CSCI 150: Exam 3 – In Class

Monday, April 17, 2023

You have 50 minutes to complete this exam. **You are not allowed to use your notes, textbook, phone or computer. Partial credit for incorrect answers can be given only if you show your work.**

Print your name : _____

For each piece of code shown, trace, showing the function stack and heap.

1.

```
def main1():
    a = [2, 5, 7]
    b = [2, 5, 7]

    c = {'a': 7, 'b': 9, 'c': 2}
    d = c
    i = 0
    for key in c:
        if c[key] in a:
            c[key] = i * 3
            b[i] = i * 3
        i += 1

    print(a)
    print(b)

    print(c)
    print(d)

    print(i)

main1()
```

2.

```
def f1(d: Dict[str, int], s: str) -> int:
    n = 0
    for key in d:
        if s not in key:
            n += d[key]

    return n


def main2():
    a = ['x', 'y', 'z']

    c = {'syzygy': 12, 'yoyo': 7}
    d = {}

    for char in a:
        d[char] = f1(c, char)

    print(a)

    print(c)
    print(d)

main2()
```

3.

```
def aa(n: int) -> int:
    if n < 5:
        return n
    else:
        return n + 3


def main3():
    lst = [8, 2, 4, 9]
    mst = lst
    pst = []

    for i in range(len(mst)):
        pst.append(mst[i])
        mst[i] = aa(mst[i])

    print(lst)
    print(mst)
    print(pst)

main3()
```

4. Write a function `easy_as_pie` which takes in a lower case string (which might have spaces or punctuation) `s`. It should return a dictionary keyed on the three letters `'p'`, `'i'`, `'e'` and return for each a count of the number of occurrences of each in `s`.

   For example:

   - `easy_as_pie('special')` should return `{'p': 1, 'i': 1, 'e': 1}`
   - `easy_as_pie('hello there, peter; i am happy to see you')` should return
     `{'p': 3, 'i': 1, 'e': 7}`
   - `easy_as_pie('xyz')` should return
     `{'p': 0, 'i': 0, 'e': 0}`
   - `easy_as_pie('pipipipipie')` should return
     `{'p': 5, 'i': 5, 'e': 1}`

```
def easy_as_pie(s: str) -> Dict[str, int]:
    # write your code here
```

5. Write a class `Pool`, whose objects represent swimming pools. Each pool is built with a parameter `capacity`, which can be any positive integer (you do not need to check that the given value is positive) and initialized with zero water. A pool should have the following attributes:

   - `capacity` – set as a parameter when the pool is created
   - `amount` – the current amount of water in the pool, which is initially set to 0.

   Pools should have the following methods:

   - `fill(n: int)` which adds n units of water to the pool. If more is added than the pool can hold, the extra overflows, and the value of `amount` should be set to the maximum capacity. (You can assume that n will always be positive here.)
   - `drain()` which removes all water from the pool.
   - `swim()` which prints `'It is a beautiful day for a swim!'` *unless* the pool is currently filled with less than 10 units of water. In that case, it should print the statement

     `The pool needs more water for you to swim. It currently contains xx`,

     where `xx` is the current amount of water in the pool.

   The following is an example of the code running in the console:

```
>>> a = Pool(50)
>>> a.swim()
The pool needs more water for you to swim. It currently contains 0.
>>> a.fill(35)
>>> a.swim()
It is a beautiful day for a swim!
>>> a.drain()
>>> a.swim()
The pool needs more water for you to swim. It currently contains 0.
>>> a.fill(6)
>>> a.swim()
The pool needs more water for you to swim. It currently contains 6.
>>> a.fill(4)
>>> a.swim()
It is a beautiful day for a swim!
```

See Next Page!!!!!

```
class Pool:

    def __init__(self, capacity: int):
        # write your code here









    def fill(self, n: int):
        # write your code here









    def drain(self):
        # write your code here









    def swim(self):
        # write your code here
```