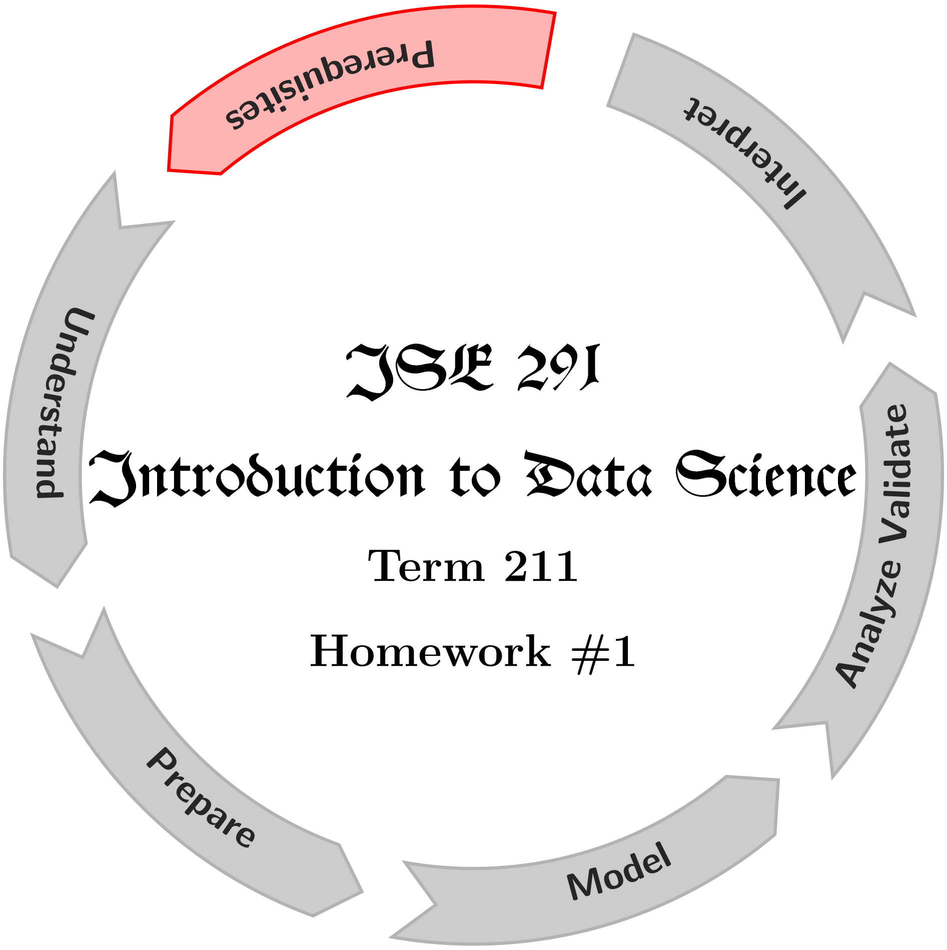
[The HW must be submitted as one .ipynb file. Write names & IDs of all the group members.]



ISE 291

Introduction to Data Science

Term 211

Homework #1

Homework Guidelines

To receive full credit, you should make sure you adhere to the following guidelines. For any questions/-comments contact your section instructor.

Homework Presentation & Submission:

- Every sub-problem (part) should be answered on a DIFFERENT CELL.
- EVERY CELL should have problem and part number clearly written in the first line.
- You should submit the solutions for the FIRST TWO problems only.
- All cells of your homework should be in CHRONOLOGICAL order. One cell per sub-problem.
- Submit entire HW as ONE single .ipynb document.
- ONE HW per group should be submitted.
- Your NAMEs, IDs, and the homework number should be clearly indicated in the FIRST CELL of the notebook.

Problem # A 50 marks

Answer all the following questions:

A-1: Let $l = [-1, -2, \ldots, -10]$ be an existing list. Construct a new list from l by dividing each even number element in l by 2. Print the new list. **Do not use NUMPY library.**

A-2: Create a new list containing the following elements using list comprehension: $[1, -1, 2, -2, 3, -3, ..., 9, -9]$

A-3: Consider the following coded string. Create a list of all contiguous (connected without space) letters in the order of their appearance.

```
"Vjg dguv rtqitcou ctg ytkvvgp uq vjcv eqorwvkpi ocejkpgu ecp rgthqto vjgo
swkemn{0 Cnuq. vjg dguv rtqitcou ctg ytkvvgp uq vjcv jwocp dgkpiu ecp wpfgtuvcpf
vjgo engctn{0 C iqqf guuc{kuv cpf c iqqf rtqitcoogt jcxg c nqv kp eqooqp0"
```

A-4: In the list obtained after executing task(s) from Part(*A*-3), remove newline characters (if any).

A-5: For the list obtained after executing task(s) from Part(*A*-4), for each word in the list do the following:

- Breakdown the word into list of letters.
- Convert each of the above letters into integer, using **ord()** function.
- Update the integer values by -2 for all the above integers.
- Convert each updated integer value into letter, using **chr()** function.
- Join the updated letters in the order to create new word.

A-6: Join the list of new words obtained after executing task(s) from Part(*A*-5), into one string of words. Print the string.

