

M.TECH - BIG DATA ANALYTICS

Lab Manual

**DA2101 – Python programming for Data
Analytics**

Index

S.No	Experiment Name	Marks	Signature
1	Language Basics – Flow Control, Functions, Sequences		
	Language Basics – Sequence contd, Dictionaries		
	Language Basics - Classes		
2	Numpy and Pandas Basics, DataFrame Merge		
	Data Transformation		
3	Group By Mechanics		
	Time Series		
4	Web Scraping and Browser Emulation		
5	Graphs and Plots - Bar Histogram, Logarithmic, Polar, Stem		
	Graphs and Plots - Specgram, Quiver, Subplots, Patches		

DA2101– Python Programming for Data Analytics

Series 1 – Operators and Expressions

1a) Loss or Profit

Write a program to calculate the loss or profit (i.e., difference on price) in a book purchase at outlet or online. The program gets the price of the book you bought at outlet as input. Then it asks for the average price of the book, if bought online. The program computes the difference between these prices and displays the output.

Sample Output:

```
Enter the price of the book when bought at outlet/store:678
```

```
Enter average price of the book if bought online:670
```

```
Difference between outlet and online price = 8
```

1b) VAT Calculator

Write a python program for VAT Calculation. The program gets the net payable amount as input from the user. Then the program asks for VAT percentage from the user. The program computes the net amount and VAT amount and displays the result to the user.

Sample Output:

```
Enter the gross amount of purchase : 500
```

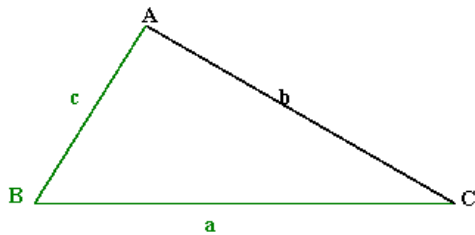
```
Enter the VAT percentage : 10
```

```
VAT amount is: 50.0
```

```
The computed net amount is: 550.0
```

1c) Side of a triangle

Write a python program to compute the third side of a triangle when two sides and an angle are given.



. We are given a, c, and $\angle B$.

$$b^2 = a^2 + c^2 - 2ac \cos(\angle B)$$

$$b = \sqrt{a^2 + c^2 - 2ac \cos(\angle B)}$$

The program gets side1, side2 and angle 1 as input from the user and computes the side 3, using the above formula.

Hint : import math module. Use functionalities like sqrt, cos from the module. Cos function expects the parameter to be in radians. Convert the degree got from user to radians using the function, radians(angle)

Sample Output:

```
Enter side1: 8
```

```
Enter side2: 10
```

```
Enter angle(in degrees): 60
```

```
The third side is: 9.16515138991168
```

Series 2 – Flow Controls & Functions

2a) Zeller's algorithm (Practice Program)

1. Write a program to compute Zeller's algorithm, which is used to tell the day of the week, provided a date is given.

Ask the user for the month as a number between 1 – 12 where March is 1 and February is 12.

If born in Jan or Feb, enter previous year.

Zeller's algorithm computation is defined as follows:

Let A, B, C, and D denote integer variables that have the following values:

A = the month of the year, with March having the value 1, April the value 2, December the value 10, and January and February being counted as months 11 and 12 of the preceding year (in which case, subtract 1 from C)

B = the day of the month (1, 2, 3, ..., 30, 31)

C = the year of the century (e.g. C = 89 for the year 1989)

D = the century (e.g. D = 19 for the year 1989)

Note: if the month is January or February, then the preceding year is used for computation. This is because there was a period in history when March 1st, not January 1st, was the beginning of the year.

Let W, X, Y, Z, R also denote integer variables. Compute their values in the following order using integer arithmetic:

$$W = (13 * A - 1) / 5$$

$$X = C / 4$$

$$Y = D / 4$$

$$Z = W + X + Y + B + C - 2 * D$$

R = the remainder when Z is divided by 7

Hint: Use integer division “//” for division

Now the value of “R” is used to find the day of the week.

If R is zero, then it is a Sunday. If R is one, then it is Monday and so on. When R = 6, the day is Saturday.

