Methods and Tools in SW Development

Homework 5

Group 1

Caden Austin (CBA169)

Caleb Byers (CTB388)

Darcie Dalafave (DAD514)

Justin Pettiss (JP2973)

Marcus Bridgman (MB3668)

Function Name: displayItem
Number of tests: 7

Items to test:

- Successful Display of Item at Index
  - inputs
    - numbers = [“apple”, “banana”, “coconut”]
    - index = 1
  - outputs
    - Check “banana” is printed
- Out of Bound Index
  - inputs
    - numbers = [“apple”]
    - index = 1
  - outputs
    - Check IndexError is raised
- Check Nonnumerical Index
  - Inputs
    - 1
      - numbers = [“apple”, “banana”, “coconut”]
      - index = “index”
    - 2
      - numbers = [“apple”, “banana”, “coconut”]
      - index = []
    - 3
      - numbers = [“apple”, “banana”, “coconut”]
      - index = {}
  - outputs
    - Check TypeError is raised
- Check Nonlist Numbers
  - Inputs
    - 1
      - numbers = “String”
      - index = 0
    - 2
      - numbers = 123
      - index = 0
  - outputs
    - Check TypeError is raised

Corrections made:

- For the “Display Items” function, a type check was added to ensure that the numbers parameter was a list instead of another iterable object like a string, this change matches the “Check Nonlist Numbers” tests

Tests passed: 7
Tests failed: 0

Function Name: dist
Number of tests: 5

Items to test:

- Successful Distance Calculation
    - Inputs
        - 1
            - Cords_one = (5, 10)
            - Cords_two = (10, 10)
        - 2
            - Cords_one = (5, 10)
            - Cords_two = (5, 17)
        - 3
            - Cords_one = (5, 10)
            - Cords_two = (17, 6.5)
        - 4
            - Cords_one = (5, 10)
            - Cords_two = (-2, -5)
    - outputs
        - 1
            - Returns 5
        - 2
            - Returns 7
        - 3
            - Returns 12.5
        - 4
            - Returns 16.55294535724685
- Check for Bad Point Format
    - inputs
        - Cords_one = (“hello”, “world”)
        - Cords_two = (“world”, “hello”)
    - outputs
        - Raises TypeError

Corrections made:

For the “Dist” function, a type check for the points was added to make sure the x,y are both numeric for both points, this aligns with the “Check for Bad Point Format” test

Tests passed: 5
Tests failed: 0

Function Name: divide
Number of tests: 7

Items to test:

- Successful Division
  - Inputs
    - 1
      - Dividend = 30
      - Divisor = 5
    - 2
      - Dividend = 2
      - Divisor = -1
    - 3
      - Dividend = -20
      - Divisor = -5
    - 4
      - Dividend = 1
      - Divisor = 4
  - outputs
    - 1
      - Print “Your numbers divided is: 6.0”
    - 2
      - Print “Your numbers divided is: -2.0”
    - 3
      - Print “Your numbers divided is: 4.0”
    - 4
      - Print “Your numbers divided is: 0.25”
- Check for Divide by Zero
  - inputs
    - Dividend = 1
    - Divisor = 0
  - outputs
    - Raise ZeroDivisionError
- Division with Floats
  - inputs
    - Dividend = 0.25
    - Divisor = 0.25
  - outputs
    - Print “Your numbers divided is: 1.0”
- Check for Bad Inputs
  - inputs
    - Dividend = “hello”
    - Divisor = 0.25
  - outputs

- Raises TypeError

Corrections made:

For the “Divide” function, input was converted to floats instead of integers so that the input can be more than just whole numbers this aligns with the “Division with Floats” tests. We also added a check to see if the denominator is zero, if it is we raise a ZeroDivisionError, this aligns with the “Check for Divide by Zero” test

Tests passed: 7
Tests failed: 0

Function Name: greetUser
Number of tests: 4

Items to test:

- Successful Greet User
  - inputs
    - 1
      - Name = “Joe Bob Dylan”
    - 2
      - Name = “France Pierre Wright”
  - outputs
    - 1
      - Print “Hello! Welcome … Joe Bob Dylan … you!”
    - 2
      - Print “Hello! Welcome … France Pierre Wright … you!”
- Check for Numbers in String
  - inputs
    - Name = “Leo 5423 Scaper”
  - outputs
    - Raises ValueError
- Check for Non-String Name
  - inputs
    - Name = (123, 456, 7890)
  - outputs
    - Raises TypeError

