Assignment 3

Total : 26pts

In [7]:
```
# create a tuple to hold your name and student number,
# unpack the tuple into separate variables and then
# print out the values

student = ("Yong Seung Rho", "W0447442")
print(f"Name: {student[0]}, StudentID: {student[1]}")
```

```
Name: Yong Seung Rho, StudentID: W0447442
```

Name: Jane Doe, StudentID: w123456

Part A - Small Bits (15pts)

For each item below, determine the appropriate Python code to generate the desired output.

In [55]:
```
# create a list of all assignment names in this course
# then loop through the list and print the names of all the
# assignments, except the current one.
list_assignment_names = ["Assignment 1", "Assignment 2", "Assignment 3", "Assignment 4", "Assignment 5"]

for name in list_assignment_names:
    if name != "Assignment 3":
        print(name)
```

```
Assignment 1
Assignment 2
Assignment 4
Assignment 5
```

In [56]:
```
# create a list of your courses this semester.
# make each item a tuple that includes both the course code and the course name
# e.g. 'PROG1700' and 'Logic and Programming I'
# then print out the first course name in the list and the last course code
list_tuple_courses = [("COMM1700", "Professional Practics for IT"),
                      ("PROG1700", "Logic and Programming"),
                      ("NETW1700", "Networking and Security"),
                      ("DBAS1007", "Data Fundamentals"),
                      ("OSYS1700", "Hardware and Operating Systems"),
                      ("DBAS2104", "Business Analyss Essentials")]

#the first course name in the list
print(f"The first course name: {list_tuple_courses[0][1]}")

#the last course code
print(f"The last course code: {list_tuple_courses[-1][0]}")
```

```
The first course name: Professional Practics for IT
The last course code: DBAS2104
```

In [57]:
```
# create a program that asks the user for a word
# then make every odd letter upper case and every even letter lower case
# print out the resulting word
while True:
    word = input("Enter a word ('e' to exit): ")
    if word == 'e':
        break

    if word.isalpha():
        list_word = list(word)
        for i, char in enumerate(word):
            if i % 2 == 0:
                list_word[i] = char.lower()
            else:
                list_word[i] = char.upper()
        converted_word= "".join(list_word)
        print(converted_word)
    else:
        print("Invalid word")
```

```
Enter a word ('e' to exit):  qwerty

qWeRtY

Enter a word ('e' to exit):  e
```

In [58]:
```
# create a dictionary that maps each letter of the alphabet to a different letter
# e.g. a = x, b = q, c = p, ...
# then use that dictionary to encrypt the following message
# the resulting encrypted message should be all capital letters and contain no
# punctuation or spaces.

message = 'This is the great and powerful Oz.'

# add your code here
dict_encrypted = {}
asc_a = ord('a')
asc_z = ord('z')

# make a dictionary
for i in range(0, 26):
    dict_encrypted.update({chr(asc_a + i): chr(asc_z - i)})
#print(dict_encrypted)

# encript the message
list_message = []

for char in message:
    if (char.isalpha()):
        if (char.isupper()):
            char = char.lower()
        if char in dict_encrypted:
            list_message.append(dict_encrypted.get(char).upper())

# make an encrypted message and print out it
str_encrypted_message = "".join(list_message)
print(str_encrypted_message)
```

```
GSRHRHGSVTIVZGZMWKLDVIUFOLA
```

In [59]:
```
# create a program that produces an acronym from a series of words
# e.g. the output should be 'NSCC'

text = "Nova Scotia Community College"

# add your code here
list_acronym = []

for line in text.split(' '):
    list_acronym.append(line[0])

str_acronym = "".join(list_acronym)
print(str_acronym)
```

```
NSCC
```

Part B - Big Bytes! (11pts)

The following are more challenging questions. Be patient when tackling these!

In [60]:
```
# create a program that converts a date in the format "mm/dd/yy"
# to a date in the format "dd-mmm-yyyy" using the provided string.
# e.g. 10/25/19 => 25-Oct-2019
# Note: you can assume all dates are >= 2000

# 3 pts

date = "10/25/19" # sample date

months = "JanFebMarAprMayJunJulAugSepOctNovDec"

# put your code here
date_split = date.split('/')
month = int(date_split[0])
day = date_split[1]
year = "20" + date_split[2]

list_months = []
nums = int(len(months) / 3)
for i in range(0, nums):
    start = i * 3
    end = start + 3
    list_months.append(months[start:end])

str_month = list_months[month - 1]
print(f"{date} => {day}-{str_month}-{year}")
```

```
10/25/19 => 25-Oct-2019
```

In [61]:
```
# create a program that generates a tuple containing the name and mark
# of the assignment with the lowest grade

# 3 pts

assignments = {
    'Assignment 1': 45,
    'Assignment 2': 65,
    'Assignment 3': 12,
    'Assignment 4': 78,
    'Assignment 5': 52
}

# put your code here
lowest_key = ""
lowest_value = 100
for key, value in assignments.items():
    if value < lowest_value:
        lowest_value = value
        lowest_key = key

tuple_lowest_grade = tuple((lowest_key, lowest_value))
print("tuple_lowest_grade =", tuple_lowest_grade)
```

```
tuple_lowest_grade = ('Assignment 3', 12)
```

In [62]:
```
# create a program to capitalize the first letter of each sentence.
# the output should look like this:

# This is a story about a man named jed.
# A poor mountaineer. Barely kept his family fed.
# Then one day he was shooting at some food,
# and out from the ground came a bubblin' crude.

# 5 pts

story = """
this is a story about a man named jed.
a poor mountaineer. barely kept his family fed.
then one day he was shooting at some food,
and out from the ground came a bubblin' crude.
"""

# put your code here
start_capital = True
CHAR_PERIOD = '.'
CHAR_SPACE = ' '

for line in story.splitlines():
    if not line:
        continue

    count = line.count(CHAR_PERIOD)
    if count > 1:
        i = 0
        for string in line.split(CHAR_PERIOD):
            if not string:
                continue

            if string.startswith(CHAR_SPACE):
                string = string.lstrip()
                string = CHAR_SPACE + string.capitalize()
                if i < count:
                    string = string + CHAR_PERIOD
            else:
                string = string.capitalize() + CHAR_PERIOD
            print(string, end = '')
            i = i + 1
```

```
        print()
    else:
        if start_capital:
            print(line.capitalize())
        else:
            print(line)

    start_capital = False
    if line.endswith(CHAR_PERIOD):
        start_capital = True
```

```
This is a story about a man named jed.
A poor mountaineer. Barely kept his family fed.
Then one day he was shooting at some food,
and out from the ground came a bubblin' crude.
```

In [ ]: