Objectives for class 14

- Review IA4-IA7
- Review Quiz 3 – Quiz 5
- Final Exam Schedule
- Group Project Schedule

Individual Assignment 4 - 2

- Enter a letter grade A/a, B/b, C/c, D/d , and then displays its corresponding numeric value 90, 80, 70, 60.

Sample Run

```
Enter a letter grade: B
The numeric value for grade B is 80
```

- Letter **in** “Aa”
- Multiple way if statements

```
letter=input("Enter a letter grade: ")

if letter in 'Aa':
    print("The numeric value for grade A is 90")
elif letter in 'Bb':
    print("The numeric value for grade B is 80")
elif letter in 'Cc':
    print("The numeric value for grade C is 70")
elif letter in 'Dd':
    print("The numeric value for grade D is 60")
else :
    print(letter, "is an invalid grade")
```

Individual Assignment 4 - 3

- Display number of days in the month.

Sample Run

```
Enter a year: 2001
Enter a month: Jan
Jan 2001 has 31 days
```

A year is a **leap** year if **either** of the following conditions is satisfied:

- The year is multiple of 400.
  (year%400) ==0 (in python)
- The year is multiple of 4 and not multiple of 100.
  (year%4) ==0 and (year%100)!=0 (in python)

```
year = int(input("Enter a year: "))
month = input("Enter a month (first
three letters with the first letter in
uppercase): ")

if month == "Jan" or month == "Mar" or
month == "May" \
or month == "Jul" or month == "Aug" or
month == "Oct" or month == "Dec":
    print(month, year, "has 31 days")
elif month == "Apr" or month == "Jun"
or month == "Sep" or month == "Nov":
    print(month, year, "has 30 days")
elif month == "Feb":
    if (year % 4 == 0 and year % 100
!= 0) or (year % 400 == 0):
        print(month, year, "has 29
days")
    else:
        print(month, year, "has 28
days")
else:
    print(month, "is not a correct
month name ")
```

Individual Assignment 5 - 1

- Prompts the user to enter a string and displays its length and its first character.

Sample Run

```
Enter a string: Programming is fun
The length of string "Programming is fun" is 18
The last character of string Programming is fun is n
```

len(s)

s[-1]

```
s = input("Enter a string: "))

print("The length of string \"Programming is fun\" is ",len(s))

print("The last character of string Programming is fun is ",s[-1])
```

Individual Assignment 5 - 2

- Decide major and year
  - M: Mathematics
  - C: Computer Science
  - I: Information Technology

Sample Run

```
Enter two characters: M1
Mathematics
Freshman
```

More efficient with list created

```
s = input("Enter two characters: ")

ms=['M','C','I']
majors=['Mathematics','Computer
Sicence','Information Technology']
years=['Freshman', 'Sophomore', 'Junior',
'Senior']

year=int(s[1])
major=majors[ms.index(s[0])]


print(major)
print(years[year-1])
```

Individual Assignment 5 - 3

- An ISBN-9 to ISBN-10 converter (without function)

If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention

$$cheksum = (d_1 * 1 + d_2 * 2 + d_3 * 3 + d_4 * 4 + d_5 * 5 + d_6 * 6 + d_7 * 7 + d_8 * 8 + d_9 * 9) \% 11$$

Sample Run 1

```
Enter the first9 digits of an ISBN as a string: 3601267
Incorrect input. It must have exact 9 digits
```

Sample Run 2

```
Enter the first 9 digits of an ISBN as a string: 013601267
The ISBN-10 number is 0136012671
```

Sample Run 3

```
Enter the first 9 digits of an ISBN as a string: 013031997
The ISBN-10 number is 013031997X
```

```
isbn9=input('Enter the first 9 digits of an ISBN as a string:')

if len(isbn9)!=9:
    print('Incorrect input. It must have exact 9 digits')
else:
    sum=0
    for i in range(1,10):
        sum=sum+int(isbn9[i-1])*i
    sum=sum%11

    if sum == 10:
        d10='X’
    else:
        d10=str(sum)
    print("The ISBN-10 number is",isbn9+d10 )
```

