while loops

COSC 101: Intro to Computing I
October 16, 2017
Spring 2019 Wales Study Group

Director: Ahmet Ay, Department of Biology and Mathematics
For more information, please email aay@colgate.edu

Information Sessions:
Monday, October 23rd at 7pm in 101 McGregory Hall and
Tuesday, October 24th at 4pm in 301 Olin Hall
Compare and contrast **for** and **while** loops:

Compare and contrast **while** loops and **ifs**:
What is the output of the following `while` loop?

```
num = 2
while num < 50:
    print(num, end=" ")
    num = num * 2
```

2 4 8 16 32
Rewrite this `while` loop as a `for` loop:

```
num = 2
while num < 50:
    print (num , end=" ")
    num = num * 2

num = 2
for i in range(5):
    print (num , end=" ")
    num = num * 2
```
Rewrite this `for` loop as a `while` loop:

```python
s = "abcxyz"
for ch in s:
    print (ch, end=" ")

s = "abcxyz"
i = 0
while i < len(s):
    print (s[i], end=" ")
i += 1
```
Rewrite this \texttt{for} loop as a \texttt{while} loop:

\begin{verbatim}
L = ['a', 'b', 'c', 'd', 'e']
for i in range(range(L)):
    print(L[-(i+1)], end=" ")

L = ['a', 'b', 'c', 'd', 'e']
i = len(L)-1
while i >= 0:
    print(L[i], end=" ")
i -= 1
\end{verbatim}
Rewrite this `for` loop as a `while` loop:

```python
for i in range(5):
    print(2 * (i + 1), end=" ")

i = 2
while i < 10:
    print(i, end=" ")
    i += 2
```
When should you use a *for* loop? And when should you use a *while* loop?
Write a program that asks the user for a lowercase string and prints each letter up to but not including the first vowel.

Use a while loop and an accumulator variable. Do not use find() or string slicing.

```python
s = input("Gimme a string: ")
i=0
while i < len(s) and s[i] not in 'aeiou':
    print(s[i], end=" ")
i += 1
```
Write a function `significance` that takes an int `(n)` and uses a `while` loop to return a string containing the digits of `n` rearranged from least to most significant.

```python
def significance(n):
    """ (int) -> str
    Takes an int (n) and returns a string containing the digits of n rearranged from least to most significant.
    >>> significance(4982)
    '2894'
    >>> significance(12453451)
    '15435421'
    """
```
Write a function `significance` that asks the user for a lowercase string and prints each letter up to but not including the first vowel.

```python
def significance(n):
    result = ""
    while n > 0:
        last_digit = n % 10
        result += str(last_digit)
        n = n//10
    return result
```
Recall the HW 04 Game Winner program that kept track of player’s scores during a game and computed the winner at the end. In that homework there was a maximum of three rounds, now that we know while loops we can write a version with unlimited rounds.

Write the following three functions:

1. `update_scores(scores, round_num)`
2. `game_winner(scores, points_needed)`
3. `play_game()`
def update_scores(scores, round_num):
    ''' (list<int>, int) -> list<int>
    Takes a list of scores and the number of the current round. Gets scores for each player for this round, and returns a new list of updated scores.
    ...
def update_scores(scores, round_num):
    # print round number label
    print("
ROUND "+str(round_num)+":")

    # create copy of score list (accumulator variable)
    new_score_list = score_list[:]

    # update each player's score
    for i in range(len(score_list)):
        score = int(input("Player "+str(i+1)+" score: "))
        new_score_list[i] += score

    # return updated scores
    return new_score_list
def game_winner(scores, points_needed):
    ''' (list<int>, int) \rightarrow int or bool
    Takes a list of scores (score_list) and the maximum number of points
    needed to win (points_needed). Determine and returns the index of
    the player who has won or False if there is no winner yet.
    
    But what if there’s a tie?
def game_winner(scores, points_needed):
    ''' (list<int>, int) -> list<int> or bool
    Takes a list of scores (score_list) and the maximum number of points needed to win (points_needed).
    Determine and returns the index(es) of the player(s) who have won or False if there is no winner yet.
    ...
def game_winner(scores, points_needed):
    # get highest score
    max_score = max(score_list)

    # check if no one has won yet
    if max_score < points_needed:
        return False

    # determine index(es) of winner(s)
    winners = []
    for i in range(len(score_list)):
        if score_list[i] == max_score:
            winners.append(i)
    return winners
def play_game():
    ''' Controls game play. Asks the user for the number of players and the maximum score needed to win. Plays the game until a player wins. Prints the number(s) of the winning player(s) and their score, along with the scores of all the players. '''
def play_game():
    # setup game
    num_players = int(input("How many players are playing? "))
    points_needed = int(input("How many points are needed to win? "))

    # create initial scores
    scores = []
    for i in range(num_players):
        scores.append(0)

    # play as long as there is no winner
    winner = False
    round_num = 1
    while winner == False:
        scores = update_scores(scores, round_num)
        winner = game_winner(scores, points_needed)
        round_num += 1
def play_game():

    <<insert code from previous slide>>

    # announce winner(s)
    if len(winner) == 1:
        print("\nThe winner is Player " + str(winner[0]+1)+"!")
    else:
        print("\nThe winners are:")
        for i in winner:
            print(" - Player " + str(winner[i]+1))

    # print final scores
    print("\nThe final scores are:")
    for i in range(num_players):
        print(" - Player " + str(i+1)+": " + str(scores[i]))