ENDG 233 – Programming with Data

Advanced Strings, Lists, Dictionaries

Week: Nov 1 – 7

Email Policy

- Email instructor if you have questions about:
  - Course material, assessments
- Email grading TA if you have questions about:
  - Assessment feedback
- Email course coordinator (endg233@ucalgary.ca) if you have questions about:
  - Video/video check technical issues, accommodations, zyLabs issues, missing grades in D2L (no zyLabs grades in D2L yet)
- If in doubt, email your instructor!

Schedule for Week 9

UNIVERSITY OF CALGARY

- Go through solutions for term test #1 and assignment #2
- Examples on advanced strings, lists, dictionaries

- Short Lab: zyLabs graded exercise based on last week's material

- Next week: No classes!

Review: List Comprehension

UNIVERSITY OF CALGARY

- Up to this point, we have been using for loops to modify lists
- Another way is to use list comprehension
- List comprehension iterates over a list, modifies each element, and returns a new list consisting of the modified elements
- Note that list comprehension is not an exact replacement of for loops, because list comprehension creates a new list object, whereas the typical for loop is able to modify an existing list

Review: List Comprehension Format

UNIVERSITY OF CALGARY

new_list = [expression **for** loop_variable_name **in** iterable]

- A list comprehension has three components:
  - An expression component to evaluate for each element in the iterable object
  - A loop variable component to bind to the current iteration element
  - An iterable object component to iterate over (list, string, tuple, enumerate, etc)

Review: List Comprehension Example

UNIVERSITY OF CALGARY

```
my_list = [10, 20, 30]
list_plus_5 = [ (i + 5) for i in my_list]

print('New list contains:', list_plus_5)
```

```
# for i in range(len(my_list)):
#     my_list[ i ] += 5
```

Output:

New list contains: [15, 25, 35]

Review: Conditional List Comprehension

UNIVERSITY OF CALGARY

- A list comprehension can be extended with an optional conditional clause that causes the statement to return a list with only certain elements

  new_list = [expression **for** name **in** iterable **if** condition]

- Using the above syntax will only add an element to the resulting list if the condition evaluates to True

Review: Another Example

UNIVERSITY OF CALGARY

```
numbers = [5, 52, 16, 7, 25]
# Return a list of only even numbers
even_numbers = [ i for i in numbers if (i % 2) == 0 ]
print('Even numbers only:', even_numbers)
```

Output:

Even numbers only: [52, 16]

Tutorial 10.1: Remove all non-alpha characters

UNIVERSITY OF CALGARY

- Write a program that removes all non-alpha characters from the given input
- Ex: If the input is:

  -Hello, 1 world$!
- the output is:

  Helloworld

- **Note**: for this program, use list comprehension

Tutorial 10.1: Remove all non-alpha characters

UNIVERSITY OF CALGARY

- **For loop solution**:

```
user_input = str(input())
user_input_no_spaces = ‘’
output_list = [ ]
for x in user_input:
        if x.isalpha():
                output_list.append(x)
user_input_no_spaces="".join(output_list)
print(user_input_no_spaces)
```

Ex:

```
new_list  = [i for i in numbers if (i % 2) == 0]
```

Tutorial 10.1: Remove all non-alpha characters

UNIVERSITY OF CALGARY

- **List comprehension** :

```
# input the string
# list comprehension to extract alphabets
# join the list to string
# print the string
```

Tutorial 10.1: Remove all non-alpha characters

UNIVERSITY OF CALGARY

- **List comprehension solution**:

```
user_input = str(input())  # input the string
# list comprehension to etract aplhabets
output_list = [ x for x in user_input if x.isalpha() ]
#join the list
user_input_no_spaces = "".join(output_list)
print(user_input_no_spaces)
```

Review: List Sorting

UNIVERSITY OF CALGARY

- Two ways to sort a list:

```
my_list = [5, 10, 3, 8, 4]

# Method 1: Sort original list
my_list.sort()

# Method 2: Sort list and store in new variable
new_list = sorted(my_list)
```

Tutorial 10.2: Filter and sort a list

UNIVERSITY OF CALGARY

- Write a program to input the string, which extracts a list of integers from input, and then outputs non-negative integers in ascending order (lowest to highest).
- Note: Program accepts the user input as string.
- If input is
  - 10 -7 4 39 -6 12 2
- Output:
  - 2 4 10 12 39

Tutorial 10.2: Filter and sort a list

UNIVERSITY OF CALGARY

```
# User inputs string with numbers
# Split into separate strings
# define empty list
# Convert strings to integers
# using a for loop. append the list
# Sort list of integers using sort function
```

```
my_list.sort()
new_list = sorted(my_list)
```

Tutorial 10.2: Filter and sort a list

UNIVERSITY OF CALGARY

Solution using for loop:

```
user_input = input()                    # User inputs string with numbers
tokens = user_input.split()             # Split into separate strings
# Convert strings to integers using a for loop
input_data = []
for token in tokens:
    if int(token) >= 0:
        input_data.append(int(token))
# Sort list of integers
input_data.sort()
for values in input_data:
    print(values, end=' ')
```

Tutorial 10.2: Filter and sort a list

UNIVERSITY OF CALGARY

```
# User inputs string with numbers
user_input = input()
tokens = user_input.split()          # Split into separate strings
# Convert strings to integers using list comprehension
input_data = [int(i) for i in tokens if int(i) >= 0]
# Sort list of integers
input_data.sort()
for values in input_data:
        print(values, end=' ')
```

