1 Review

Lists and Functions

2 Functions

A function is a block of code that can be called at any time in a program.

- Functions allow programmers to avoid creating code that has lots of repetitions.
- Functions allow programmers to break the structure of a program into blocks (or modules), which means that different problems can be separated out and dealt with independently.

3 Lists

Lists are a collection of data items
This links to a number of code examples

3.1 What we should be able to do with lists:

- Create one and access items in the list using index notation.
- Find how many items are in the list
- Loop through the list and print out items (have more than one way of doing)
- Understand that there are basic **methods** associated with a list. These mean that we can add, remove, update elements in a list.

3.2 Activities

- Create a function that takes a list as an argument, prints out the list (horizontally), and returns the length of the list
- Create a program that asks for the user to enter 5 numbers, it then prints out the sum, average and product of the numbers (each part is a function).
- Create a program that asks for the user to enter 5 numbers, it then prints a menu asking giving users and option to get the sum, product, average of the numbers

4 Validating User Input

- Create a function to get input of a single letter from a user.
- Create a function to get an integer from a user
- Create a function to get a string between 2 and 5 letters
- Create a function to get an integer from a user with boundaries that might change

```
names = ["Alice","Belinda", "Chansing", "Debbie", "Eloise", "Floss"]
# view the list
# view the list with a counter
# find how many elements are in the list
# remove an elements from a list
# access an element in a list


# loop through list
for x in names:
    print(x, end="")
print()
# loop using a counter
for i in range(0, len(names)):
    print(i)
names.remove("Alice")
print(names)
names.append("Alice")
print(names)
names.sort()
print(names)
del names[3]
print(names)
print(names.pop(0))
print(names)
```

Listing 1: Lists

```
/usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/FruitBowlGitHub/lists.py
AliceBelindaChansingDebbieEloiseFloss
0
1
2
3
4
5
['Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss']
['Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss', 'Alice']
['Alice', 'Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss']
['Alice', 'Belinda', 'Chansing', 'Eloise', 'Floss']
Alice
['Belinda', 'Chansing', 'Eloise', 'Floss']
```

Listing 2: Lists

```
names = ["Alice","Belinda", "Chansing", "Debbie", "Eloise", "Floss"]

```

```
def print_list(L):
    my_string = ""
    for x in L:
        my_string =  my_string + x + " , "
    print(my_string)

def print_with_indexes(L):
    for i in range(0,len(L)):
        output = "{} : {}".format(i, L[i])
        print(output)


def remove_item(L):
    print_list(L)
    choice = input("Who would you like to remove?  ")
    if choice in L:
        L.remove(choice)
        print("{} has been removed".format(choice))
    else:
        print("You choice is not in the list")
    print_list(L)


def add_to_list(L):
    print_list(L)
    choice = input("Who would you like to add?  ")
    L.append(choice)
    print_list(L)

def remove_at_index(L):
    print_with_indexes(L)
    choice = int(input("Enter index number to remove:   "))
    if 0 <= choice < len(L):
        del L[choice]
        print_with_indexes(L)
    else:
        print("This is not a valid index number")

def sort_list(L):
    L.sort()
    print_with_indexes(L)



#print_list(names)
#remove_item(names)
#print_with_indexes(names)
```

```
#remove_at_index(names)
add_to_list(names)
sort_list(names)
```

Listing 3: Lists

```
def get_numbers():
    num_list = []
    counter = 0
    while counter < 3:
        num = int(input("Please enter a number "))
        num_list.append(num)
        counter += 1
    num_list.sort()
    print_list(num_list)
    return num_list


def print_list(l):
    my_string = ""
    counter = 0
    for x in l:
        if counter == len(l) - 1:
            my_string = my_string + str(x)
        else:
            my_string = my_string + str(x) + " , "

        counter += 1
    print(my_string)


def add_nums(l):
    _sum = 0
    for x in l:
        _sum += x
    return _sum


def product_nums(l):
    product = 1
    for x in l:
        product = product * x
    return product


def average_nums(l):
    _sum = add_nums(l)
    average = _sum / len(l)
```

```python
    return average


def print_output(m, v):
    print(30 * "-")
    print("Your {} is :  {}".format(m, v))
    print(30 * "-")


def run_menu():
    menu = True
    menu_list = ["Enter set", "Get average", "Get sum", "Get product"]
    set = []
    while menu is True:
        for i in range(0, len(menu_list)):
            print("{:<5}{:^10} {:20}".format(i, "--", menu_list[i]))
        user_choice = input("Please choose your option:  ")
        if user_choice == "q":
            return
        else:
            user_choice = int(user_choice)
        if user_choice != 0 and len(set) == 0:
            print("You have an empty set")
            continue
        if user_choice == 0:
            set = get_numbers()
        elif user_choice == 1:
            average = average_nums(set)
            print_output("average", average)
        elif user_choice == 2:
            sum = add_nums(set)
            print_output("sum", sum)
        elif user_choice == 3:
            product = product_nums(set)
            print_output("product", product)
        else:
            print("Unrecognised entry")


if __name__ == "__main__":
    run_menu()
```

Listing 4: Menu and Functions