| Course Code: | 22CS1ESPYP | Course Title: | Introduction to Python Programming |
|---|---|---|---|
| Credits: | 3 | L – T – P | 2-0-1 |

**Course objectives:**

The objectives of the course are to facilitate the learners to

• Master the fundamentals of writing Python scripts, learn core Python scripting elements such as variables and flow control structures, discover how to work with lists and sequence data.

• Write Python functions to facilitate code reuse, make their code robust by handling errors and exceptions properly, Explore Python's object-oriented features, Search text using regular expressions, Use Python to read and write files

**Teaching-Learning Process (General Instructions)**

Given below are sample strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecture method (L) does not mean only the traditional lecture method, but a different type of teaching method may be adopted to develop the outcomes.
2. Show Video/animation films to explain the programming concepts such as iteration, slicing and recursion.
3. Demonstration of program execution and debugging of errors.
4. Encourage collaborative (Group) Learning in the class.

| UNIT-1 | [5 hours] |
|---|---|

**Python Basics**: Variables, expressions, and statements: Values and types, Variables, Variable names and keywords, Statements, Operators and operands, Expressions, Order of operations, Modulus operator, String operations, Asking the user for input, Comments, choosing mnemonic variable names, Debugging,
**Conditional execution:** Boolean expressions, Logical operators, Conditional execution, Alternative execution, chained conditionals, Nested conditionals, catching exceptions using try and except, Short-circuit evaluation of logical expressions
**Iteration:** Updating variables, the while statement, Infinite loops, break, finishing iterations with continue, Definite loops using for, Loop patterns, Counting and summing loops, Maximum and minimum loops

| UNIT-2 | [5 hours] |
|---|---|

**Strings:** A string is a sequence, Getting the length of a string using len, Traversal through a string with a loop, String slices, Strings are immutable, Looping and counting, The in operator, String comparison, string methods, Parsing strings, Format operator
**Lists:** A list is a sequence, Lists are mutable, Traversing a list, List operations, List slices, List methods, Deleting elements, Lists and functions, Lists and strings, Parsing lines, Objects and values, Aliasing, List arguments

| UNIT-3 | [5 hours] |
|---|---|

**Dictionaries:** Dictionary as a set of counters, Dictionaries and files, Looping and dictionaries, Advanced text parsing
**Tuples:** Immutable, comparing tuples, Tuple Assignment, Dictionaries and Tuples, Multiple Assignments with Dictionaries, Using Tuples as keys in Dictionary
**Functions:** Function calls, Built-in functions, Type conversion functions, Random numbers, Math functions, Adding new functions, Definitions and uses, Flow of execution, Parameters and arguments, Fruitful functions and void functions, Why functions

| UNIT-4 | [5 hours] |
|---|---|

**Object-Oriented Programming:** Managing Larger Programs, Getting Started, Using Objects, starting with Programs, Subdividing a Problem, Our First Python Object, Classes as Types, Object Lifecycle, Many Instances, Inheritance, Classes and Methods, Operator overloads
**Exceptions**: Exception Class Hierarchy, User-Defined Exceptions

| UNIT-5 | [ hours] |
|---|---|
| **Regular expressions:** Character matching in regular expressions, extracting data using regular expressions, combining searching and extracting, Escape character<br><br>**Files:** Persistence, opening files, Text files and lines, reading files, searching through a file, Letting the user choose the file name, Using try, except, and open, Writing files | |

## LIST OF LAB PROGRAMS

| Q.No | Lab Programs |
|---|---|
| 1 | Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Make sure to ask the user to enter the number of numbers in the sequence to generate. |
| 2 | Write a program that asks the user for a number and then prints out a list of all the divisors of that number. |
| 3 | Write a program to compute distance between two points taking input from the user (Pythagorean Theorem). |
| 4 | Write a Program for checking whether the given number is a even number or not. |
| 5 | Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero. |
| 6 | Write a program to find the sum of all primes below two million. |
| 7 | **a)** Write a program to count the numbers of characters in the string and store them in a dictionary data structure.<br>**b)** Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure. |
| 8 | **a)** Write a Python program that takes this list and makes a new list that has only the even elements of this list in it.<br>**b)** Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate Boolean. |
| 9 | **a)** Write a program combine_lists that combines these lists into a dictionary.<br>**b)** Write a program to print each line of a file in reverse order. |
| 10 | **a)** Write a program to count frequency of characters in a given file.<br>**b)** Write a program to compute the number of characters, words and lines in a file |

