

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

**Duration:** 44 classroom hours + 36 lab hours

**Objective:** To introduce the student to Python programming & R programming concepts.

**Prerequisites:** Knowledge of programming in any language like C, C++ and some basic statistical knowledge.

**Evaluation method:** Theory exam– 40% weightage  
Lab exam – 40% weightage  
Internal exam– 20% weightage

**List of Books / Other training material**

**Text Book:**

1. Python for Everybody Exploring Data using Python, Charles R. Severance, Shroff Publishers & Distributors, 1st Edition.

**Reference Book:**

1. Introduction to Computer Science using Python, Charles/ Wiley
2. Python Power!: The Comprehensive Guide
3. Python Crash Course: A Hands-on, Project-Based Introduction to Programming
4. Beginning Programming with Python For Dummies Learning Python by: Fabrizio Romano
5. Python Projects by Laura Cassell , Alan Gauld / Wiley
6. Python Cookbook by David B. Brain K. Jones / Shroff / O'reilly Publisher
7. Head First Python by Paul Barry / Shroff / O'reilly Publisher
8. Professional Iron Python by John Paul Muller / Wiley India Pvt Ltd
9. Beginning Programming with Python for Dummies by John Paul Muller / Wiley India Pvt Ltd

**Note:** Each session mentioned is for theory and of 2 hours duration. Lab assignments are indicatives, faculty need to assign more assignments for better practice.

**Session 1:**

- Installing Python
- Introduction to Python
- Basic Syntax,
- Data Types, Variables, Operators, Input/output,
- Declaring variable, data types in programs
- Your First Python Program
- Flow of Control (Modules, Branching)
- If, If- else, Nested if-else
- Looping, For, While,
- Nested loops
- Control Structure
- Uses of Break & Continue

## Suggested Teaching Guidelines for Python & R Programming PG-DBDA March 2024

### Lab Assignments:

Q.1. Using for loop, write and run a Python program for this algorithm.

Here is an algorithm to print out  $n!$  ( $n$  factorial) from  $0!$  to  $10!$  :

1. Set  $f = 1$
2. Set  $n = 0$
3. Repeat the following 10 times:
  - a. Output  $n, "!" = "$ ,  $f$
  - b. Add 1 to  $n$
  - c. Multiply  $f$  by  $n$

Q.2. Modify the program above using a while loop so it prints out all of the factorial values that are less than 2 billion. (You should be able to do this without looking at the output of the previous exercise.)

### Session 2:

- Pass, Strings and Tuples
- Accessing Strings
- Basic Operations
- Assigning Multiple Values at Once
- Formatting Strings
- String slices

### Lab Assignments:

Q.1. Write a program that asks the user how many days are in a particular month, and what day of the week the month begins on (0 for Monday, 1 for Tuesday, etc), and then prints a calendar for that month. For example, here is the output for a 30-day month that begins on day 4 (Thursday):

```
S M T W T F S
      1  2  3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

Q. 2. Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:

```
****
*****
*****
```

Q. 3. Write a version of a palindrome recognizer that also accepts phrase palindromes such as "Go hang a salami I'm a lasagna hog.", "Was it a rat I saw?", "Step on no pets", "Sit on a potato pan, Otis", "Lisa Bonet ate no basil", "Satan, oscillate my metallic sonatas", "I roamed under it as a tired nude Maori", "Rise to vote sir", or the exclamation "Dammit, I'm mad!". Note that punctuation, capitalization, and spacing are usually ignored.

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

Q. 4. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

**Session 3:**

- Dictionaries
- Introducing Dictionaries
- Defining Dictionaries
- Modifying Dictionaries
- Deleting Items from Dictionaries

**Lab Assignments:**

Q. 1. In cryptography, a Caesar cipher is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it to communicate with his generals. ROT-13 ("rotate by 13 places") is a widely used example of a Caesar cipher where the shift is 13. In Python, the key for ROT-13 may be represented by means of the following dictionary:

```
key = {'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u', 'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'z', 'n':'a', 'o':'b',  
'p':'c', 'q':'d', 'r':'e', 's':'f', 't':'g', 'u':'h', 'v':'i', 'w':'j', 'x':'k', 'y':'l', 'z':'m', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R',  
'F':'S', 'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A', 'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G',  
'U':'H', 'V':'I', 'W':'J', 'X':'K', 'Y':'L', 'Z':'M'}
```

Your task in this exercise is to implement an encoder/decoder of ROT-13. Once you're done, you will be able to read the following secret message:

Pnrfne pvcure? V zhpu cersre Pnrfne fnynq!

Note that since English has 26 characters, your ROT-13 program will be able to both encode and decode texts written in English.

**Session 4:**

- Working with Lists
- Introducing Lists
- Defining Lists
- Declare, assign and retrieve values from Lists
- Accessing list
- Operations in Lists
- Adding Elements to Lists
- Searching Lists
- Deleting List Elements
- Using List Operators

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

- Mapping Lists
- Joining Lists and Splitting Strings
- Historical Note on String Methods

**Lab Assignments:**

1. Find the largest and smallest number in the list which taken as input from user using list operations.
2. Write a Python program to create multiple lists

**Session 5 & 6:**

- Function and Methods
- Defining a function
- Calling a function
- Types of functions
- Function Arguments
- Anonymous functions(Lambda, Map, List comprehension)
- Global and local variables
- Using Optional and Named Arguments
- Using type, str, dir, and Other Built-In Functions
- Concepts of Modules

**Lab Assignments:**

Q. 1. Given a dictionary of students and their favourite colours:

