/",number2,"=",number1/number2)
except ZeroDivisionError:
    print("You can't divide a number to zero")
```

Output:
```
please enter a number: 10
10 / 13 : 0.7692307692307693
```
Example: User-defined Exceptions

```python
# class Error is derived from super class Exception
class Error( Exception):
    # Error is derived class for Exception, but
    # Base class for exceptions in this module
    pass

class TransitionError( Error):
    # Raised when an operation attempts a state
    # transition that's not allowed.
    def __init__(self, prev, nex, msg):
        self.prev = prev
        self.next = nex

        # Error message thrown is saved in msg
        self.msg = msg

    try:
        raise TransitionError(2, 3*2, "Not Allowed")

    # Value of Exception is stored in error
    except TransitionError as error:
        print('Exception occurred: ', error.prev, error.next, error.msg)
```

Output:
Exception occurred: 2 6 Not Allowed
Example: User-defined Exceptions

```python
class Error(Exception):
    """Base class for other exceptions"""
    pass

class InputTooSmallError(Error):
    """Raised when the entered alphabet is smaller than the actual one"""
    pass

class InputTooLargeError(Error):
    """Raised when the entered alphabet is larger than the actual one"""
    pass

#you need to guess this alphabet
alphabet = 'm'

while True:
    try:
        abp = input("Enter an alphabet: ")
        if abp < alphabet:
            raise InputTooSmallError
        elif abp > alphabet:
            raise InputTooLargeError
        break
    except InputTooSmallError:
        print("The entered alphabet is too small, try again!")
    except InputTooLargeError:
        print("The entered alphabet is too large, try again!")

print("Congratulations! You guessed it correctly.")
```