## Course outcomes (Course Skills Set)

| Course Code | CO | COURSE OUTCOME (CO) |
|---|---|---|
| | CO 1 | Apply knowledge of Python programming for various applications |
| | CO 2 | Analyse the given Python program to identify bugs |
| | CO 3 | Design Python programs/ applications for a given requirement. |
| | CO 4 | Ability to conduct practical experiments for given requirements using python. |

## COs and POs Mapping

| COs | POs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | | | | | | | | | | | |
| **CO2** | | 3 | | | | | | | | | | |
| **CO3** | | | 3 | | | | | | | | | |
| **CO4** | | | | 3 | | | | | | | | |

**Assessment Details (both CIE and SEE)**

| Assessment Plan (for 50 marks of CIE) | | |
|---|---|---|
| **Tool** | **Remarks** | **Marks** |
| **Internals** | **Two** | **20** |
| **AAT** | **Mini-project** | **5** |
| **Lab Component** | Lab CIE + Lab Test | **25** |
| **Self-Study Component** | *NA* | -- |
| **Total** | | **50** |

**Semester End Examination: (QP PATTERN)**

| Unit-1 | Mandatory | One Question to be asked for 20 Marks |
|---|---|---|
| Unit-2 | Mandatory | One Question to be asked for 20 Marks |
| Unit-3 | Internal Choice | Two Questions to be asked for 20 Marks each |
| Unit-4 | Internal Choice | Two Questions to be asked for 20 Marks each |
| Unit-5 | Mandatory | One Question to be asked for 20 Marks |

**Suggested Learning Resources:**

| Prescribed Text Book | | | | | |
|---|---|---|---|---|---|
| Sl. No | Book Title | Authors | Edition | Publisher | Year |
| 1 | Python for Everybody: Exploring Data Using Python 3 | Charles R. Severance | Fourth | University of Michigan | 2016 |
| 2 | Learning to Program using Python | Cody Jackson | Second | Packt Publishing | 2018 |

| Reference Text Book | | | | | |
|---|---|---|---|---|---|
| Sl. No | Book Title | Authors | Edition | Publisher | Year |
| 1 | Programming Python | Mark Lutz | First | O'Reilly Media | 2010 |
| 2 | Python Essential Reference | David M. Beazley | Fourth | Pearson | 2009 |
| 3 | Core Python Applications Programming | Wesley J Chun | Third | Pearson | 2015 |

| E-Book | | | | | |
|---|---|---|---|---|---|
| Sl. No | Book Title | Authors | Edition | Publisher | Year | URL |

| 1 | Think Python | Allen B. Downe | Second | Green Tea Press, Needham, Massachus etts | 2014 | https://greenteapress.com/ thinkpython2/thinkpython 2.pdf |
|---|---|---|---|---|---|---|
| 2 | A Hands-On, Project-Based Introductio n to Programmin g | Eric Matthes | First | No Starch Press | 2016 | https://t.ly/fEOq (URL Shortened) |

| MOOCS Courses | | | | |
|---|---|---|---|---|
| Sl. No | Course name | Course Offered by | Year | URL |
| 1 | An Introduction to Interactive Programming in Python (Part 1) | Coursera | 2021 | https://www.coursera.org/ course/interactivepython1 |
| 2 | An Introduction to Interactive Programming in Python (Part 2) | Coursera | 2021 | https://www.coursera.org/ course/interactivepython2 |
| 3 | Introduction to Python Programming | **edx** | **2021** | https://www.edx.org/profe ssionalcertificate/introductio n- topython-programming |

\*\*\*\*\*\*