```
people={'Arham':'Blue','Lisa':'Yellow','Vinod':'Purple','Jenny':'Pink'}
```

1. Find out how many students are in the list
2. Change Lisa's favourite colour
3. Remove 'Jenny' and her favourite colour
4. Sort and print students and their favourite colours alphabetically by name

Write a function `translate()` that will translate a text into "rövarspråket" (Swedish for "robber's language"). That is, double every consonant and place an occurrence of "o" in between. For example, `translate("this is fun")` should return the string "tothohisos isos fofunon".

Q. 2. Write a program that contains a function that has one parameter, `n`, representing an integer greater than 0. The function should return `n!` (`n` factorial). Then write a main function that calls this function with the values 1 through 20, one at a time, printing the returned results. This is what your output should look like:

```
1      1
2      2
3      6
4     24
5    120
6   720
7  5040
8 40320
```

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

9        362880  
10       3628800

Q. 2. We can define sum from 1 to x (i.e.  $1 + 2 + \dots + x$ ) recursively as follows for integer  $x \geq 1$ :

1, if  $x = 1$   
 $x + \text{sum from 1 to } x-1$  if  $x > 1$

Complete the following Python program to compute the sum  $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$  recursively:

```
def main():  
    # compute and print  $1 + 2 + \dots + 10$   
    print sum(10)  
def sum(x):  
    # you complete this function recursively main ()
```

Q. 3. Define a function overlapping () that takes two lists and returns True if they have at least one member in common, False otherwise.

Q. 4. Write a function find\_longest\_word() that takes a list of words and returns the length of the longest one.

Q. 5. Write a function filter\_long\_words() that takes a list of words and an integer n and returns the list of words that are longer than n

Q. 6. Define a simple "spelling correction" function correct () that takes a string and sees to it that  
1) two or more occurrences of the space character is compressed into one, and  
2) inserts an extra space after a period if the period is directly followed by a letter.  
e.g. correct ("This is very funny and cool.Indeed!") should return "This is very funny and cool. Indeed!"

Q. 7. In English, present participle is formed by adding suffix -ing to infinite form: go -> going. A simple set of heuristic rules can be given as follows:

- If the verb ends in e, drop the e and add ing (if not exception be, see, flee, knee, etc.)
- If the verb ends in ie, change ie to y and add ing
- For words consisting of consonant-vowel-consonant, double the final letter before adding ing
- By default, just add ing

Your task in this exercise is to define a function make\_ing\_form() which given a verb in infinitive form returns its present participle form. Test your function with words such as lie, see, move and hug. However, you must not expect such simple rules to work for all cases.

**Session 7:**

- Working with Tuples
- Introducing Tuples
- Accessing tuples

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

- Operation

**Lab Assignments:**

1. Write a Python program to find the repeated items of a tuple.
2. Write a Python program to sort a tuple by its float element.  
Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]  
Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]
3. Write a Python program to count the elements in a list until an element is a tuple.  
Sample input : list = [10, 20, 30, (40,50), 60]  
Sample output = 3
4. Write a Python program to compute element-wise sum of given tuples, using “zip()” function  
Original tuples:  
(1, 2, 3, 4)  
(3, 5, 2, 1)  
(2, 2, 3, 1)  
Element-wise sum of the said tuples:  
(6, 9, 8, 6)

**Session 8 & 9:****Advanced Python:**

- Object Oriented Python
- OOPs concept
- What's an Object?
- Indenting Code
- Native Data types
- Declaring variables
- Referencing Variables
- Object References
- Class and object
- Attributes, Inheritance
- Overloading & Overriding
- Data hiding
- Regular Expressions Using python
- Object Oriented Linux Environment

**Lab Assignments:**

1. Create a class 'Student' with rollno, studentName, course ,dictionary of marks(subjectName -> marks [5]). Provide following functionalities  
A. initializer  
B. override \_\_str\_\_ method  
C. accept student data  
D. Print student data for given id.  
E. Print Student who has failed in any subject. Write menu driven program to test above functionalities.( accept records of 5 students and store those in list )
2. Write a menu driven program to maintain student information. for every student store studentid, sname, and m1,m2,m3 marks for 3 subject. also store gpa in student class, add a function in student class to return GPA of a student

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

- Calculate GPA()
- $\text{gpa} = (1/3) * m1 + (1/2) * m2 + (1/4) * m3$
- Create an array to store Multiple students.
  1. Display All Student
  2. Search by id
  3. Search by name
  4. calculate GPA of a student
  5. Exit

**Session 10 & 11:**

- Operations Exception