Individual Assignment 6 - 3

- An ISBN-9 to ISBN-10 converter (with function)

```
def isbn9toisbn10(isbn9):
    sum=0
    for i in range(1,10):
        sum=sum+int(isbn9[i-1])*i
    sum=sum%11

    if sum == 10:
        d10='X’
    else:
        d10=str(sum)
    return isbn9+d10

isbn9=input('Enter the first 9 digits of an ISBN as a string:')

if len(isbn9)!=9:
    print('Incorrect input. It must have exact 9 digits')
else:
    print("The ISBN-10 number is",isbn9toisbn10(isbn9) )
```

Individual Assignment 6 - 1

Sample Run

```
Enter an integer, the input ends if user enters ” End ": 1
Enter an integer, the input ends if user enters " End ": 2
Enter an integer, the input ends if user enters " End ": -1
Enter an integer, the input ends if user enters " End ": 3
Enter an integer, the input ends if user enters " End ": 0
Enter an integer, the input ends if user enters " End ": End
The number of even numbers is 2
The average is 1.0
```

```
s = input("Enter an integer, the input ends if user enters \”End\"")
count=0
evencount=0
sum=0

while s!=‘End’:
    num=int(s)
    count=count+1
    if num%2==0:
        evencount=evencount+1
    sum=sum+num

    s = input("Enter an integer, the input ends if user enters \”End\"")

if count==0:
    print("No number entered")
else:
    print("The number of even numbers is",evencount)
    print("The average is",round(sum/count,2))
```

Individual Assignment 6 - 2

- Displays the number of vowels and consonants in the string.

Sample Run

```
Enter a string: Programming is fun
The number of consonants is 11
The number of vowels is 5
```

```
s=input("Enter a string:")
nvowels=0
ncons=0
for i in range(0,len(s)):
    ch=s[i]
    if ch.isalpha():
        if ch in 'AEIOU' or ch in 'aeiou’:
            nvowels=nvowels+1
        else:
            ncons=ncons+1

print("The number of consonants is ",ncons)
print("The number of vowels is ",nvowels)
```

Individual Assignment 7 - 1

- Display number of days in years

Sample Run

```
Enter a start year:2000
Enter an end year:2005
year    days
2000    366
2001    365
2002    365
2003    365
2004    366
2005    365
```

```
def isleapyear(year):
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

def caldays(year):
    if isleapyear(year):
        return 366
    else:
        return 365

def main():
    startyear = int(input("Enter a start year:"))
    endyear = int(input("Enter an end year:"))

    print("year",'\t',"days")
    for i in range(startyear, endyear+1):
        ndays=caldays(i)
        print(i,'\t',ndays)

main()
```

Individual Assignment 7 - 2

convert.py

```
def celsiusToFahrenheit(celsius):
    fahrenheit = (9 / 5) * celsius + 32
    return fahrenheit

def fahrenheitToCelsius(fahrenheit):
    celsius = (5 / 9) * (fahrenheit - 32)
    return celsius
```

Individual Assignment 7 - 2

test.py

```
import convert

print("1. Convert Celsius to Fahrenheit")
print("2. Convert Fahrenheit to Celsius")


choice=int(input("Enter your choice (1 or 2):"))

if choice == 1:
    celsius=int(input("Enter a temperature in Celsius: "))
    print("Fahrenheit is",int(convert.celsiusToFahrenheit(celsius)))
elif choice == 2:
    fahrenheit=int(input("Enter a temperature in Fahrenheit: "))
    print("Celsius is",int(convert. fahrenheitToCelsius(fahrenheit)))
```

Individual Assignment 8 - 1

- Count the number of given ratings from the list.

