Thursday Sep 19, 2024

In-class Handout

COSC 101C Intro to Computing I

Prof. Forrest Davis

Name:

Discuss and complete the following questions with the person nearest you. You **may** be asked to share your thoughts with the class.

1. Write good docstrings for the following functions:

```python
import math
def funcA(radius: float) -> float:
    return math.pi * (radius ** 2)
```

```python
def funcB(hours: int, minutes: int, seconds: int) -> int:
    minutes = minutes + (hours * 60)
    seconds = seconds + (minutes * 60)
    return seconds
```

```python
import math
def funcA(radius: float) -> float:
    """ Returns the area of a circle

    Parameters:
        radius (float): Radius of the circle
    Returns:
        float: Area of the circle
    """
    return math.pi * (radius ** 2)
```

```python
def funcB(hours: int, minutes: int, seconds: int) -> int:
    """ Returns the total seconds

    Parameters:
        hours (int): Number of hours
        minutes (int): Number of minutes
        seconds (int): Number of seconds
    Returns:
        seconds (int): Total seconds
    """
    minutes = minutes + (hours * 60)
    seconds = seconds + (minutes * 60)
    return seconds
```

2. What is the output of the following code snippet? Use a trace table for `square()`

```
def square(original_number: int) -> int:
    running_total = 0
    for counter in range(original_number):
        running_total = running_total + original_number
    return running_total

def main() -> None:
    to_square = 10
    result = square(to_square)
    print("The result of", to_square, "squared is", result)
main()
```

Output: `The result of 10 squared is 100`

iteration	counter	running_total
1	0	10
2	1	20
3	2	30
4	3	40
5	4	50
6	5	60
7	6	70
8	7	80
9	8	90
10	9	100

3. What is the output of the following code snippet? Use a trace table where appropriate.

```
def newtonSqrt(number: int, num_guesses: int) -> float:
    approx = 0.5 * number
    for i in range(num_guesses):
        approx = 0.5 * (approx + number/approx)
        print(i, approx)
    return approx
def main() -> None:
    print(newtonSqrt(100, 2))
main()
```

```
0 26.0
1 14.923076923076923
14.923076923076923
```

Output: `14.923076923076923`

iteration	approx
0	50.0
1	26.0
2	14.923076923076923

4. Rearrange the following lines of code to create a program that will add up the first n odd numbers where n is provided by the user.

```
        oddnum += 2 #1
    return sum #2
    oddnum = 1 #3
def oddTotal(n: int) -> int: #4
    total = oddTotal(n) #5
    for i in range(n): #6
main() #7
    sum = 0 #8
        sum += oddnum #9
    n = int(input("How many odd numbers would you like to add together? ")) #10
    print(total)#11
def main() -> None:#12
```

```
def oddTotal(n: int) -> int:
    sum = 0
    oddnum = 1
    for i in range(n):
        sum += oddnum
        oddnum += 2
    return sum

def main() -> None:
    n = int(input("How many odd numbers would you like to add together? "))
    total = oddTotal(n)
    print(total)

main()
```

5. Write a program that flips a coin as many times as the user requested and outputs the percentage of flips that were heads.

```
import random
def flipCoin(numFlips: int) -> float:
    """Returns proportion of numFlips that are heads"""
    total_heads = 0
    for i in range(numFlips):
        flip = random.randint(0, 1)
        total_heads += flip
    return total_heads/numFlips

def main() -> None:
    numFlips = int(input("How many times do you want the coin flipped? "))
    proportion = flipCoin(numFlips)
    print(f"{round(proportion*100, 2)}% of the flips were heads")

main()
```