GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 2nd Semester
Subject Name: Programming in Python
Subject Code: 629403

With effective from academic year 2020-21

1. Learning Objectives:

- To develop proficiency in creating based applications using the Python Programming Language.
- To be able to understand the various data structures available in Python programming language and apply them in solving computational problems.
- To be able to draw various kinds of data visualization techniques using PyLab, matplotlib and Pandas
- To be able to understand the creation DB API in Python

2. Prerequisites: Basic Concept of Programming Language

3. Contents:

Unit	Course Content	Weightage percentage
Unit I	**Introduction to Python** The basic elements of Python, Objects, expressions and numerical Types, Variables and assignments, IDLE Branching programs, Strings and Input, Iteration Structured Types, Mutability and Higher-order Functions: Tuples, Ranges, Lists and Mutability (Cloning and list comprehension), Strings, Tuples and Lists, Dictionaries	**15%**
Unit II	**Functions, Exception, Modules and Files** **Functions:** Difference between a Function and a Method, Defining a Function, Calling a Function, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object Reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a Function, Recursive Functions, Anonymous Functions or Lambdas (Using Lambdas with filter() Function, Using Lambdas with map() Function, Using Lambdas with reduce() Function), Function Decorators, Generators, Structured Programming, Creating our Own Modules in Python, The Special Variable __name__ **Exceptions**: Errors in a Python Program (Compile-Time Errors, Runtime Errors, Logical Errors), Exceptions, Exception Handling, Types of Exceptions, The Except Block, the assert Statement, User-Defined Exceptions, Logging the Exceptions 20% **Files:** Files, Types of Files in Python, Opening a File, Closing a File, Working with Text Files Containing Strings, Knowing Whether a	**25%**

	File Exists or Not, Working with Binary Files, The with Statement, Pickle in Python, The seek() and tell() Methods	
Unit III	**Classes and Object-oriented Programming** Abstract Data Types and classes, Inheritance, Encapsulation and Information hiding Mortgages and Extended Examples Case Study: Banking Application	**20%**
Unit IV	**Advanced Topics I: Data Science and Data Visualization using Python** **Data Science Using Python:** Data Frame (Creating Data Frame from an Excel Spreadsheet, Creating Data Frame from .csv Files, Creating Data Frame from a Python Dictionary, Creating Data from Python List of Tuples, Operations on Data Frames) **Data Visualization:** Bar Graph, Histogram, creating a Pie Chart, Creating Line Graph **Plotting:** Plotting using PyLab, Plotting mortgages and extended examples	**15%**
Unit V	**Advanced Topics II: Regular Expressions** **REs and Python:** Regular Expressions, Sequence Characters in Regular Expressions, Quantifiers in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expressions on Files, Retrieving Information from a HTML File Case Study: Screen Scrapping	**25%**
Unit VI (*)	**Python's Database Connectivity** Verifying the MySQL dB Interface Installation, Working with MySQL Database, Using MySQL from Python, Retrieving All Rows from a Table, Inserting Rows into a Table, Deleting Rows from a Table, Updating Rows in a Table, Creating Database Tables through Python	**-**

Note: (*) Only for Practical exam.

4. Text Book(s):

John V Guttag. “Introduction to Computation and Programming Using Python”, 2nd Edition, Prentice Hall of India

1. R Nageswara Rao, Core Python Programming, 2nd Edition, Dreamtech Press

5. Reference Books:

1) Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson
2) Luke Sneeringer, Professional Python, WROX
3) Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python, Pearson

4) Doug Hellmann, The python 3 standard Library by example, Pearson Education
5) Alex Martelli, Python Cookbook, O'REILLY
6) Laura Cassell, Python Projects, WROX
7) Daniel Y Chen, Pandas for Everyone: Python Data Analysis, 1st Edition, Pearson Eduction

Web Resources:

1) Charles Severance, Python for informatics: www.pythonlearn.com
2) Swaroop C H. "A Byte of Python", http://www.swaroopch.com/notes/python
3) "Python Programming", http://en.wikibooks.org/wiki/Python_Programming
4) "The Python Tutorial", http://docs.python.org/release/3.0.1/tutorial/
5) "Learn Python the Hard way", http://learnpythonthehardway.org/
6) Dive into Python 3: http://www.diveintopython.net/