A-7: Create a python program, that does the following:

- Ask the user to enter a number from the following choices: $10, 20, 30, 40, 50, 60, 70, 80, 90, 100$
- Returns the roman numerals and the word form for the entered number.

Hint(*A*-5): ‘a’ is chr(97), and ord(‘a’) is 97.
Hint(*A*-7): Use the following list for roman numerals correspondingly to the above numbers:
“X”, “XX”, “XXX”, “XL”, “L”, “LX”, “LXX”, “$LXXX$”, “XC”, “C”
☞ *Note: Solve all the above questions using Python only. Do NOT use* ***Numpy*** *or any other libraries.*

Problem #B 50 marks

Answer all the following questions:

B-1: Create a random integer numpy nd-array of size 8x9, where the integer values range from 0 to 9. Set the random generator seed to be 211. Call the nd-array as **S**.

B-2: Print the dimensions, shape and size of **S**.

B-3: Print squares of all the elements of the last but one row (*index -2*) of **S**.

B-4: Print $\frac{1}{10}^{th}$ of all the elements of the second column from the end (*index -2*)of **S**.

B-5: Print alternate rows starting from row indexed 1, and all the columns of **S**.

B-6: Print all the rows, and alternate columns starting from column indexed 2 of **S**.

B-7: Print **S** with all rows reversed.

B-8: Print **S** with all columns reversed.

B-9: Print a slice of **S** containing rows indexed [2,4,5] in order, and columns indexed [3,6,1] in order.

B-10: Create a copy of **S**, say **P**. Subtract -9 from all the middle column elements of **P**. Insert a new row of all ones in the middle **P**, such that the new dimesions of **P** become 9x9. Print both **S** and **P**.

☞ *Note: Solve all the above questions in Python using **Numpy** library.*

Problem #C (Practice only. No submission required.)

Consider the following python methods, available in naive python, or numpy library:

C-1: `len()`

C-2: `str.replace()`

C-3: `str.join()`

C-4: `np.random.randint()`

C-5: `np.random.rand()`

C-6: `np.sort()`

C-7: `np.argsort()`

C-8: `array.copy()`

C-9: `array.reshape()`

C-10: `np.argmin()`

Answer the following questions for each of the above methods:

- State the purpose/usage of the method.
- List all the argument of the method.
- Classify the arguments as positional or keyword arguments.
- Write the default values for each of the keyword arguments.

☞ *Note: You must use **help()** function from python to answer all the above questions.*

Problem #D (Practice only. No submission required.)

Answer the following questions:

D-1: Create a random integer numpy nd-array of size 5x8, where the integer values range from 10 to 90. Set the random generator seed to be 211. Call the nd-array as **M**.

D-2: Subtract 5 from all the elements of **M**.

D-3: Subtract 5 from all the third row elements of **M**.

D-4: Subtract 5 from all the fifth column elements of **M**.

D-5: Divide all the row 2 and 3 elements of **M** with 10.

D-6: Create a copy of **M**, as **M1**. In **M1**, update each element as follows: if the element's vlaue is < 50, then update it to 0, otherwise update the value to 100. Print both the arrays (**M** and **M1**).

D-7: Split **M** into two nd-arrays of size 3x8 and 2x8.

D-8: Create a list of 8 random integers, whose value range from 1 to 5. Append (attach) the list as a sixth row to **M**.

D-9: Create a list of 5 random integers, whose value range from 1 to 5. Append (attach) the list as a ninth column to **M**.

D-10: Execute the following cells, and explain your observations:

In [1]:

```
list1 =[1,2,3,4]
print((lambda x: x * 3 )(list1))
```

In [2]:

```
array1 =np.array([1,2,3,4])
print((lambda x: x * 3 )(array1))
```

☞ *Note: Solve all the above questions using Python & **Numpy** library.*

Problem #E (Practice only. No submission required.)

Explain the following **Python** codes. In the following codes, **np** stands for **numpy** library.:

Code-1: ______________________________

In [1]:
```
L=["".join(["A",str(i+1)]) for i in range(10)]
```

Code-2: ______________________________

In [2]:
```
print("Head" if np.random.rand() < 0.5 else "Tail")
```

Code-3: ______________________________

In [3]:
```
x,y = 5,30
y,x = x+1,y+2
print(x,y)
```

Code-4: ______________________________

In [4]:
```
S,L=["A","B","C","D","E","F"],[1, 2, 3, 4, 5, 6]
for i,z in enumerate(zip(S,L)):
   print(i,z)
```

Code-5: ______________________________

In [5]:
```
print(L[-4:],"\n", L[::-1])
```

Code-6: ______________________________

In [6]:
```
K=L
K[1:3]=0,0,0
print(K,"\n", L)
```

Code-7: ______________________________

In [7]:
```
h = {a:np.random.randint(10,100) for a in range(1,11)}
print(h)
```

Code-8: ______________________________

In [8]:
```
v=[np.random.randint(10,100) for i in range(12)]
m=sum(v)/len(v)
s=list(map(lambda e: (e-m)**2, v))
print(sum(s)/(len(v)-1))
```