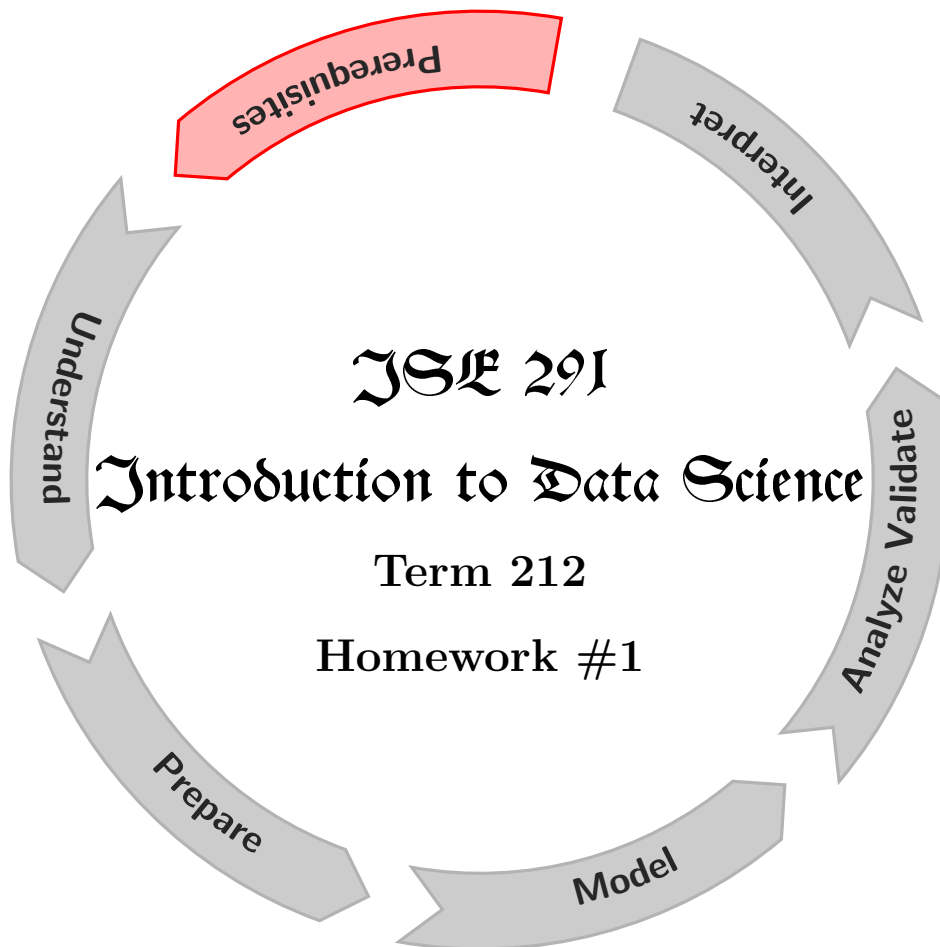


[The HW must be submitted as one .ipynb file. Write name & ID in the provided template.]



#### 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:

- You should submit the solutions for the **FIRST TWO** problems only.
- Every sub-problem (part) should be answered on a DIFFERENT CELL as given in the template.
- EVERY CELL should have problem and part number clearly written in the first line.
- All cells of your homework should be in CHRONOLOGICAL order. One cell per sub-problem.
- Any text should be written as comment in the code cell. Do NOT modify code cell into markdown cell.
- Submit entire HW as ONE single .ipynb document.
- **Do NOT add/delete** any cell in the given template.

## Problem # A

50 marks

Answer all the following questions:

☞ *Note: Solve all the following questions using Python only. Do NOT use **Numpy** or any other libraries for this problem.*

A-1: State the purpose and usage of `len()` function. Let `l1` be a list of 5 elements, and `l2` be a list containing 10 lists, where each inner list contain 3 elements. What will be the result of `print(len(l1))` and `print(len(l2))`.