6. Unit wise coverage from Textbook(s):

Unit #	Book#	Chapter
I	1	Chapter : 2,5 (Except 5.4)
II	2	Chapter :9,16, 17(Pages 441 to 456)
III	1	Chapter : 8
IV	1	Chapter : 11
	2	Chapter : 25
V	2	Chapter : 18
VI	2	Chapter : 24 (Pages 663 to 681)

Accomplishment

At the end of the course, the student should be able to:

- Ability to create robust applications using the Python programming language
- Ability to create applications for solving computational problems using the Python Programming Language

Practical List

Tools: Python 3.x, IDLE

Part I: Core Python

A Basics

1	Write a Python Program to Convert Celsius to Fahrenheit and vice –a-versa.
2	Write a program in python to swap two variables without using temporary variable.

3	Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal
4	Write a program to make a simple calculator (using functions).
5	Write a program in python to find out maximum and minimum number out of three user entered number.
6	Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found.
7	Write a Python program to check if the number provided by the user is an Armstrong number.
8	Write a Python program to check if the number provided by the user is a palindrome or not.
9	Write a Python program to perform following operation on given string input: a) Count Number of Vowel in given string b) Count Length of string (do not use Len ()) c) Reverse string d) Find and replace operation e) check whether string entered is a palindrome or not
10	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: **** ********* *******
11	Write a program in python to implement Fibonacci series up to user entered number. (Use recursive Function)
12	Write a program in python to implement Factorial series up to user entered number. (Use recursive Function)
13	Write a program in Python to implement readline, readlines, write line and writelines file handling mechanisms.

B. **Advanced**

14	Write a program in python to implement Salary printing file read operation. (File format: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations. a) Print Salary Slip for given Employee Number b) Print Employee List for Given Department Number
15	Write a program in python to implement Railway Reservation System using file handling technique. System should perform below operations. a. Reserve a ticket for a passenger. b. List information all reservations done for today’s trains.
16	Write a Python program to implement module.
17	Write a program which will implement decorators for functions and methods in python.
18	Write a program to read CSV file and generate output using HTML table.
19	Write a program to process CSV file using CSV module.
20	Desirable: Write a program to process JSON and XML data.

21	Create Web Database Application “Address Book” with options to a) add/ insert a record b) modify a record c) display a record d) delete a record
22	Create Web Database Application “Event Registration” with options to a) Event Registration b) Cancel Registration c) display a record

Part II: Advanced Topic: Data Analysis

1	Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame d. Finding maximum and minimum values e. Displaying statistical information f. Performing queries g. Data Analysis using groupby()

Part III: Advanced Topic: Data cleaning

1	Handling dirty data / missing data

Part IV: Advanced Topic: Python for Data Visualization
Library: pylab, matplotlib, seaborn

1	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)
2	Using a data file, draw a. Bar Graph b. Histogram c. Pie Chart d. Line Chart
3 **(*)**	Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame d. Finding maximum and minimum values e. Displaying statistical information f. Performing queries g. Handling missing data

(*): Topics from Books 2 (Pages 694 to 705)

Part V: Advanced Python Programming: Regular Expressions

1	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)
	a) Recognize following strings bit, but, bat, hit, hat or hut
	b) Match any pair of words separated by a single space, that is, first and last names.
	c) Match any word and single letter separated by a comma and single space, as in last name, first initial.
	d) Match simple Web domain names that begin with www and end with a ".com" suffix; for example, www.yahoo.com. Extra Credit: If your regex also supports other high-level domain names, such as .edu, .net, etc. (for example: www.foothill.edu).
	e) Match a street address according to your local format (keep your regex general enough to match any number of street words, including the type designation). For example, American street addresses use the format: 1180 Bordeaux Drive. Make your regex flexible enough to support multi-word street names such as: 3120 De la Cruz Boulevard.
2	Create utility script to process telephone numbers such that
	a. Area codes (the first set of three-digits and the accompanying hyphen) are optional, that is, your regex should match both 800-555-1212 as well as just 555-1212.
	b. Either parenthesized or hyphenated area codes are supported, not to mention optional; make your regex match 800-555-1212, 555-1212, and also (800) 555-1212.
3	Chapter End Practical List of Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson