|  |
| --- |
| **COURSE FILE** |

***Faculty Details***

|  |  |
| --- | --- |
| Name of the Faculty | N. KATHIRVEL |
| Designation | TEACHING FELLOW |
| Department | IT/CSE |

***Course Details***

|  |  |  |  |
| --- | --- | --- | --- |
| Name of the Programme | B.TECH. PHARMACEUTICAL TECHNOLOGY | Batch | 2018-2022 |
| Semester & Year | I& I | No. of Students | 49 |
| Subject Code & Name | GE8151-PROBLEM SOLVING AND PYTHON PROGRAMMING | | |

**PROGRAMME EDUCATIONAL OBJECTIVES (PEO’s)**

**PEO-1** Provide proficiency in technical knowledge to responsibly and critically analyze to solve

the technological problems

**PEO-2** Motivate research and development activities to develop novel products and provide

sustainable solutions to meet the societal needs

**PEO-3** Provide high professionalism to work in diverse and innovative environments with

Modern tools

**PEO-4** Develop ethical attitude, provide communication and managerial skills, and induce the

ability for life-long learning.

**PROGRAMME OUTCOMES (PO’s)**

**After completion of graduation in Pharmaceutical Technology, the students will be able to demonstrate the ability to:**

1. Apply knowledge of mathematics, science and technology in the discipline.
2. Identify, formulate, research literature, and analyse complex engineering problems for its solution.
3. Design and develop system processes that meet the specified needs with appropriate consideration for public health, safety, cultural, societal, and environmental.
4. Design the experiments, its analysis and interpretation of data, synthesis of the information using research-based knowledge for complex problems.
5. Use modern engineering tools, software and equipment to meet the needs in the area of Pharmaceutical Technology.
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues relevant to the professional engineering practices.
7. Apply knowledge of the impact of pharmaceutical technology solutions in a societal and global context.
8. Demonstrate ethical principles and commitment to responsibilities and norms of the Pharmaceutical technology practices.
9. Work effectively as an individual and as well as member in teams of diversified professionals.
10. Communicate effectively.
11. Understand the philosophies of project management principles in Pharmaceutical technology.
12. Showcase urge for self-education and life-long learning

. **SYLLABUS**

**UNIT I ALGORITHMIC PROBLEM SOLVING 9**

Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

**UNIT II DATA, EXPRESSIONS, STATEMENTS 9**

Python interpreter and interactive mode; values and types: int, float, Boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

**UNIT III CONTROL FLOW, FUNCTIONS 9**

Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

**UNIT IV LISTS, TUPLES, DICTIONARIES 9**

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, merge sort, histogram.

**UNIT V FILES, MODULES, PACKAGES 9**

Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file. TOTAL : 45 PERIODS

**TEXT BOOKS:**

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist’’, 2ndedition, Updated for Python Shroff /O’Reilly Publishers, 2016 (<http://greenteapress.com/wp/thinkpython/>
2. 2. Guido van Rossum and Fred L. Drake Jr, “An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.

**REFERENCES**:

3. Charles Dierbach, “Introduction to Computer Science using Python: A Computational Problem-Solving Focus, Wiley India Edition, 2013.

4. John V Guttag, “Introduction to Computation and Programming Using Python’’, Revised and expanded Edition, MIT Press , 2013

5. Kenneth A. Lambert, “Fundamentals of Python: First Programs”, CENGAGE Learning, 2012.

6. Paul Gries, Jennifer Campbell and Jason Montojo, “Practical Programming: An Introduction to Computer Science using Python 3”, Second edition, Pragmatic Programmers, LLC, 2013.

7. Robert Sedgewick, Kevin Wayne, Robert Dondero, “Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.

8. Timothy A. Budd, “Exploring Python”, Mc-Graw Hill Education (India) Private Ltd.,, 2015.

**COURSE OBJECTIVES & OUTCOMES**

**COURSE OBJECTIVES**

The student should be made to:

* To know the basics of algorithmic problem solving
* To read and write simple Python programs.
* To develop Python programs with conditionals and loops.
* To define Python functions and call them.
* To use Python data structures –- lists, tuples, dictionaries.
* To do input/output with files in Python

**COURSE OUTCOMES**

Upon completion of the course, the student should be able to:

* Develop algorithmic solutions to simple computational problems.
* Read, write, execute by hand simple Python programs.
* Structure simple Python programs for solving problems.
* Decompose a Python program into functions.
* Represent compound data using Python lists, tuples, and dictionaries.
* Read and write data from/to files in Python Programs.

**Mapping CO-PO**

Mapping CO – PO:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 3 | 2 | 3 | 3 | 3 | 2 | 3 |  |  |  |  |  |
| CO2 | 2 | 3 | 2 | 3 |  | 2 | 2 |  |  |  |  |  |
| CO3 | 3 | 2 | 2 | 2 | 2 | 2 | 1 |  |  |  |  |  |
| CO4 | 2 | 2 | 2 | 2 | 1 | 1 | 1 |  |  |  |  |  |
| CO5 | 3 | 3 | 1 | 1 | 2 | 1 | 1 |  |  |  |  |  |
| CO6 | 2 | 3 | 2 | 3 |  | 2 | 2 |  |  |  |  |  |

3– Excellent ; 2 – Good ; 1 - Average

**Staff HOD**