CS 1: Introduction to Computer Programming

Recitation 1: Prepare to Battle Ships Solutions

Welcome to your first CS 1 "problem solving session" of the term! These sessions aim to review some of the material covered in lecture to help you identify common mistakes we've seen students make in the past (and programming) and lead you through some practice exercises with help!

Common Errors

Strings are Immutable

Input/Output Example

```
1 s = "hello"
2 s[0] = 'H'  # Causes TypeError as str objects do not support item assignment
3 s.replace('h', 'H')
4 print(s)  # s was not modified as we did not catch its return value
5 s = s.replace('h', 'H')
6 print(s)  # s was correctly assigned the returned string from the replace method
```

```
>> TypeError
>> hello
>> Hello
```

Strings can Only be Concatenated to Strings

Input/Output Example

```
1 grad_yr = 2028
2 print("I will graduate in " + grad_yr)  # Causes TypeError as cannot concatenate int to str
3 print("I will graduate in " + str(grad_yr))  # Casting allows us to concatenate
```

```
>> TypeError
>> I will graduate in 2028
```

Strings and Lists are 0-indexed

Input/Output Example

```
1 lst1 = ["printme", "-", "-"]
2 print(lst1[0])
3 print(lst1[1])
```

```
>> printme
>> -
```

Boolean Logic is Important in Conditional Statements

Input/Output Example

```
1  time_left = 0
2  enemies_killed = True
3
4  # Incorrect as currently only one condition suffices
5  if time_left > 0 or enemies_killed:
6      print("You won!")
7  else:
8      print("Unfortunately, you lost!")
9
10 # Correct as both conditions need to be fulfilled
11 if time_left > 0 and enemies_killed:
12     print("You won!")
13 else:
14     print("Unfortunately, you lost!")
```

```
>> You won!
>> Unfortunately, you lost!
```

Input/Output Example

```
1  grade = 101
2  # Incorrect as always True for integers
3  if grade > 59 or grade < 101:
4      print("Pass!")
5  else:
6      print("Fail!")
7
8  # Correct, requires grade to be within a range
9  grade = 101
10 if 60 <= grade =< 100:
11     print("Pass!")
12 else:
13     print("Fail!")
```

```
>> Pass!
>> Fail!
```

Append is used to Add Elements to a List

Input/Output Example

```
1  nbrs1 = [1,2,3,4]
2  nbrs2 = [1,2,3,4]
3  print(nbrs1 + 5)
4  nbrs2.append(5)  # Modifies the existing list in place
5  print(nbrs2)
```

```
>> TypeError
>> [1,2,3,4,5]
```

Problem Solutions

String Concatenation

Worked Example

Red Door prices

Print a sentence presenting the price of an `iced_matcha_large` at Red Door.

```
1    iced_matcha_large = 6.25
2    print("Large iced matcha lattes cost " + str(iced_matcha_large) + "dollars at Red Door.")
```

Faded Example

Introductions

Print an introduction of yourself using `name`, `age`, and `house_or_residence`.

```
1    name = name
2    age = age
3    house_or_residence = house
4    print("Hi, my name is " + name + " , I am " + str(age) + " and live in " + house + " .")
```

Your Turn

Red Door order

Print a sentence detailing the **price of a few Red Door items and the total amount for the order**.
You should **store the prices in separate variables beforehand**.
The total amount should **not be hardcoded**.

Sample Outputs:

- `I pay 6.5 dollars for a 2.5 dollar croissant and a 4 dollar iced coffee!`
- `I like to order an avocado toast for 7 dollars with an apple for 1 dollar, bringing my total order to 8 dollars.`

Solution:

```
1  ham_and_cheese = 8
2  lemonade = 5
3  print("For a ham and cheese that costs " + str(ham_and_cheese) " dollars and a lemonade that costs " +
       str(lemonade) + "dollars, the total cost of my order comes down to " + str(lemonade +
       ham_and_cheese) " dollars.")
```

Indexing strings, lists, and tuples

Worked Example

Last character

Print the last character of the following string.

```
1    nbrs = "1234567890"
2    print(nbrs[len(nbrs)-1]) # Why do we have the -1? What does len(nbrs) return?
```

Find the i^{th}...

