Golf Clubs and Exceptions

Wrestling with Python

THE ASSIGNMENT

Golf Scoring Assignment

- This assignment has been structured in the same way as the exam board assignments
- There are three tasks
- Today we are going to focus on the first task

Score Card Information

- The Score Card contains the following information
  - Name of the golfer
  - 18 scores, one each hole
    - The score contains the par for the hole and the number of strokes the player took

Reading in Numbers

- This task involves reading in numbers
- Some of the numbers are in particular ranges
  - The par value should be in the range 3-5
- We can write a method that will read numbers in for us
- We can use exceptions to help with number entry

EXCEPTIONS

Exceptions and Errors

```
vString = input('Enter a number: ')
print(vString)
vInt = int(vString)
```

- This sequence of statements reads in a number from the user
- The text version (the string the user types in) is converted into an integer

Exceptions and Errors

```
vString = input('Enter a number: ')
print(vString)
vInt = int(vString)
```

- This method converts the string into an integer
- It works fine if the user types in sensible text
- But if they type in stupid stuff it fails

Exception errors

```
Traceback (most recent call last):
  File "C:/Users/Rob/PycharmProjects/Exceptions/Exceptions.py", line
4, in <module>
NameError: name 'fred' is not defined
```

- The Python interpreter is confused because we have entered text and it was expecting a number
- This causes the program to stop

Catching Exceptions

```
try:
    vString = input('Enter a number: ')
    print(vString)
    vInt = int(vString)
    print('Number has value: ')
    print (vInt)
except:
    print('Please enter an integer')
```

- We can catch exceptions by using the try – except construction

Catching Exceptions

```
try:
    vString = input('Enter a number: ')
    print(vString)
    vInt = int(vString)
    print('Number has value: ')
    print (vInt)
except:
    print('Please enter an integer')
```

- This is the block of code that might throw an exception
- It contains the call of int that will cause the program to stop

Catching Exceptions

```
try:
    vString = input('Enter a number: ')
    print(vString)
    vInt = int(vString)
    print('Number has value: ')
    print (vInt)
except:
    print('Please enter an integer')
```

- If an exception is “thrown” by the program it will jump to this line
- The code after the except will then handle the exception for us

Catching Exceptions

```
try:
    vString = input('Enter a number: ')
    print(vString)
    vInt = int(vString)
    print('Number has value: ')
    print (vInt)
except:
    print('Please enter an integer')
```

- At the moment we are handling every exception that might be thrown, which is not sensible
  - We might get exceptions we are not expecting

Exception errors

```
Traceback (most recent call last):
  File "C:/Users/Rob/PycharmProjects/Exceptions/Exceptions.py", line
4, in <module>
NameError: name 'fred' is not defined
```

- The exception that we are interested in is “NameError”
  - – I typed in the word fred rather than a number and the program could not make sense of it
- We can put this in explicitly

UNIVERSITY OF Hull

Catching Exceptions by Name

```
try:
    vString = input('Enter a number: ')
    print(vString)
    vInt = int(vString)
    print('Number has value: ')
    print (vInt)
except NameError:
    print('Please enter an integer')
```

- Now our program will only catch NameError exceptions
- Any other errors will cause the program to fail
  - This is what we want, we don’t want to hide errors

Chaining except

```
try:
    vString = input('Enter a number: ')
    print(vString)
    vInt = int(vString)
    print('Number has value: ')
    print (vInt)
except NameError:
    print('Please enter an integer')
except:
    print('Something bad has happened')
```

- You can chain except clauses so that you can pick up different exceptions

Summary

- We can catch exceptions and deal with them
- We could use this to make a method that repeatedly accepted input from the user until they typed in a valid string
- This would be a useful component of the Golf Scoring application