17CS42O1–PYTHON PROGRAMMING

(CSE)

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Category:** | Open Elective | **Credits:** | 3 |
| **Course Type:** | Theory | **Lecture – Tutorial – Practical:** | 2-2-0 |
| **Pre-requisite:** | Require the fundamental concepts of computers and any programming basics | **Sessional Evaluation:**  **Univ.Exam Evaluation:**  **Total Marks:** | 40  60  100 |

|  |  |  |
| --- | --- | --- |
| **Course**  **Objectives:** | Students undergoing this course are expected to: | |
| 1.Learn object oriented programming using an easy to use language  2. Use iterators and generators.  3. Learn to test objects and handle changing requirements.  4. Test cases and handle refactoring to identify its advantages.  5. Exposed to programming over the web to develop various applications.  6.Learn to create and utilize the advantages of packages | |
| **Course Outcomes:** | Upon successful completion of the course, the students will be able to: | |
| **CO1** | Understand the concepts of object oriented programming in python. |
| **CO2** | Compose a group of characters and utilization of strings into various applications |
| **CO3** | Use generators and iterators to develop different applications |
| **CO4** | Develop test cases and handle refactoring to identify its advantages. |
| **CO5** | Use serializing objects to program over the web. |
| **CO6** | Understand to create and utilize the advantages of packages. |
| **Course Content:** | UNIT-I  **Introduction:** Function declaration, import, objects, indenting as requirement, exceptions, unbound variables, case sensitive, scripts, native data types, booleans numbers, Lists: tuples, sets, dictionaries, Comprehensions: list comprehensions, dictionary comprehensions, set comprehensions.  UNIT-II  **Strings**: Strings, unicode, formatting, string methods, bytes, encoding, regular expressions verbose, case studies  UNIT-III  **Classes**: Closures, list of functions, list of patterns, file of patterns, generators, defining classes, instantiating classes, instance variables, iterators, assert, generator expressions  UNIT-IV  **Files**: Reading and writing text files, binary files, stream objects, standard input, output and error.  UNIT-V  **XML and serialization**: XML, atom feed, parsing HTML, searching for nodes, html, generation, serializing objects, pickle files, versions, debugging, serializing to JSON.  UNIT-VI  Packaging python libraries: Directory structure, writing your setup script, classifying your package, examples of good package classifiers, checking your setup script for errors, creating a source distribution, creating a graphical installer, building installable packages for other operating systems, adding your software to the python package index, the many possible futures of python packaging. | |
| **Text books &**  **Reference books:** | **Text books:**  1. “Dive into Python 3”, by Mark Pilgrim, Apress, 2009.  2. “How to think like a computer scientist - learning with python”, by Allen Downey,  Jeffrey Elkner, Chris Meyers, Green Tea Press, 2002.  **Reference books:**  1. “Introduction to computation and programming using python”, by John V. Guttag,  Prentice Hall of India, 2014  2. “Learning python: Powerful object-oriented programming”, by Mark Lutz,  5th Edition, O’Reilly, Shroff Publishers and Distributors, 2013 | |
| **e-Resources** | 1.<https://nptel.ac.in/courses>  2.<https://freevideolectures.com/university/iitm> | |