Sample Run 1

```
Please enter a rating: 5
3 ratings from our data
set are with value 5
```

```
ratings_list=[4,5,1,2,3,5,4,2,3,1,5,4,4,3]

rating=int(input("Please enter a rating:"))

count=ratings_list.count(rating)

print(count,"ratings from our data set are
with value", rating)
```

Individual Assignment 8 - 2

- Count the frequency of ratings(0,1,2,3,4,5)

Sample Run

```
Rating 0: 1
Rating 1: 2
Rating 2: 2
Rating 3: 3
Rating 4: 4
Rating 5: 3
```

```
ratings_list=[4,5,1,2,3,5,4,2,3,1,5,4,4,3]

counts=[0]*5

for i in range(0,len(ratings_list)):
    counts[ratings_list[i]-1]=counts[ratings_list[i]-1]+1

for i in range(0,len(counts)):
    print("Rating",i+1,":",counts[i])
```

Individual Assignment 8 - 3

- Displays the person(s) who gives the highest rating, and the person(s) who gives the lowest rating

```
ratings_list=[4,5,1,2,3,5,4,2,3,1,5,4,4,3]
names_list=["john", "alex", "anna", "max", "allen",
"bria", "david", "jane", "cathy", "jecissa",
"grace", "nick", "paul", "hans",]
```

Sample Run

```
Highest rating is 5, given by alex,bria,grace
Lowest rating is 1, given by anna,jecissa
```

```
ratings_list=[4,5,1,2,3,5,4,2,3,1,5,4,4,3]
names_list=["john", "alex", "anna", "max", "allen", "bria",
"david", "jane", "cathy", "jecissa", "grace", "nick", "paul",
"hans",]

max=max(ratings_list)
min=min(ratings_list)

maxnames=""
minnames=""

for i in range(0,len(ratings_list)):
    if ratings_list[i]==max:
        maxnames=maxnames+','+names_list[i]
    elif ratings_list[i]==min:
        minnames=minnames+','+names_list[i]

print("Highest rating is",max, "given by",maxnames[1:])
print("Lowest rating is",min, "given by",minnames[1:])
```

Quiz 3

- Check if **c** is a newline character.

c == ‘\n’

- Assume we have assigned values to a list of variable below.

```
s1 = "Welcome to Python"
s2 = "to"
x = 1
```

```
max(2, 2, x)          _______1_________
"come" not in s1      ____False____
```

Quiz 4

Will the following program terminate?

```
balance = 10

while True:
    if balance < 2:
        break
    balance = balance + 3
```

Answer should be NO

Quiz 4

- What is the output of the following code?

```
count = 1
while count < 2:
    count += 1
    print(count, end = ' ')
```

Answer is 2

Quiz 4

Suppose that s1 and s2 are two strings, given as follows:

```
s1 = "Welcome to Python"
s2 = "1234"
```

What are the results of the following expressions?

a. s1[0] _____W_____

b. s1[:3] _____Wel_____

c. s1[1:3]) _____el_____

d. len(s2) _____3_____

e. s2*2 _____12341234_____

f. s1.count('o') _____3_____

g. s1.isupper() _____False_____

h. s2.isdigit() _____True_____

Quiz 5

Given the following function, what will be displayed by the call **nPrint(n=4, message='a')**?

```
def nPrint(message, n):
    while n > 0:
        print(message, end = '')
    n -= 1
```

Answer should be infinite loop

Objectives covered in Final Exam

--- Chapter 4 ---

- 4.1 To solve mathematics problems by using the functions in the math module (§4.2)
- 4.2 To represent and process strings and characters (§4.3).
- 4.3 To encode characters using ASCII and Unicode (§4.3.1).
- 4.4 To use the ord function to obtain ar numerical code for a character and the chrfunction to convert a numerical code to a character (§4.3.2).
- 4.5 To represent special characters using the escape sequence (§4.3.3).
- 4.6 To test substrings using the in and not in operators (§4.3.8).
- 4.7 To compare strings (§4.3.9).
- 4.8 To use string functions min, max, and len (§4.3.10).

