

<b>Name of Faculty</b>	:	Faculty of Engineering & Technology
<b>Name of Program</b>	:	Bachelor of Technology (B. Tech)
<b>Course Code</b>	:	1BIT01
<b>Course Title</b>	:	Python Programming
<b>Type of Course</b>	:	Basic Engineering (BE)
<b>Year of Introduction</b>	:	2023-24

<b>Prerequisite</b>	:	High level language (C/C++/Java), Web Programming
<b>Course Objective</b>	:	Develop a strong foundation in Python programming language, including its syntax, data types, and control structures
<b>Course Outcomes</b>	:	At the end of this course, students will be able to:
	CO1	Interpret the fundamental python syntax, semantics and fluent in the use of python control flow statements. Express proficiency in the handling of strings and functions
	CO2	Determine the methods to create and manipulate python programs by utilizing the data structures like lists, dictionaries, tuples and sets
	CO3	Identify the commonly used operations involving file systems and regular expressions
	CO4	Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism as used in Python along with magic methods

### Teaching and Examination Scheme

Teaching Scheme (Contact Hours)			Credits	Examination Marks				
L	T	P		Theory Marks		Practical Marks		Total Marks
			C	SEE	CIA	SEE	CIA	
2	0	2	3	70	30	30	20	150

*Legends: L-Lecture; T-Tutorial/Teacher Guided Theory Practice; P - Practical, C - Credit, SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)*

### Course Content

Unit No.	Topics	Teaching Hours	Weightage	Mapping with CO
1	<b>Basics of Python:</b> Using the Python Interpreter, Variables, Identifiers and Keywords, Numbers and Expressions	02	10%	CO1
2	<b>Data Structures: Lists, Tuples, Dictionaries and Strings:</b>	05	16%	CO1 CO2

	Common Sequence Operations: Indexing, Slicing, Adding Sequences, Multiplication, Membership, Length, Minimum, and Maximum, Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, the del statement, Tuples and Sequences, Sets, Dictionaries, Comparing Sequences and Other Types, Basic String Operations			
3	<b>Control structures and Function:</b> Conditional Branching: if Statements, break and continue Statements, and else Clauses on Loops, pass Statements Loops: while Loops, for Loops, Defining Functions, More on Defining Functions: Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations	05	16%	CO3
4	<b>Modules and Scoping Rules:</b> Executing modules as scripts, The Module Search Path, "Compiled" Python files, Packages: Importing * From a Package, Intra-package References, Packages in Multiple Directories	02	10%	CO3 CO4
5	<b>Exceptions Handling::</b> Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions	05	15%	CO4
6	<b>Magic Methods, Properties, and Iterators:</b> Constructors, Item Access: The Basic Sequence and Mapping Protocol, Properties: The property Function, Static Methods and Class Methods, getattr, setattr, and Friends, Iterators, Generators, Generator Expressions	05	15%	CO4
7	<b>Regular Expression and File Handling:</b> What is a regular expression?, Regular expressions with special characters, Regular expressions and raw strings, Extracting matched text from strings, Substituting text with regular expressions, Writing and Reading Binary Data, Writing and Parsing Text Files, Iterating over File Contents, Writing and Parsing XML Files, Random Access Binary Files	06	18%	CO4

Suggested Distribution of Theory Marks Using Bloom's Taxonomy						
Level	Remembrance	Understanding	Application	Analyse	Evaluate	Create
Weightage	20	30	30	20	0	0

NOTE: This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

#### Suggested List of Experiments/Tutorials

Sr. No.	Name of Experiment/Tutorial	Teaching Hours
1	Experiment on Lists	02
2	Experiment on Tuples	02
3	Experiment on Dictionaries	02
4	Experiment on Strings	02
5	Experiment on Control Structures and iterators	04
6	Experiment on Functions and magic methods	02
7	Experiment on Modules and scoping rules	02
8	Experiment on Exception handling	04
9	Experiment on Regular expressions	02
10	Experiment on file handling	08

#### Major Equipment/ Instruments and Software Required

Sr. No.	Name of Major Equipment/ Instruments and Software
1	Python IDLE
2	Anaconda Python
3	PyCharm

#### Suggested Learning Websites

Sr. No.	Name of Website
1	<a href="https://www.python.org/">https://www.python.org/</a>
2	<a href="http://www.diveintopython3.net/">http://www.diveintopython3.net/</a>
3	<a href="https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django">https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django</a>
4	<a href="https://www.fullstackpython.com/django.html">https://www.fullstackpython.com/django.html</a>

#### Reference Books

Sr. No.	Name of Reference Books
1	David Beazley, Brian K. Jones, "Python Cookbook", 3rd edition, O'REILLY, 2016
2	Brett Slatkin, "Effective Python: 59 Specific Ways to Write Better Python", Novatec, 2016
3	Allen Downey, "Think Python: How to Think Like a Computer Scientist"
4	Mark Lutz "Learning Python", 4th Edition, O'REILLY, 2016