

Iteration II

Meta Activity: Team Disruptions

Common disruptions to learning in teams include: talking about topics that are off-task, teammates answering questions on their own, entire teams working alone, limited or no communication between teammates, arguing or being disrespectful, rushing to complete the activity, not being an active teammate, not coming to a consensus about an answer, writing incomplete answers or explanations, ignoring ideas from one or more teammates.

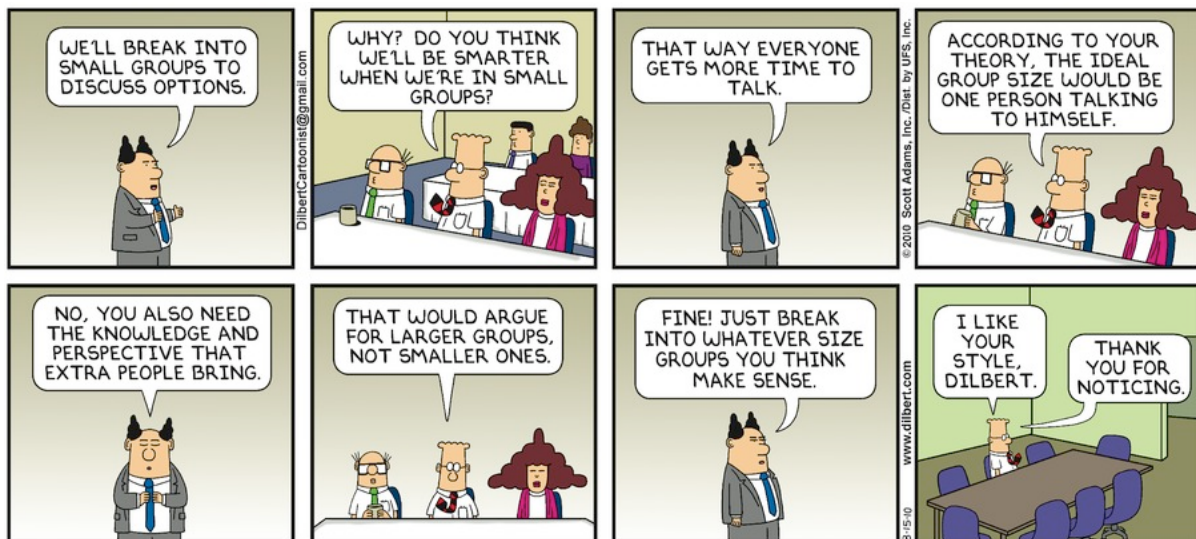
1. Pick four of the disruptions listed above. For each one, find something from the role cards that could help improve the team's success. Use a different role for each disruption.

a) Manager:

b) Presenter:

c) Recorder:

d) Reflector:



Functions and for loops practice

2. Evaluate the following for loop:

```
1 def main() -> None:
2     for i = 0 to 5:
3         print(i, end = "")
4 main()
```

3. What is the output of the following code snippet? Draw the loop table for each for loop.

```
1 def main() -> None:
2     for i in range(4):
3         print("i:", i+1)
4
5     for j in range(4,0,-1):
6         print("j: "+str(j))
7 main()
```


4. What is the output for print_pattern('cat',3)?

```
1 def print_pattern(word: str, copies: int) -> None:
2     phrase = " * "
3     for count in range(copies):
4         phrase = phrase + word + " * "
5     print(phrase)
```


5. What is the output for `print_pattern(3, 'hi')`?

```

1 def print_pattern(cnt: int, instr: str) -> None:
2     mystr = "Be"
3     for i in range(cnt):
4         mystr = mystr + instr * i + "!" * (cnt-i)
5         print(mystr)

```


6. What is the output for `print_pattern(3)`?

```

1 def ants(repeat: int) -> None:
2     ants = 1
3     for i in range(repeat):
4         print('The ants go marching', i, 'by', ants)
5         ants = ants * 2
6         print('hurrah ' * (i + 1))
7     print('And ' + str(ants) + ' ants go marching down into the ground')

```


7. Write a function called `cheer` that asks a user for his/her name and prints a “cheer” in a particular format. If the user types “Fred”, the output should be:

```
F !  
rr !  
eee !  
dddd !
```

That is, the program should print a number of copies of each letter of the name on a separate line. The number of copies should increase with the number of lines printed. Each line should end with an exclamation point – and there should be no space between the exclamation point and the letters of the name. **Hint:** nested loops are not needed for this problem. As another example, if the user types the name “Abi”, the output should be:

```
A !  
bb !  
i i i !
```

8. Write a function called `cheer` that asks a user for a team’s name and prints a “cheer” in a particular format. That is, the program should print, on separate lines, each letter of the name with “Gimme an” before it, and an exclamation point afterward, then print one line with the team’s name repeated for the same number of letters in the name, with exclamation points after each occurrence of the name. If the user types “Gate”, the output should be:

```
Gimme an G !  
Gimme an a !  
Gimme an t !  
Gimme an e !  
Gate!Gate!Gate!Gate!
```

9. Write a function called `get_int` that takes in a list of digits, calculates and returns the number corresponding to the digits in the list.

For example, `get_int([3,7,1])` must return 371.

10. Write a function that take in a list and a count and returns a list with count elements picked at random from the input list (with replacement):

11. Write a function that take in a list and a count and returns a list with count elements picked at random from the input list (without replacement):

12. Write a function `doubleReverse` that takes a list as a parameter. You can assume that every item in the list is a string. The function should return a new list that contains the strings in reverse order and with each string reversed. You can assume the list is non-empty.

Example: `doubleReverse(['moo', 'cow', 'moo!'])` returns `['!oom', 'woc', 'oom']`.