VI Semester

Python Programming			
Course Code	21EC643	CIE Marks	50
Teaching Hours/Week (L:T:P:S)	2:0:2:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	3	Exam Hours	3

Course objectives:

- To learn programming using Python
- Develop application using Python

Teaching-Learning Process (General Instructions)

The sample strategies, which the teacher can use to accelerate the attainment of the various course outcomes are listed in the following:

1. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the delivered lessons shall develop student's theoretical and programming skills.
2. State the need for learning Programming with real-life examples.
3. Support and guide the students for self–study.
4. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress
5. Encourage the students for group learning to improve their creative and analytical skills.
6. Show short, related video lectures in the following ways:
   - As an introduction to new topics (pre-lecture activity).
   - As a revision of topics (post-lecture activity).
   - As additional examples (post-lecture activity).
   - As an additional material of challenging topics (pre-and post-lecture activity).
   - As a model solution of some exercises (post-lecture activity).

Module-1
Python Basics, Python language features, History , Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, Flow control, Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit(), Functions, def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number Textbook 1: Chapters 1 – 3

Teaching-Learning Process	Chalk and talk method, Simulation of modulation techniques
	RBT Level: L1, L2, L3

Module-2
Data Structures: Lists: The List Data Type, Working with Lists Strings: Manipulating Strings, Working with Strings, Useful String Methods Tuples and Dictionaries, basics Using Data Structures to Model Real-World Things, Manipulating Strings. Textbook 1: Chapters 4 – 6

Teaching-Learning Process	Chalk and talk method/Power point presentation
	RBT Level: L1, L2, L3

Module-3
Pattern Matching with Regular Expressions, Finding Patterns of Text Without Regular Expressions, Finding Patterns of Text with Regular Expressions, More Pattern Matching with Regular Expressions,, The findall() Method, Character Classes, Making Your Own Character Classes, The Caret and Dollar Sign Characters, The Wildcard Character, Review of Regex Symbols.   Reading and Writing Files, Files and File Paths, The os.path Module, The File Reading/Writing Process, Saving Variables with the shelve Module, Saving Variables with the pprint. pformat() Function   Textbook 1: Chapters 7, 8

Teaching-Learning Process	Chalk and talk method / PowerPoint Presentation **RBT Level:** L1, L2, L3

Module-4
Classes and objects: Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying, Classes and functions: Time, Pure functions, Modifiers, Prototyping versus planning, Classes and methods: Object-oriented features, Printing objects, Another example, The init method, The_str_ method, Operator overloading, Type-based dispatch, Polymorphism.   Textbook 2: Textbook 2: Chapters 15 – 18

Teaching-Learning Process	Chalk and talk method / PowerPoint Presentation **RBT Level:** L1, L2, L3

Module-5
HTTP, The World's simplest Web Browser, Retrieving an image over HTTP, Retrieving web pages with urllib, Parsing html and scraping the web, Parsing HTML using RE, BeautifulSoup, Reading binary files using urllib, XML, Parsing XML, Looping through nodes, JSON, Parsing JSON, API, geocoding Web Service, Security & API usage, What is database?, Database Concepts, Database Browser, Creating a database table, SQL, Spidering Twitter, Basic data modeling, Programming with multiple tables, Three kinds of Keys, JOIN   Text book : Chapter 2, 13, 15

Teaching-Learning Process	Chalk and talk method/Power point presentation **RBT Level:** L1, L2, L3

Course outcomes (Course Skill Set)
At the end of the course the student will be able to:

1. To acquire programming skills in Python
2. To demonstrate data structure representation using Python
3. To develop the skill of pattern matching and files in Python
4. To acquire Object Oriented Skills in Python
5. To develop the ability to write database applications in Python

Assessment Details (both CIE and SEE)

The weightage of Continuous 5 End Examination) taken together.

Continuous Internal Evaluation:

Three Unit Tests each of **20 Marks (duration 01 hour)**

1. First test at the end of 5th week of the semester
2. Second test at the end of the 10th week of the semester
3. Third test at the end of the 15th week of the semester

Two assignments each of **10 Marks**

4. First assignment at the end of 4th week of the semester
5. Second assignment at the end of 9th week of the semester

Group discussion/Seminar/quiz any one of three suitably planned to attain the COs and POs for **20 Marks (duration 01 hours)**

6. At the end of the 13th week of the semester

The sum of three tests, two assignments, and quiz/seminar/group discussion will be out of 100 marks and will be **scaled down to 50 marks**

(to have less stressed CIE, the portion of the syllabus should not be common /repeated for any of the methods of the CIE. Each method of CIE should have a different syllabus portion of the course).

CIE methods /question paper is designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester End Examination:

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the subject (**duration 03 hours**)

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module. Marks scored out of 100 shall be reduced proportionally to 50 marks

---

Suggested Learning Resources:

Text Books:

1. Al Sweigart, "Automate the Boring Stuff with Python",1st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/) (Chapters 1 to 8)
2. Allen B Downey, "Think Python: How to Think Like a Computer Scientist", 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf) (Chapters 15 - 18) (Download pdf/html files from the above links)
3. Charles R. Severance, "Python for Everybody: Exploring Data Using Python 3", 1st, Create Space Independent Publishing Platform, 2016

Web links and Video Lectures (e-Resources)

- https://www.youtube.com/watch?v=_xQNeOTRvig
- https://www.youtube.com/watch?v=kqtD5dpn9C8

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Write a program to generate Fibonacci series
- Write a program to find factorial of a number using function.
- Write a menu driven program to implement stack using Lists
- Create a DB using dictionaries containing key as USN and related fields containing Name, gender, Marks1, Marks2 & Marks3 of students. Implement the following functions to perform i) Update Name/gender/marks ii) search for usn and display the relevant fields iii) delete based on search for name iv)generate the report with avg marks more than 70%
- Write a program to implement search and replace multiple occurrences of a given substring in the main string in a list.
- Write a function called most_frequent that takes a string and prints the letters in decreasing order of frequency.
- Write a program that reads a file, display the contents, builds a histogram of the words in the file and print most common words in the file.
- Write a program that searches a directory and all of its subdirectories, recursively, and returns a list of complete paths for all files with a given suffix.

- Write python code to extract From: and To: Email Addresses from the given text file using regular expressions. https://www.py4e.com/code3/mbox.txt.

- Consider the sentence *"From rjlowe@iupui.edu Fri Jan 4 14:50:18 2008"*, Write python code to extract email address and time of the day from the given sentence

- Write a program to read, display and count number of sentences of the given file.

- Write a program that gets the current date and prints the day of the week.

- Write a function called print_time that takes two Time objects and prints total time it in the form hour:minute:second.

- Write a program that takes a birthday as input and prints the user's age and the number of days, hours, minutes and seconds until their next birthday.