Suggested Teaching Guidelines for

Python Programming & Advance Analytics – PG-DBDA Feb 19

Duration: 44 class room hours + 46 lab hours

Objective: To introduce the student to Python programming & advance analytics

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. Introduction to Computer Science using Python, Charles/ Wiley

Reference Book:

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

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.

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.

Suggested Teaching Guidelines for

Python Programming & Advance Analytics – PG-DBDA Feb 19

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

Session 5:

- Function and Methods

- Defining a function
- Calling a function
- Types of functions
- Function Arguments
- Anonymous functions
- Global and local variables
- Using Optional and Named Arguments
- Using type, str, dir, and Other Built-In Functions

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
9	362880
10	3628800

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

1, if x = 1
x + 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 + ... + 10
print sum(10)
def sum(x):
# you complete this function recursively main()
```

Suggested Teaching Guidelines for

Python Programming & Advance Analytics – PG-DBDA Feb 19

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 6:

- Working with Tuples
- Introducing Tuples
- Accessing tuples
- Operations

Session 7 & 8:
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

Suggested Teaching Guidelines for

Python Programming & Advance Analytics – PG-DBDA Feb 19

- Attributes, Inheritance
- Overloading & Overriding
- Data hiding
- Regular Expressions Using python
- Object Oriented Linux Environment

Session 9:

- Operations Exception
- Exception Handling
- Except clause
- Try finally clause
- User Defined Exceptions

Session 10 & 11:

- Python Libraries
- Libraries and Functionality Programming
- Debugging basics

Session 12 & 13:

- Working with Pandas
- Data wrangling with Pandas
- Working with Numpy Scipy
- Data cleaning with Python

Session 14 & 15:

- Working with beautiful soup
- Working with matplotlib seaborn
- Working with ggplot, plotly
- Working with selenium & scrapy

Session 16:

- Introduction to Business Analytics using some case studies
- Case studies: Making Right Business Decisions based on data

Session 17:

- Visualization and Exploring Data
- Descriptive Statistical Measures
- Probability Distribution and Data

Session 18:

- Sampling and Estimation
- Statistical Interfaces
- Predictive modeling and analysis

Session 19 & 20:

- Regression Analysis
- Forecasting Techniques
- Simulation and Risk Analysis
- Optimization, Linear, Non linear, Integer

Session 21:

- Decision Analysis
- Strategy and Analytics
- Overview of Factor Analysis
- Directional Data Analytics
- Functional Data Analysis

Session 22:

- Dimensionality issues
- Ridge & lasso regression
- bias/variance trade off
- density, PCA, FA
- feature selection, Bagging and boosting
- Simulation : Monte carlo