- Exception Handling
- Except clause
- Try finally clause
- User Defined Exceptions
- Logging in Python

**Lab Assignments:**

1. Create a short program that prompts the user for a list of grades separated by commas. Split the string into individual grades and use a list comprehension to convert each string to an integer. Use a “try” statement to inform the user when the values they entered cannot be converted.
2. Investigate what happens when there is a “return” statement in both try clause and finally clause of try statement.
3. Create a file named “data.txt”, Open it for reading using python, use try block to catch exception that arises when file doesn’t exist.

**Session 12, 13 & 14:**

- Working with Pandas
- Data wrangling with Pandas
- Working with NumPy
- Data cleaning with Python

**Lab Assignments:**

1. Write a NumPy program to read a CSV data file and store records in an array.
2. Write a NumPy program to convert a PIL Image into a NumPy array. Also convert a NumPy array to an image. Display the image.  
Sample Output:  
[[[255 255 255 0]  
.....  
[255 255 255 0]]]
3. Write a NumPy program to convert Pandas dataframe to Numpy array with headers.
4. Write a Pandas program to read a dataset from diamonds
  - DataFrame and modify the default columns values and print the first 6 rows
  - calculate the mean of each numeric column of diamonds DataFrame.
  - calculate count, minimum, maximum price for each cut of diamonds DataFrame
  - print a concise summary of diamonds DataFrame.
  - count the duplicate rows of diamonds DataFrame.

*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

**Session 15 & 16:**

- Working with beautiful soup
- Working with matplotlib, seaborn
- Working with ggplot, plotly

**Lab Assignments:**

1. Extract any website data using Beautiful Soup.
2. Import csv file using pandas, find correlation and plot heatmap of correlation using seaborn, plot the scatter plot for any two highest correlated columns using matplotlib and plotly.

**Session 17:**

- Load Images using pillow
- Load audio files using scikit-learn(scipy.io)

**Lab Assignments:**

1. Load an Image using Pillow library,
  - Rotate the Image by 90 degrees
  - State the use of convert and palette functions and use them on loaded image
  - Split the image
  - Import Different filters using Pillow library and apply those filters on loaded Image

**Session 18:**

- Connecting DB's with Python
- Working with DB's using Python
- Accessing and Manipulating DB's
- Creation of Python virtual Environment

**Lab Assignments:**

1. Create Python virtual environment.
2. Establish connection between MySQL/MSSQL and Python.
3. Load Data file using mysql query into python environment.

**R-Programming:****Session 19:**

- The R project for Statistical Computing
- Why R
- Introduction & Installation of R
- R Basics, Finding Help,
- Code Editors for R,
- Exploring R Gui
- Exploring RStudio
- Basic Mathematical & Arithmetic operations in R

**Assignments:**

1. Read Introduction to R, R basics and Scope of R.
2. Explore R GUI and Rstudio



*Suggested Teaching Guidelines for*  
**Python & R Programming**  
**PG-DBDA March 2024**

**Session 20:**

- Data Objects- Data Types & Data Structures (e.g. lists, Arrays, matrices, data frames)
- Packages in R
- Working with Packages
- Handling Data in R Workspace
- Reading & Importing data from Text files, Excel files, Multiple databases
- Exporting Data from R

**Lab Assignments:**

1. Write R program to compute sum, mean and product of a given vector elements.
2. Write R program to call the (built-in) dataset airquality.
3. Write R program to create a list of dataframes and access each of those data frames from the list.
4. Write R Program to create dataframe,
  - Get the statistical Summary and nature of the data of dataframe
  - Add new column in dataframe.
  - Sort dataframe using multiple columns.
  - Export Dataframe to excel file using writexl package.

**Session 21:**

- Introduction to tidy verse (group of packages)
- Manipulating and Processing Data in R
- Creating, Accessing and Sorting data frames
- Extracting, Combining, Merging, reshaping data frames

**Lab Assignments:**

1. Load one XML file and one Json file in R studio, Print the data in both files one by one and get statistical summary of data.
2. Extract Data from any website using R packages.
3. Call the built in dataset “diamonds” in R and plot Pie chart and Bar graph

**Session 22:**

- Functions
- Built in functions in R (numeric, character, statistical)
- Interactive reporting with R markdown
- Case study

**Lab Assignments:**

1. Examine the built in ChickWeight data (the help gives background about the data). The function split will prove useful to do the following (as will a script).
  - Construct a plot of weight against time for chick number 34.
  - For chicks in diet group 4, display box plots for each time point.
  - Compute the mean weight for chicks in group 4, for each time point. Plot this mean value against time.
  - Repeat the previous computation for group 2. Add the mean for group 2 to the existing plot.
  - Add a legend and a title.
  - Copy and paste the graph into Word.