	Program	**B. Tech Computer Science and Engineering**					**Program Code**	**24BTHCSE**
	Course	**Programming in Python**					**Course Code**	**24BTELY204/ 205**
	Semester	**I**	**Credits**	**4**	**Theory**	**Practical**	**Total Hours**	**60**
					3	**2**		

	COURSE OBJECTIVES:
1)	Understand the core syntax and semantics of Python programming language.
2)	Equip students with the need for working with the strings and functions.
3)	Familiarize the process of structuring the data using lists—dictionaries—tuples and sets.
4)	Illustrate the use of regular expressions and File handling process

	General Instructions for Teaching-Learning:
1)	Appropriate Models, PowerPoint presentation, Videos, shall be used to enhance visualization before hands on practice.
2)	Emphasize hands-on coding practice from the beginning.
3)	Encourage peer collaboration through pair programming, code reviews, and group projects.
4)	Curate a collection of self-paced learning resources, including online tutorials, documentation, and video lectures, to support students with different learning preferences and backgrounds.

Module	**Topics**	**Hours**
I	**Introduction To Python** Introduction: Introduction to Python programming, writing a Python program, Python Interactive Shell, Building blocks, Values and Variables, Variables and Assignment, Identifiers, Expressions: Arithmetic, Logical, Relational, Operators and operands, Order of operations, Control codes within strings, Composition, Comments, User input and Output, Errors in Python: syntax errors, run-time exceptions, and logic error. Control Structures: Conditional expressions, Simple IF statement, IF...ELSE statement, Compound Boolean expressions, pass statement, Nested conditionals, Multi-way Decision statements, Iteration statement , While, for statement, nested loops, break and continue statements, while/else and for/else statements.	9
II	**Functions** Functions and Modules: Function calls, Built-in Functions, Standard Mathematical functions, Random functions, Time functions, system specific functions, eval and exec functions, writing functions, Global variables, Default parameters, importing a module using import statement, Functions as Data, return statement, Anonymous function using Lambda expression—Generators, Local function definitions, Recursive function.	9
III	**Lists—Tuples—Sets** Lists, Operations on list: Indexing, Slicing, Concatenation, Repetitions, Updating, Membership, Comparison, List methods, list loops, List and Functions, Cloning lists, Multidimensional lists, Nested lists, Comparison between Lists and Generators. Tuples: Tuple assignment, Tuples as return values, Mutability and tuples, Comparison between Lists and Tuples, Sets: Creating set, set methods, set operations	9
IV	DICTIONARIES AND STRINGS:	9

	The Dictionary Data Type, Dictionary methods, Pretty Printing, Using Data Structures to model Real-world things, Illustrative examples of Dictionaries. Working with Strings, Double Quotes, Escape Characters, Raw strings, Multi-line strings with triple quotes, Indexing and slicing strings, in and not in operator with strings; Putting strings inside other strings, Useful string methods**:** upper(), lower(), isupper(), and islower() String Methods **,** isX String Methods , startswith() and endswith() String Methods, join() and split(), rjust(), ljust(), and center(),strip(), rstrip(), and lstrip() string methods.	
V	Files handling and Regular Expressions Files: Text files, writing a text file, reading from a text file, writing a binary file, reading from a binary file, seek(). Forming Regular expression, Matching, Searching and replacing operations on RE-Defining regular expressions, compiling regular expressions, using regular expressions, using match objects to extract a value, Extracting multiple items, Replacing multiple items	9

COURSE OUTCOMES:

At the end of Course Students will be,

CO1: Use appropriate operators, control structures and looping statements to construct of Python programmes

CO2: Implement Python programs using Bulit in, user defined and recursive functions.

CO3: Apply methods to create and manipulate lists, tuples and sets.

CO4: Use appropriate methods to implement Dictionaries and Strings

CO5: Illustrate the concepts of file handling and Regular Expressions.

Text Books:

1. Al Sweigart, “Automate the Boring Stuff with Python “ , 1st edition, No Starch Press, 2015
2. Richard L. Holterman, ” Fundamentals of Python Programming”, 2017
3. Gowrishankar S, Veena A, “Introduction to Python Programming”, 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372

Reference Books:

1. Allen Downey, Jeffrey Elkner, Chris Meyers, “How to Think Like a Scientist –Learning with Python “, First Edition, Green Tea Press, 2002,
2. Charles Dierbach, ” Introduction to Computer Science Using Python”, Wiley Publication
3. Learning with Python “, Green Tea Press, 2002, First Edition.
4. Dave Kuhlman, ” A Python Book: Beginning Python, Advanced Python, and Python Exercises”.
5. Mark Lutz, “Learning Python”, 5th Edition”, O’Reilly
6. Micheal Urban, Joel Murach, “Murachs Python Programming”, 2016
7. Zed Shaw, ” Learn Python the Hard way”, 3rd Edition, Addison Wesley, 2013
8. Bill Lubanovic, ” Introducing Python- Modern Computing in Simple Packages”, O ‘Reilly Publications

E-Learning Sources:

1. 1. https://www.w3schools.com/python/
2. 2. https://www.geeksforgeeks.org/python-programming-language/

PYTHON PROGRAMMING LAB

1: Gain programming skills in Python to solve simple problems using looping structures, decision structures, functions, modules strings, lists, tuples, dictionaries, sets. 2: Implement concepts of objects, inheritance, exception handling, file handling, data visualization using Python programming.	
1	Write simple Python program to display message on screen
2	Write simple Python program using operators: a) Arithmetic Operators b) Logical Operators c) Bitwise Operators
3	Write Python program to perform following operations on Tuples: a) Create Set b) Access Set elements c) Update Set d) Delete Set
4	Develop user defined Python function for given problem: a) Function with minimum 2 arguments b) Function returning values
5	5. Write a program in Python to demonstrate following operations: a) Method overloading b) Method overriding
6	Write a program in Python to handle user defined exception for given problem.
7	Write a program to double a given number and add two numbers using lambda()?
8	Write a program for map() function to double all the items in the list?
9	Demonstrate a python code to implement abnormal termination?
10	Write a python program to write the content “hi python programming” for the existing file.
11	Write a python program to display a particular month of a year using calendar module.
12	Write a Regular Expression to represent all 10-digit mobile numbers. Rules: 1. Every number should contain exactly 10 digits. 2. The first digit should be 7 or 8 or 9 Write a Python Program to check whether the given number is valid mobile no. or not?
13	Write a Regular Expression to represent all RGM language (Your own language) identifiers. Rules: 1. The allowed characters are a-z, A-Z, 0-9, #. 2. The first character should be a lowercase alphabet symbol from a to k. 3. The second character should be a digit divisible by 3. 4. The length of identifier should be at least 2. Write a python program to check whether the given string is RGM language identifier or not?
14	Implement the following Searching and Sorting techniques in Python by using functions. i) Linear Search ii) Binary Search iii) Selection Sort iv) Bubble Sort v) Insertion vi) Merge Sort viii) Quick Sort
15	Write a Python program to return multiple values at a time using a return statement.