Corrections made:

For the “Greet User” function, a type check was added for the components of the name to ensure they are string objects, this aligns with “Check for Non-String Name” test. We also added a test to make sure user’s names only consists of alphabetical characters, this aligns with the “Check for Numbers in String” test

Tests passed: 4
Tests failed: 0

Function Name: isPalindrome
Number of tests: 7

Items to test:

- Successful Palindrome Classifying
  - inputs
    - 1
      - Word = “racecar”
    - 2
      - Word = “tacocat”
  - outputs
    - 1
      - Return True
    - 2
      - Return True
- Successful Non-Palindrome Classifying
  - inputs
    - 1
      - Word = “word”
    - 2
      - Word = “thisisnotapalindrome”
  - outputs
    - 1
      - Return False
    - 2
      - Return False
- Capitalized Word Classifying
  - inputs
    - 1
      - Word = “RaCEcAr”
    - 2
      - Word = “TaCOcAt”
  - outputs
    - 1
      - Return True
    - 2
      - Return True
- Check for Bad Type
  - inputs
    - Word = 123
  - outputs
    - Raises TypeError

Corrections made:

With the “is Palindrome” function, a type check was added for the input parameter to ensure it is a string, this aligns with the “Check for Bad Type” test. We also lowercased temp so that if there are uppercase letters in the string it won’t be misclassified, this aligns with the “Capitalized Word Classifying” tests

Tests passed: 7
Tests failed: 0

Function Name: numbers
Number of tests: 7

Items to test:

- Successful Numbers
    - Inputs
        - 1
            - Dividend = 30
            - Divisor = 5
        - 2
            - Dividend = 2
            - Divisor = -1
        - 3
            - Dividend = -20
            - Divisor = -5
        - 4
            - Dividend = 1
            - Divisor = 4
    - outputs
        - 1
            - Return 6.0
        - 2
            - Return -2.0
        - 3
            - Return 4.0
        - 4
            - Return 0.25
- Check for Divide by Zero
    - inputs
        - Dividend = 1
        - Divisor = 0
    - outputs
        - Raise ZeroDivisionError
- Division with Floats
    - inputs
        - Dividend = 0.25
        - Divisor = 0.25
    - outputs
        - Return 1.0
- Check for Bad Inputs
    - inputs
        - Dividend = “hello”
        - Divisor = 0.25
    - outputs

- Raises TypeError

Corrections made:

With the “numbers” function, a try/except was added that will pick up a ZeroDivisionError and a TypeError if invalid values are provided to the function, this aligns with the “Check for Divide by Zero” and “Check for Bad Inputs” tests

Tests passed: 7
Tests failed: 0

Function Name: openFile
Number of tests: 5

Items to test:

- Successful File Open
  - inputs
    - File = “testing.txt”
  - outputs
    - Print “File opened.”
- Check Nonexistent File
  - inputs
    - File = “nonexistant.txt”
  - outputs
    - Raises FileNotFoundError
- Check Bad File Formats
  - inputs
    - 1
      - File = 123
    - 2
      - File = []
    - 3
      - File = {}
  - outputs
    - 1
      - Raises TypeError
    - 2
      - Raises TypeError
    - 3
      - Raises TypeError

Corrections made:

With the “Open File” function, a try/except was added that will pick up a FileNotFoundError and a TypeError if invalid values or a nonexistent filename are provided to the function, this aligns with the “Check Nonexistent File” and “Check Bad File Formats” tests

Tests passed: 5
Tests failed: 0

Function Name: sq
Number of tests: 7

Items to test:

- Successful Square Root
    - inputs
        - 1
            - Num = 4
        - 2
            - Num = 9
        - 3
            - Num = 20
    - outputs
        - 1
            - Return 2
        - 2
            - Return 3
        - 3
            - Return 4.47213595499958
- Successful Square Root with Floats
    - inputs
        - 1
            - Num = 4.8
        - 2
            - Num = 9.0
        - 3
            - Num = 15.75
    - outputs
        - 1
            - Return 2.1908902300206643
        - 2
            - Return 3
        - 3
            - Return 3.968626966596886
- Checks Negative Input
    - input
        - Num = -1
    - output
        - Raises ValueError

Corrections made:

For the “Sq” function, a value check was added to check if the number provided is negative or not, this prevents getting into complex numbers and aligns with the “Checks Negative Input” test

Tests passed: 7
Tests failed: 0