- 4.9 To obtain a character in a string using the index operator [] (§4.3.11).
- 4.10 To obtain a substring in a string using the slicing operator [start : end] (§4.3.12).
- 4.11 Use repetition operator * to duplicate strings (§4.3.6).
- 4.12 To introduce objects and methods (§4.5).
- 4.13 To introduce the methods in the str class (§4.6).
- 4.14 To program using characters and strings (§4.7.1).
- 4.15 To invoke the print function with the end argument (§4.3.4).

--- Chapter 5 ---

- 5.1 To write programs for executing statements repeatedly using a while loop (§5.2).
- 5.2 To control a loop with the user's confirmation and a sentinel value (§5.5).
- 5.3 To develop loops following the loop design strategy (§5.4).
- 5.4 To use for loops to implement counter-controlled loops (§5.6).
- 5.6 To implement program control with break. (§5.10).

--- Chapter 6 ---

- 6.1 To understand what is a function (§6.1)
- 6.2 To define functions with formal parameters (§6.2).
- 6.3 To distinguish the differences between the functions that return and do not return a value (§6.4).
- 6.4 To invoke functions with actual parameters (i.e., arguments) (§6.3).
- 6.5 To determine the scope of variables (§6.8)
- 6.6 To invoke a function using positional arguments or keyword arguments (§6.5)
- 6.7 To invoke functions defined from another program (§6.7)
- 6.8 To apply the concept of function in software development and design (§6.13)

--- Chapter 7 ---

- 7.1 To describe why lists are useful in programming (§7.1).
- 7.2 To learn how to create lists (§7.2).
- 7.3 To use the len, min, max, sum, and random.shuffle functions with a list (§7.2.2).
- 7.4 To access list elements by using indexed variables (§7.2.3).
- 7.5 To obtain a sublist from a larger list by using the slicing operator [start : end : step] (§7.2.4).
- 7.6 To use the + (concatenation), * (repetition), and in/not in operator
- 7.7 To traverse elements in a list using a for loop (§7.2.6).
- 7.8 To invoke a list's append, count, extend, index, insert, pop, remove, reverse, and sort methods (§7.2.9).
- 7.9 To split a string into a list using the str's split method (§7.2.10).
- 7.10 To develop and invoke functions that pass list arguments (§7.6).
- 7.11 To develop and invoke functions that return lists (§7.7).

--- Chapter 8 ---

- 8.1 To learn how a two-dimensional list can represent two-dimensional data (§8.1).
- 8.2 To access elements in a two-dimensional list by using row and column indexes (§8.2).
- 8.3 To program common operations for two-dimensional lists (displaying lists, summing all elements) (§8.2).

Final Exam : Schedule

- 7:30 pm to 9:30 pm, on Tuesday, Dec. 5th
- Remember to join your assigned WebEx meeting room on time

- 7:15 pm to 7:30 pm – Q&A – Zoom meeting room

Final Exam : Style of Questions

- Multiple Choices
- Fill in blanks
- Software development
  - System Analysis
  - System Design
  - Implementation

Final Exam : How to Prepare?

- Review slides from week 7 to week 14
- Review in-class exercises
- Review individual assignments 4 to 8
- Review quiz 3 to 5

Test Design Matrix

	Multiple Choices	Fill In blanks	Problem Solving	
For loop and range function	4	0	2	**6**
While loop	2	0	0	**2**
String concepts and string processing	6	5	3	**14**
Function creation and call	12	5	0	**17**
1D list	4	10	2	**16**
2D list	2	0		**2**
Software Development [System analysis, System Design, Implementation]	0	0	43	43
	30	**20**	**50**	**100** (Points)

Group Project

- Due on Monday midnight, Dec. 11th
- You can use Zoom or WebEx to record your presentation