Sample Output:

```
                Zellers algorithm
Returns the day of the week for any Julian or Gregorian calendar date..

Enter month:    10
Enter day:      5
Enter year:     07
Enter century:  20
Its a wednesday
```

2b) Luhn's algorithm (Graded Program)

Write a program to verify valid credit card number. A valid card number passes a digit-sum test known as the Luhn checksum algorithm. Luhn's algorithm states that if you sum the digits of the number in a certain way, the total sum must be a multiple of 10 for a valid number. Systems that accept credit cards perform a Luhn test before contacting the credit card company for final verification.

The algorithm for summing the digits is the following. Consider each digit of the credit card to have a zero-based index: the first (starting from right) is at index 0, and the last is at index 15. Start from the rightmost digit and process each digit one at a time. For digits at even-numbered indexes (the 14th digit, 12th digit, etc.), simply

add that digit to the cumulative sum. For digits at odd-numbered indexes (the 15th, 13th, etc), double the digit's value, then if that doubled value is less than 10, add it to the sum. If the doubled number is 10 or greater, add each of its digits separately into the sum.

The following pseudocode describes the Luhn algorithm to sum the digits:

4408041254369873 is an example credit card number that passes the Luhn algorithm. The following figure shows the algorithm summing the latter number in detail. Notice how digits at even indexes are doubled and potentially split into two digits if they exceed 10 when doubled. For example, the number 7 at index 8 which is doubled to 14 which split to make 1+4.

An example checksum using the Luhn algorithm.

CC # 4408 0412 5436 9873

	4	4	0	8	0	4	1	2	7	4	3	6	9	8	5	3
Scale	*2		*2		*2		*2		*2		*2		*2		*2	

	8	4	0	8	0	4	2	2	14	4	6	6	18	8	10	3

Sum = 8 + 4 + 0 + 8 + 0 + 4 + 2 + 2 + 1+4 + 4 + 6 + 6 + 1+8 + 8 + 1+0 + 3
= 70

70 is divisible by 10, therefore this card number is valid.

Write a program where the user can type in a credit card number and receive a message stating whether the number was valid. The program should have a function `validate_number()` that takes the credit card number as argument and prints the message "Valid Credit Card Number" or "Invalid Credit Card Number" accordingly. The program should print an error message and exit if the length of the credit card number is not equal to 16.

Sample Output:

Enter the credit card number: 4408041254369873

Valid Credit Card number

Series 3 – Sequences (Strings & Tuples)

3a) Strings – Watson-Crick Complemented Palindrome Check

Write a program for checking if the given DNA sequence is Watson Crick complemented Palindrome or not.

Steps:

1. Get the string from the user.
2. Write a function `palindrome()` that returns the reversed sting of the original string
3. Write a function `watson_crick_complement()` that returns the complemented string. The logic for this is as follows:
 - a. A DNA Sequence, looks like “AGTCGTCA....”.
 - b. Components A and T are complement, that is if you find a A replace with its complement T and vice versa.
 - c. Similarly C and G are complements
 - d. The final string constructed after traversing the whole string and replacing the complements (A-T,C-G) is the Watson-Crick Complemented String
 - e. The function returns this string
4. If the reversed string and complemented strings are equal, then print that “The given string is a Watson Crick Complemented Palindrome Sequence”
5. If the strings are not equal print “The given string is not a Watson Crick Complemented Palindrome Sequence”

Sample Output:

```
>>>

Enter the DNA sequence to be processed : ACGT

The given string is a Watson Crick Complemented Palindrome Sequence
>>> ===== RESTART =====
>>>

Enter the DNA sequence to be processed : AGTCCTTGGA

The given string is not a Watson Crick Complemented Palindrome Sequence
>>>
```

3b) Tuples – Sort tuples

Get two tuples as input from the user. Each tuple entry ends when the user types “quit”

Combine the two tuples to a single tuple and print the elements of both the tuples sorted together in reversed order. Assume that both the tuples have only numbers.

Sample Output:

Start Entering Inputs:

5

3

7

quit

6

4

5

quit

The input tuples are : (5, 3, 7) (6, 4, 5)

The comined reverse sorted output tuple is:

(7, 6, 5, 5, 4, 3)

>>>

Series 4 - Sequences (Lists & Dictionaries)

5a)List - Second smallest element

Get the list of elements from the user till the user types “end”. Now find the second smallest element the user has entered. Assume we enter only numbers or “end” to this program.