Print the fifth class of the schedule.

```
1    student_classes = ["MA1a", "PH1a", "CH1a", "WR2", "CS1", "CS9"]
2    print(student_classes[4]) # Why 4 and not 5?
```

Faded Example

Gold!

Print the variable treasure by accessing it from the treasure_maps.

```
1    treasure = "gold!"
2    treasure_map1 = ["", "", "", treasure, ""]
3    print(treasure_map1[3])
4    treasure_map2 = [[[],[]],[[],treasure],[]]
5    print(treasure_map2[1][1])
6    treasure_map3 = [[[],[]],[[],(treasure)],[]]
7    print(treasure_map3[1][1][0])
8    treasure_map4 = ((0,1),(0,1,0),(1,0,(1,0),(1,0,1,0,treasure)))
9    print(treasure_map4[2][3][4])
```

Your Turn

2D boards

Make a **2D board of size 3x3** with each cell containing the coordinate pair of that cell in the board. Then, **print the coordinate pair located in the last cell** (lower right) of the board.

Solution:

```
board = [[(0,0), (1,0), (2,0)],
         [(0,1), (1,1), (2,1)],
         [(0,2), (1,2), (2,2)]]
print(board[2][2])
```

Nested Loops

Worked Example

Pairs

Print all pairs of numbers, (x, y) where $x \leq 10$ and $y \leq 20$.

```
for i in range(11):    # Why did we use 11 insteead of 10?
   for j in range(21): # Could we use x instead of j in this line?
      print((i, j))    # Why did we use the extra parentheses?
```

Faded Example

Triples 1

Print all triples of numbers, (x, y, z) where $x \leq 10$, $y \leq 20$, and $z \leq 30$.

```
for z in range( 11 ):
   for  anything  in range(21):
      for baz in range(21):
         print(  (z, anything, baz)  )
```

Your Turn

Triples 2

Print all triples of numbers, (x, y, z) where $x < y < z < 20$. Order doesn't matter.

Solution:

```
for z in range(20):
   for y in range(z):
      for x in range(y):
         print((x, y, z))
```

Conditionals

Worked Example

Pokemon evolution

To evolve, most Pokémon need to have reached a certain level. For instance, Charmander needs to be at least level 16 to evolve into Charmeleon, and Charmeleon evolves into Charizard at level 36.
Write a short program that prints what evolution Charmander has reached, if any.

```
level = 37
if level >= 36: # Does the ordering of the conditonals matter? Why?
   print("It's a Charizard!")
elif level >= 16:
   print("It's a Charmeleon!")
else:
   print("It's a Charmander!")
```

Faded Example

Movie ticket

A movie theater offers tickets at $\$8$ for children under 18, $\$6$ for seniors starting at 65 years of age, and $\$10$ for adults. `Members` also receive 30% off. Find the price of a ticket based on `age` and `member` status.

```
age = 19
member = True

if [member] :
   if [age < 18] :
      return 8 * 0.7
   elif [age >= 65] :
      return 6 * 0.7
   else:
      return [10] * 0.7
else:
   if [age < 18] :
      return 8
   elif [age >= 65] :
      return 6
   else:
      return [10]
```

Your Turn

Amusement park

Write a short program to determine whether you can go to the amusement park or not!
You should define 4 boolean variables that will serve as criteria: `is_weekend`, `is_sunny`, `friends_available`, and `have_money`.
Your conditions to go to the amusement park are that it is either `is_sunny` or `is_weekend`, and that `have_money` and `friends_available`.
Print "I can go!" if your requirements are fulfilled and "I can't go..." otherwise and feel free to initialize your 4 variables to whatever you would like!

Solution:

```
1 is_sunny = False
2 is_weekend = True
3 friends_available = True
4 have_money = True
5
6 if (is_sunny or is_weekend) and friends_available and have_money:
7    print("I can go!")
8 else:
9    print("I can't go...")
```