Programming Club Meeting 17 Slides

Exceptions

Exceptions

- An exception is when the program runs into an error that makes running the rest of the code impossible for some reason
  - Said to have been “raised” or “thrown”
  - Ex: division by 0 (no answer, so the rest of the program can’t really run)
- Normally stops a Python program and outputs and error message
- For our programs it’s usually not too bad, but can be a major problem if the code is important and meant to run 24/7
- There’s a large number of different exceptions
  - “Exception” is the base/default one
- Can choose to throw an error with the raise keyword
  - Put type (“Exception” if nothing better) then the text to be outputted

Code

```
1 # Raises a div by 0 error if divisor is 0
2 x = 0
3 if (x == 0):
4     raise ZeroDivisionError("Cannot divide by zero")
5 else:
6     print(5 / x)
```

Output

```
Line 4: ZeroDivisionError: Cannot divide by zero
```

Try

Code

```
# Try
try:
    print("1")
    raise ZeroDivisionError("Error")
    print("Also error")
except:
    pass
print("2")
```

Output

```
1
2
```

- Exceptions thrown inside a try block won't stop the program
  - Exception is still thrown, just doesn't halt the program
- Once exception is raised, the rest of the try block is skipped
- Two options: "look before you leap" or "ask for forgiveness" (Python convention)

Except

- Allows you to handle an error thrown inside the try block
  - Need at least one under a try block
- Often called a “catch” block because it catches the thrown error
- When an exception is raised, the except block runs
- May tell the user what happened and try to fix the problem so that the program can keep running
  - Could also log the error so that it can be dealt with later
- Can have different except statements for different exceptions
  - Default should be last

Output

```
5.0

There was an error
inf
```

Code

```
# Except, won't run
try:
    print(5 / 1)
except:
    print("There was an error")

print()

# Except, will run
try:
    print(5 / 0)
except:
    print("There was an error")
    print(float("Inf"))
```

Code

```
# Different Exceptions for Types
try:
    print(5 / 0)
except NameError:
    print("Name error")
except ZeroDivisionError:
    print("Div by 0")
except:
    print("Other error")
```

Output

```
Div by 0
```

Else and Finally

Code

```
# Else and Finally
num = 5
try:
    x = 1 / num
except:
    # imagine this is a log
    print("Infinite result")
else:
    # imagine this is a log
    print("Finite result")
finally:
    print("Calculation complete")
```

Output

```
Finite result
Calculation complete
```

- Else runs if there is no error
  - Probably not super useful, may be helpful if you only want the 1 key statement that may cause problems in the try block
  - Want to be careful because catching an unexpected error can hide it and cause bugs
- Finally runs regardless of whether or not an error is raised
  - Can be used to close things like files when finished
  - Often not necessary, but groups code together and has some situational uses

Practice Problems

Practice Problem 1: Estimate Square Root

- Src: https://www.101computing.net/square-root-estimation-algorithms/
- Goal: Write a Python program that will estimate the square root of an inputted number via the Babylonian method.
- Relevant Information:
  - The Babylonian method is as follow:
  - Set the variable ‘x’ to 1 with ‘number’ representing the number whose square root is being estimated
  - Set ‘x’ to (x + number / x) / 2, repeat this 99 more times
  - The final ‘x’ value is your estimated square root

Practice Problem 2:

Energy Rating Calculator

- Src: https://www.101computing.net/light-bulb-energy-rating-calculator/
- Goal: Write a Python program that will determine the energy rating of a lightbulb based on user inputs. If the rating is A or G, throw an error.
- Relevant Information:
  - Efficiency ratio = (luminous flux) / (power consumption)
  - Luminous flux is in lumens (lm), power consumption is in kilowatts per 1,000 hours (kW/1000hr)
  - The energy rating is a letter grade based on the ratio
  - A ≥ 210, B ≥ 185, C ≥ 160, D ≥ 135, E ≥ 110, F ≥ 85, G < 85
  - If the rating is A, throw an exception with the text “The rating is too high”
  - If the rating is G, throw an exception with the text “The rating is too low”
  - Ex:
    - 1600 lm, 8 kW/1000hr: B
    - 1800 lm, 5 kW/1000hr: A (error)
    - 1100 lm, 12 kw/1000hr: F
    - 84 lm, 1 kw/1000hr: G (error)

Practice Problem 3: Input Errors

- Goal: Write a Python program that prompts the user to input the number of players for a game. This value should be converted to an integer. If an error is thrown in the coercion or the final value is not 1 or 2, the user should be prompted to enter a new value.

Next Meeting: Review