

Exam 2 review

COSC 101, Introduction to Computing I, 2021-10-18

Announcements

- Exam 2 Wednesday during class
 - Review materials posted on Moodle
- Homework 6 due Thursday, October 28

Exam topics

- functions: defining and calling
- how functions work: parameter passing, return values, variable scope
- docstrings and doctests
- nested for loops
- while loops
- Plus topics from the last exam:
 - Types (int, str, float, bool)
 - Expressions, variables, and assignment
 - Interactive programs (input, print)
 - Modules (import, random, math, turtle)
 - Conditional statements (if/elif/else)
 - Repetition with definite loops (for)
 - Accumulator patterns

Code tracing

What is the output of the following programs?

```
#1)
def funcA(f):
    f = f * 2
    return f * 3

def funcB(g):
    g = g + 2
    return g + 3

h = 5
i = funcA(h)
print(i)
j = funcB(funcA(i))
print(j)
```

30
185

```
#2)
def funcC(m):
    n = m + 1
    o = m * r
    q = m - 1
    print(n, o, q)
    return m // r
```

```
q = 1
r = 2
r = funcC(8)
print(q, r)
```

9 16 7
1 4

```
#3)
s = "ah"
for i in range(1,4):
    t = ""
    for j in s:
        t = t + (j * i)
    print(t)
```

```
ah
aahh
aaahhh
```

```
#4)
def funcD(x):
    y = 0
    while x > 1:
        x = x / 2
        y += 1
    return y
print(funcD(5))
print(funcD(32))
```

```
3
5
```

```
#5)
def funcE(x):
    y = 0
    while y <= x:
        y = y * 2
    return y
print(funcE(5))
```

Writing code

#1) Write a function called `get_amount` that takes a minimum bill size (as an integer) asks a user how much money to withdraw from an Automated Teller Machine (ATM) until the user enters an amount that is a multiple of the minimum bill size; the function should return the amount (as an integer). You can assume the user will only enter digits.

```
def get_amount(minimum):
    amount = 0
    while amount <= 0 or amount % minimum != 0:
        amount = int(input("How much money would you like to withdraw? "))
    return amount
```

#2) Write a function called `print_multiples` that takes a positive integer `n` and prints the first multiple of 1, the first two multiples of 2, the first three multiples of 3, etc. up to and including the first `n` multiples of `n`. For example, `print_multiples(5)` should output:

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

```
def print_multiples(n):
    """
    >>> print_multiples(5)
    1
    2 4
    3 6 9
    4 8 12 16
    5 10 15 20 25
```

```

"""
for i in range(1, n+1):
    for j in range(1,i+1):
        print(i*j, end=" ")
    print("")

```

#3) Write a function called `pete_and_repeat` that asks the user the following riddle: Pete and Repeat were on a boat. Pete fell out, who was left on the boat? If the user enters (Repeat), then the program should again ask the user to answer the riddle. If the user enters something else, then the program should print (Wrong answer) and again ask the user to answer the riddle.

```

def pete_and_repeat():
    answer = ""
    while answer == "Repeat" or answer != "Repeat":
        answer = input("Pete and Repeat were on a boat. Pete fell out, who was left on the boat? ")
        if answer != "Repeat":
            print("Wrong answer")
pete_and_repeat()

```

#4) Customers at the Cost Mincer grocery store have complained about the long checkout lines, so the store manager wants to measure the average checkout time per item. Your task is to write a program that, for each customer, asks for the number of items purchased and the time required to complete the transaction. The program should continue to ask for inputs until 0 is entered for the number of items. After all inputs have been entered, the program should output the average checkout time per item.

Example input/output:

```

Number of items? 20
Checkout time (seconds)? 80
Number of items? 60
Checkout time (seconds)? 160
Number of items? 0
Average checkout time per item is 3.0 seconds

```

```

done = False
total_items = 0
total_time = 0
while not done:
    items = int(input("Number of items? "))
    if items == 0:
        done = True
    else:
        time = int(input("Checkout time (seconds)? "))
        total_items += items
        total_time += time

average = total_time / total_items
print("Average checkout time per item is", average, "seconds")

```

```

Number of items? 20
Checkout time (seconds)? 80
Number of items? 60
Checkout time (seconds)? 160
Number of items? 0

```

```

Average checkout time per item is 3.0 seconds

```

#5) The value of the mathematical constant pi (π) can be estimated using the Madhava-Leibniz series:

$$\pi = 4/1 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 \dots$$

Write a function called `estimate_pi` that uses the Madhava-Leibniz series to estimate pi until the estimate rounded to two decimal digits is 3.14.

```
def estimate_pi():
    """
    >>> estimate_pi()
    3.1350137774059244
    """
    add = True
    estimate = 0.0
    i = 1
    while round(estimate, 2) != 3.14:
        if add:
            estimate = estimate + 4/i
            add = False
        else:
            estimate = estimate - 4/i
            add = True
        i = i + 2
    return estimate
```

#6) Write a function called `print_monthly_calendar` that takes the number of days in a month and the day of the week on which the first day falls and prints a monthly calendar. For example `print_monthly_calendar(30, 4)` should output:

```

        01 02 03 04
05 06 07 08 09 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

```
def print_monthly_calendar(days, day_of_week):
    """
    >>> print_monthly_calendar(31, 1)
    01 02 03 04 05 06 07
    08 09 10 11 12 13 14
    15 16 17 18 19 20 21
    22 23 24 25 26 27 28
    29 30 31
    >>> print_monthly_calendar(30, 4)
        01 02 03 04
    05 06 07 08 09 10 11
    12 13 14 15 16 17 18
    19 20 21 22 23 24 25
    26 27 28 29 30
    """
    for i in range(day_of_week-1):
        print(" ", end='')
    for d in range(1,days+1):
        if day_of_week == 8:
            print()
            day_of_week = 1
        if d < 10:
            print("0" + str(d)+" ", end='')
        else:
            print(str(d)+" ", end='')
        day_of_week = day_of_week + 1
```