A-2: State the purpose and usage of `enumerate()` function. Let `l1` be a list of 5 elements. What will be the result of `print(list(enumerate(l1)))` and `print(dict(enumerate(l1)))`.

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

A-4: State the purpose and usage of `str.replace()`, `str.join()` and `str.split()` functions. Which of the above functions transforms a string into a list of strings? Which of the above functions transforms a list of strings into one string?

A-5: Consider the following coded string. Create a list of words (all contiguous characters) in the order of their appearance.

```
"Rfc 'cqr npmep_kq _pc upgrrcl qm rf_r amknsrgle k_afglcq a_l ncpdmpk rfck
osgaijw?jqm* rfc 'cqr npmep_kq _pc upgrrcl qm rf_r fsk_l 'cgleq a_l slbcpqr_lb
rfck ajc_pjw, ? emmb cq-qwgqr _lb _ emmb npmep_kkcp f_tc _ jmr gl amkml,"
```

A-6: In the list obtained after executing task(s) from Part(A-5), remove newline characters (if any).

A-7: For the list obtained after executing task(s) from Part(A-6), for each word in the list do the following:

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

A-8: Join the list of new words obtained after executing task(s) from Part(A-7), into one string of words. Print the string.

A-9: What is dictionary type in python. What are the possible valid data types for the keys. Can a dictionary have mixed data types for the keys (for example, one key is string, another key is integer, etc)? Create a dictionary that contains `[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]` as keys, and corresponding roman numerals `["X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC", "C"]` as values.

A-10: Do the following:

- Create a lambda function, that takes an integer, and returns the following:
  - if  $x \geq 0$ , then the function returns  $x^2 + 2x + 1$ .
  - if  $x < 0$ , then the function returns  $x^2 - 2x + 1$ .
- Execute the above function for list `[1, -1, 2, -2, 3, -3, ..., 9, -9]` without using FOR or any loop statement.

Hint(A-7): 'a' is `chr(97)`, and `ord('a')` is 97.

**Problem #B****50 marks**

Answer all the following questions:

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

*B-1:* How to convert an existing list into a numpy array? How to convert an existing numpy array into a list?

*B-2:* Create a random integer numpy nd-array of size 10x7, where the integer values range from 1 to 20 (20 inclusive). Set the random generator seed to be 212. Call the nd-array as **S**. . Print the dimensions, shape and size of **S**.

*B-3:* What is the difference between **S[:,0]** and **S[:,[0]]**?

*B-4:* What is the difference between **S[0,:]** and **S[0]**?

*B-5:* Print squares of all the elements of the last but one column (*index= -2*) of **S** in the ascending order.

*B-6:* Print alternate rows starting from row indexed 1, and all the columns of **S**.

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

*B-8:* Print a slice of **S** containing rows indexed [2,4,5] in order, and columns indexed [3,6,1] in order.

*B-9:* Print all the elements of **S** that are greater than or equal to 10.

*B-10:* Create a copy of **S**, say **P**. Subtract  $-5$  from all the middle column elements of **P**. Insert a new row of all ones in the middle **P**, such that the new dimensions of **P** become 11x7. Print both **S** and **P**.

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

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

*C-1:*     `range()`  
*C-2:*     `str.capitalize()`  
*C-3:*     `str.strip()`  
*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()`  
*C-11:*    `np.hstack()`  
*C-12:*    `np.vstack()`  
*C-13:*    `np.split()`

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

```
In [2]: 1 array1 =np.array([1,2,3,4])
        2 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]: 1 L=["".join(["A",str(i+1)]) for i in range(10)]
```

Code-2: \_\_\_\_\_

```
In [2]: 1 print("Head" if np.random.rand() < 0.5 else "Tail")
```

Code-3: \_\_\_\_\_

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

Code-4: \_\_\_\_\_

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

Code-5: \_\_\_\_\_

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

Code-6: \_\_\_\_\_

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

Code-7: \_\_\_\_\_

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

Code-8: \_\_\_\_\_

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