Sample Output:

```
>>> ===== RESTART =====
>>>

Start entering inputs:
45
108
23
12
9
570
78
6502
end

Input list is:  [45, 108, 23, 12, 9, 570, 78, 6502]
[9, 12, 23, 45, 78, 108, 570, 6502]
The second smallest number in the input list is:  12
>>> |
```

5b) Lists – Find the longest string in the list of strings

Get the list of elements from the user till the user types “end”. Now find the longest string the user has entered. Assume we enter only strings or “end” to this program.

Sample Output:

```
Start entering inputs:
Hello
World
Name
Q
Run
end

Input list is:  ['Hello', 'World', 'Name', 'Q', 'Run']
The longest string in the input list is:  Hello
>>>
```

5c)Dictionary - Ceaser Cipher encryption

Create a dictionary that has the input characters, normal alphabets (as keys) and the corresponding encoded characters (we assume shift right by three positions for Ceaser Cipher) (as values). Your dictionary will only have lower case alphabets as keys (a – z). So these characters in the Input string will be converted to cipher. Replace any other characters (that are not in the dictionary’s key list) with a “?”

Get the input string from the user and output the encrypted string as output to the user.

Sample entries in dictionary

```
‘a’ -> ‘d’
‘b’ -> ‘e’
‘c’ -> ‘f’
```


and so on....

Sample Output:

```
>>>
```

```
Enter the plain text:
```

```
This is Python class.
```

```
The Cipher text using 3 char right shift scheme is : ?klv?lv??bwkrq?fodvv?
```

```
>>> |
```

Series 5 – Classes

6a) Calculator Class

Define a class “Calc”.

Calc
+number1 +number2
getNumbers() add() sub() mul() div()

The method “getNumbers()” get the values of number1 and number2 from the user and the operation from the user.

The methods add,sub,mul and div performs the corresponding arithmetic operations on the numbers and prints the result

Sample Output:

```
Enter first number: 45
```

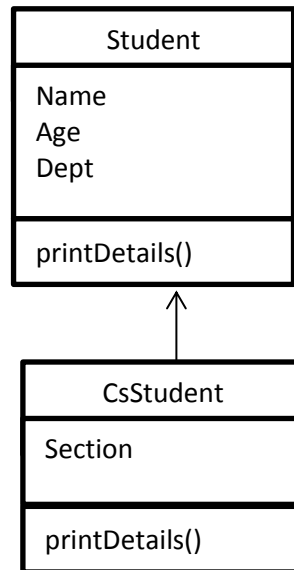
```
Enter second number: 8
```

```
Enter operation (add/sub/mul/div):div
```

```
The result is : 5
```

6b)OOP – Inheritance

Create a Student class and the child class CsStudent as follows,



- The child class has a variable called `studCount` that maintains the number of instances created.
- The child class has one additional property called `section`
- The call to parent's `printDetails` prints all the properties of parent
- The child class' `printDetails` method first calls the parent's one and then prints the `section` attribute alone.

Sample Output:

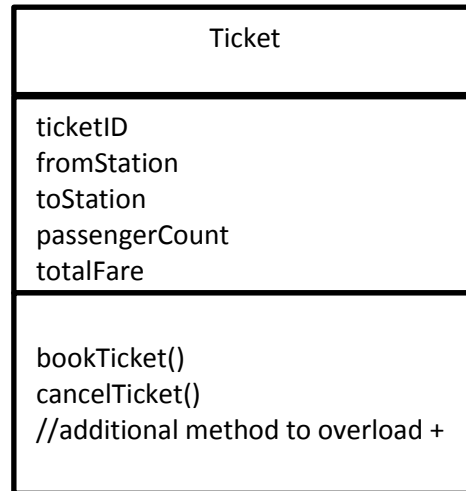
```
>>>
The Student details are:
  Felicia 22 IT
C
The Student details are:
  Sally 21 IT
E

TotalStudent Count:  2
```

6c) OOP – Operator Overloading

Write a program to realize the below class.

Apart from the instance variables, given in the class diagram, include one additional class level attribute called “ticketCount”.



- The method “bookTicket” increments the class variable ticketCount by 1 and prints the current ticketCount.
- The method “cancelTicket” decrements the ticketCount by 1 and prints the current ticketCount.
- Also include another method to overload the “+” operator for the “Ticket” class. The “+” operator when applied, should add the total fare of the two instances and print the sum
For example t1+ t2(where t1 and t2 are two instances of Ticket class) should print the sum of total fares of the two instances.

