CSc 110
Score Keeper

Benjamin Dicken

Score Keeper

- What if we wanted a program that we could use to keep track of player scoring during a sporting event
  - Each time points were scored, associated the player with the points scored
  - Be able to check how many points a player has scored

- What are the challenges?

Score Keeper

```
Enter command: ADD Booker 3
Enter command: ADD James 2
Enter command: ADD Ayton 2
Enter command: ADD Booker 2
Enter command: ADD Booker 2
Enter command: GET Booker
Booker has 7 points.
Enter command: GET James
James has 2 points.
Enter command: GET Ayton
Ayton has 2 points.
Enter command: EXIT
(show scores)
```

Could we have implemented this before knowing about lists?

Why are lists specifically useful to solve this problem?

```
Enter command: ADD Booker 3
Enter command: ADD James 2
Enter command: ADD Ayton 2
Enter command: ADD Booker 2
Enter command: ADD Booker 2
Enter command: GET Booker
Booker has 7 points.
Enter command: GET James
James has 2 points.
Enter command: GET Ayton
Ayton has 2 points.
Enter command: EXIT
(show scores)
```

```
players = []
points = []

def get_command():
    ''' Get a command from user '''

def get_index(player):
    ''' Get the index of the player '''

def show_scores():
    ''' Show all player scores '''

def main():
    while True:
        command = get_command()
        # ???????

main()
```

Implement get_command

```
def get_command():
    '''
    Should accept an input from the user and split the
    command into a list of strings and return the list.
    '''
```

Implement get_command

```
def get_command():
    '''
    Should accept an input from the user and split the
    command into a list of strings and return the list.
    '''
    user_input = input('Enter command: ')
    return user_input.split(' ')
```

```
players = []
points = []

def get_command():
    user_input = input('Enter command: ')
    return user_input.split(' ')

def get_index(player):
    ''' Get the index of the player '''

def show_scores():
    ''' Show all player scores '''

def main():
    while True:
        command = get_command()
        # ???????

main()
```

in and find

- Can use the **in** keyword with lists
- Can use the **index()** function

Implement get_index

```
def get_index(player): # player should be a string
    '''
    This function should determine the index in the players
    and points lists that corresponds to the player
    Parameter. If the player does not exist in the list,
    add to points and players.
    The player parameter variable is an indicator of one of
    the athletes playing in the sporting event.
    '''
```

Implement get_index

```
def get_index(player):
    '''
    This function should determine the index in the players
    and points lists that corresponds to the player
    parameter.
    The player parameter variable is an indicator of one of
    the athletes playing in the sporting event.
    '''
     if player not in players:
         players.append(player)
         points.append(0)
     return players.index(player)
```

```
def get_command():
    user_input = input('Enter command: ')
    return user_input.split(' ')

def get_index(player):
    if player not in players:
        players.append(player)
        points.append(0)
    return players.index(player)

def show_scores():
    ''' Show all player scores '''

def main():
    while True:
        command = get_command()
        # ???????

main()
```

Implement show_scores

```
def show_scores():
    '''
    Show the scores for each player. Print out the results.
    What type of loop should be used?
    Example output:
      Booker: 17
      Durant: 13
      Ayton: 9
    '''
```

Implement show_scores

```
def show_scores():
    '''
    Show the scores for each player. Print out the results.
    '''
    for i in range(len(players)):
        print(players[i] + ': ' + str(points[i]))
```

```
players = []
points = []

def get_command():
    user_input = input('Cmd: ')
    return user_input.split(' ')

def get_index(player):
    if player not in players:
        players.append(player)
        points.append(0)
    return players.index(player)

def show_scores():
    for i in range(len(players)):
        print(players[i] + ': ' + str(points[i]))
```

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            # ???????
        elif command_type == 'GET':
            # ???????
        elif command_type == 'EXIT':
            # ???????
        else:
            # ???????


main()
```

Activity

Implement the missing components

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            # (A)
        elif command_type == 'GET':
            # (B)
        elif command_type == 'EXIT':
            # (C)
        else:
            # (D)
```

```
players = []
points = []

def get_command():
    . . .
def get_index(player):
    . . .
def show_scores():
    . . .
```

Activity

Implement the missing components

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
           index = get_index(command[1])
           points[index] += int(command[2])
        elif command_type == 'GET':
            # (B)
        elif command_type == 'EXIT':
            # (C)
        else:
            # (D)
```

```
players = []
points = []

def get_command():
    . . .
def get_index(player):
    . . .
def show_scores():
    . . .
```

Activity

Implement the missing components

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            index = get_index(command[1])
            points[index] += int(command[2])
        elif command_type == 'GET':
            index = get_index(command[1])
            print(command[1], 'has', points[index], 'points.')
        elif command_type == 'EXIT':
            # (C)
        else:
            # (D)
```

```
players = []
points = []

def get_command():
    . . .
def get_index(player):
    . . .
def show_scores():
    . . .
```

Implement the missing components

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            index = get_index(command[1])
            points[index] += int(command[2])
        elif command_type == 'GET':
            index = get_index(command[1])
            print(command[1], 'has', points[index], 'points.')
        elif command_type == 'EXIT':
            show_scores()
            break
        else:
            # (D)
```

```
players = []
points = []

def get_command():
    . . .
def get_index(player):
    . . .
def show_scores():
    . . .
```

Implement the missing components

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            index = get_index(command[1])
            points[index] += int(command[2])
        elif command_type == 'GET':
            index = get_index(command[1])
            print(command[1], 'has', points[index], 'points.')
        elif command_type == 'EXIT':
            show_scores()
            break
        else:
            print('?')
```

```
players = []
points = []

def get_command():
    . . .
def get_index(player):
    . . .
def show_scores():
    . . .
```

```
players = []
points = []

def get_command():
    user_input = input('Cmd: ')
    return user_input.split(' ')

def get_index(player):
    if player not in players:
        players.append(player)
        points.append(0)
    return players.index(player)

def show_scores():
    for i in range(len(players)):
        print(players[i] + ': ' + str(points[i]))
```

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            index = get_index(command[1])
            points[index] += int(command[2])
        elif command_type == 'GET':
            index = get_index(command[1])
            print(command[1], 'has', points[index], 'points.')
        elif command_type == 'EXIT':
            show_scores()
            break
        else:
            print('?')
```

Don't use global variables

```
players = []
points = []

def get_command():
    user_input = input('Cmd: ')
    return user_input.split(' ')

def get_index(player):
    if player not in players:
        players.append(player)
        points.append(0)
    return players.index(player)

def show_scores():
    for i in range(len(players)):
        print(players[i] + ': ' + str(points[i]))
```

```
def main():
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            index = get_index(command[1])
            points[index] += int(command[2])
        elif command_type == 'GET':
            index = get_index(command[1])
            print(command[1], 'has', points[index], 'points.')
        elif command_type == 'EXIT':
            show_scores()
            break
        else:
            print('?')
```

```
def get_command():
    user_input = input('Cmd: ')
    return user_input.split(' ')

def get_index(players, points, player):
    if player not in players:
        players.append(player)
        points.append(0)
    return players.index(player)


def show_scores(players, points):
    for i in range(len(players)):
        print(players[i] + ': ' + str(points[i]))
```

```
def main():
    players = []
    points = []
    while True:
        command = get_command()
        command_type = command[0]
        if command_type == 'ADD':
            index = get_index(players, points, command[1])
            points[index] += int(command[2])
        elif command_type == 'GET':
            index = get_index(players, points, command[1])
            print(command[1], 'has', points[index], 'points.')
        elif command_type == 'EXIT':
            show_scores(players, points)
            break
        else:
            print('?')
```