Review: Dictionaries

Example:

```
prices = {}  # Create empty dictionary
prices['banana'] = 1.49  # Add new entry
print(prices)

prices['banana'] = 1.69  # Modify entry
print(prices)

del prices['banana']  # Remove entry
print(prices)
```

Output:

```
{'banana': 1.49}
{'banana': 1.69}
{}
```

Tutorial 10.3: Soccer team roster

UNIVERSITY OF CALGARY

- This program will store roster and rating information for a soccer team. Coaches rate players during tryouts to ensure a balanced team.
- **Step 1**: Prompt the user to input five pairs of numbers: A player's jersey number (0 - 99) and the player's rating (1 - 9)
  - Store the jersey numbers and the ratings in a dictionary
  - Output the dictionary's elements with the jersey numbers in ascending order (i.e., output the roster from smallest to largest jersey number)
  - ***Hint***: *Dictionary keys can be stored in a sorted list*

Tutorial 10.3: Soccer team roster

```
soccer_team = {}
menu_op = ‘’
for i in range(1, 6):
        jersey_num = int(input(f'Enter player {i}\'s jersey number:\n'))
        rating_num = int(input(f'Enter player {i}\'s rating:\n'))
        soccer_team[jersey_num] = rating_num
        print(‘’)
list_of_jersey_nums = sorted(list(soccer_team.keys()))
print('ROSTER’)
for i in list_of_jersey_nums:
        print(f'Jersey number: {i}, Rating: {soccer_team[i]}’)
```

Tutorial 10.3: Soccer team roster

- **Step 2**: Implement a menu of options for a user to modify the roster
  - The program initially outputs the menu, and repeats the menu after a user chooses an option
  - The program ends when the user chooses the option to Quit

```
MENU
a - Add player
d - Remove player
u - Update player rating
r - Output players above a rating
o - Output roster
q - Quit

Choose an option:
```

Tutorial 10.3: Soccer team roster

UNIVERSITY OF CALGARY

```
while menu_op != 'q’:
        # Print menu
        print('\nMENU')
        print('a - Add player')
        print('d - Remove player')
        print('u - Update player rating')
        print('r - Output players above a rating')
        print('o - Output roster')
        print('q - Quit\n')
        menu_op = input('Choose an option:\n')
```

Tutorial 10.3: Soccer team roster

UNIVERSITY OF CALGARY

- **Step 3**: Implement the "Output roster" menu option
- **Step 4**: Implement the "Add player" menu option
  - Prompt the user for a new player's jersey number and rating
  - Append the values to the dictionary
- **Step 5**: Implement the "Delete player" menu option
  - Prompt the user for a player's jersey number
  - Remove the player from the roster (delete the jersey number and rating)

Tutorial 10.3: Soccer team roster

UNIVERSITY OF CALGARY

```
if menu_op == 'a':                          # Add player
        jersey_num = int(input('Enter a new player\'s jersey number: \n'))
        rating_num = int(input('Enter the player\'s rating: \n'))
        soccer_team[jersey_num] = rating_num
elif menu_op == 'd':                        # Delete player
        jersey_num = int(input('Enter a jersey number: \n'))
        del soccer_team[jersey_num]
elif menu_op == 'u':                        # Update player
        jersey_num = int(input('Enter a jersey number: \n'))
        rating_num = int(input('Enter a new rating for player: \n'))
        soccer_team[jersey_num] = rating_num
```

Tutorial 10.3: Soccer team roster

UNIVERSITY OF CALGARY

- **Step 6**: Implement the "Update player rating" menu option
  - Prompt the user for a player's jersey number
  - Prompt again for a new rating for the player, and then change that player's rating
- **Step 7**: Implement the "Output players above a rating" menu option
  - Prompt the user for a rating
  - Print the jersey number and rating for all players with ratings above the entered value

Tutorial 10.3: Soccer team roster

UNIVERSITY OF CALGARY

```
elif menu_op == 'r':                    # Output players above a rating
        rating_num = int(input('Enter a rating: \n'))
        list_of_jersey_nums = sorted(list(soccer_team.keys()))
        print('ABOVE', rating_num)
        for i in list_of_jersey_nums:
                if soccer_team[i] > rating_num:
                        print(f'Jersey number: {i}, Rating: {soccer_team[i]}')
elif menu_op == 'o':                    # Output roster
        list_of_jersey_nums = sorted(list(soccer_team.keys()))
        print('ROSTER')
        for i in list_of_jersey_nums:
                print(f'Jersey number: {i}, Rating: {soccer_team[i]}')
```

Exercise 10.4: Word frequencies (dictionaries)

UNIVERSITY OF CALGARY

- Implement the build_dictionary() function to build a word frequency dictionary from a list of words.
- Ex: If the words list is: ["hey", "hi", "Mark", "hi", "mark"]
- the dictionary returned from calling build_dictionary(words) is:
  - {'hey': 1, 'hi': 2, 'Mark': 1, 'mark': 1}

Exercise 10.4: Word frequencies (dictionaries)

UNIVERSITY OF CALGARY

- The main code builds the word list from an input string, calls build_dictionary() to build the dictionary, and displays the dictionary sorted by key value.
- Ex: If the input is:
  - hey hi Mark hi mark
- The output is:
  - Mark: 1
  - hey: 1
  - hi: 2
  - mark: 1