Sample Output:

```
Current Ticket Count:  1
Current Ticket Count:  2
Ticket Count 2
Total Fare
1900
```

Lab Series II

Data Wrangling

Given data set explains the personal expenditure in 5 different states of India for five years (2000-2004). Each tab has the data for a state. The expenditure is recorded in thousands of indian rupee.

Perform the following operation on the data set using pandas.

1. Read the data and create data frames for each state
2. Merge these data frames into a single one with hierarchical indexing, columns represents years, first level row represents state and the next the category of expenditure.
3. Add a new column to the merged data set called "Region".
4. Replace all -999 and -1000 with NaN
5. Remove any rows that have expenditure in any one column greater than or equal to 20.

Lab Series 3

Group By Mechanics

Create two data frames from the csv files

df1 from "Regions.csv" – Gives the state region mapping and

df2 from "stat218.csv" – Gives the statewise irrigation source details

Perform the following operations

- Merge the two dataframes. The resultant dataframe should have all columns from df1 along with the "Region" column from df2, with inner join based on State names.
Hint: Remove unnecessary or redundant columns after merge by using "del" command
- Group the resultant dataframe by Region to display the mean of "CULTURABLE COMMAND AREA" column
- Use the split-apply-combine principle to sort the dataframe by the column Region.

Time Series:

Create a time series with date range index of 100 days from "2011-01-01" with the frequency of every week thursday. Values could be random numbers.

Perform the following operations on the series

- Resample the series on the frequency of Business Month end
- Truncate the series so that it has entries only between 2011-10-01 and 2012-05-19
- Shift the index by two days.

DA2101 Python Programming for Data Analytics

Lab Series 4 – Web Scraping

Question 1:

Connect to the website “<http://www.srmuniv.ac.in/academics/engineering>” and save its contents to a local html page. Construct the below data frame from the page. Use urllib.

<code>text/javascript</code>	<no of times this tag has occurred>
<code>text/css</code>	<no of times this tag has occurred>
<code>col-md-3 univ-logo</code>	<no of times this text has occurred>

Question 2:

Connect to the website “[srmuniv.ac.in](http://www.srmuniv.ac.in)” and display the names of all forms. Also display all the hidden fields in each form. Use mechanize.

Question 3:

Connect to the website “[srmuniv.ac.in](http://www.srmuniv.ac.in)”. Type the content “Big Data” in the search text box, submit the form, store the resultant page in a local html file. Use urllib.

Question 4:

Connect to the website “[srmuniv.ac.in](http://www.srmuniv.ac.in)”. Type the content “Big Data” in the search text box, submit the form, store the resultant page in a local html file. Use mechanize.

Question 5:

Connect to the website “[srmuniv.ac.in](http://www.srmuniv.ac.in)” and fetch all the elements that has the below css classes. Print all the contents of each of these tags. Use BeautifulSoup.

- col-md-12
- container
- content

DA2101 Python Programming for Data Analytics

Lab Series 5 – Graphs and Plots

Question 1:

Demonstrate pie chart construction for the below data frame. All the text and axes annotations should be in place.

State	No. of maintained rural water sources - Primary
Tamil Nadu	78900
Kerala	90823
Andhra Pradesh	57568
Orissa	89723
Jharkhand	56789
Nagaland	4536

Question 2:

Construct a bar chart for the below data. Also include all text (in bars), axes annotations in your graph.

State	No. of irrigation wells rural- Primary
Tamil Nadu	78900
Kerala	90823
Andhra Pradesh	57568
Orissa	89723
Jharkhand	56789
Nagaland	4536

Question 3:

Divide the drawing area into 4 subplots. Plot the cos values of the below four ranges in four subplots using stem plots. One subplot for each range.

Range	Step Value
-------	------------

0 to pi	0.1
0 to 2pi	0.01
0 to pi	0.2
0 to 3pi	0.25

Question 4:

Draw a histogram. Generate a random 10x5 matrix. Divide it into 10 cells. Annotate text and axes. Stack the histogram. Use 5 different colors.

Question 5:

Draw two polar plots for $\sin(\theta)$ and $-\sin(\theta)$. Annotate and provide grid for both radius and angle. Assume θ .