



**Course Title: PYTHON PROGRAMMING**

**Course Level: UG**

**Course Code: CSIT216**

**Credit Units: 04**

L	T	P/ S	SW/F W	TOTAL CREDIT UNITS
3	-	2	-	4

**Course Objectives:**

The aim of this course is to

- Provide in-depth knowledge of developing and debugging Python Programs.
- Illustrate and manipulate core data structures like Lists, Dictionaries, Tuples, and Strings.
- Understand the concept of files and exception handling.

**Pre-requisites:** Experience using a web browser and email.

**Course Contents/Syllabus:**

	Weightage (%)
<b>Module I: Introduction</b>	
Basic concepts: Functional Programming, OOPS and Data Structures Getting Started: Running Code in the Interactive Shell, Input, Processing and Output, Editing, Saving and Running a Script, Working of Python. Variables, Expressions and Statements: Values and Data Types, Variables, Keywords, String Literals, Escape Sequences, Operators and Operands, Expressions and Statements, Interactive mode and Script mode, Order of Operations, Comments	15
<b>Module II: Conditional Statements and Loops</b>	
Modulus Operator, Boolean Expressions, Logical Operators, Conditional Execution “if statement”, Alternative Execution “else clause”, Chained Conditionals “elif clause”, Nested Conditionals, While statement, For loop, Break and Continue Statement	15
<b>Module III: Functions and Recursion</b>	
Function Calls, Type Conversion Functions, Math Functions, Composition, Adding new functions, Parameters and Arguments, Stack Diagrams, Importing modules with “from”, Recursion, Stack Diagram for Recursive Functions, Infinite Recursion	25

String Functions: Traversal, Comparison, Searching, Counting, Pre-defined String Functions, In Operator	
<b>Module IV: Lists, Dictionaries and Tuples</b>	
Lists: List as a Sequence, Traversing a list, List Operations, List Slices, List Methods, Map, filter and Reduce, Deleting Elements, Lists and Strings, Objects and Values, Aliasing, List Arguments Dictionaries: Dictionary as a set of counters, Looping and Dictionaries, Reverse Look Up, Dictionaries and Lists, Memos, Global Variables, Long Integers Tuples: Tuple Assignment, Tuples as return values, Variable Length argument tuples, Lists and Tuples, Dictionaries and Tuples, Comparing Tuples, Sequences of sequences	<b>25</b>
<b>Module V: Files</b>	
Text files and their Formats, Reading from a file, Writing to a file, Accessing and Manipulating Files and Directories on the Disk, Format Operator, Filenames and paths Exception Handling: Errors, Exceptions, Handling Exceptions, Raising Exceptions, Try..Finally, The with Statement, Catching Exceptions, Databases, Pickling, Pipes	<b>20</b>

### Student Learning Outcomes:

After completion of this course, student will be able to:

- Solve basic Python programs.
- Demonstrate basic Python decisions and iterations.
- Define custom functions and call built-in Python functions.
- Explain data structures of Python.
- Identify exceptions and document code.

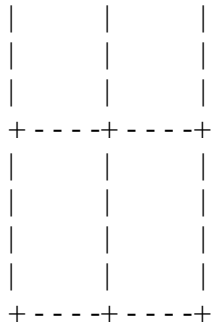
**Pedagogy for Course Delivery:** Subject will be taught based on class room lectures and practical.

### Lab/Practical details:

#### List of Experiments:

1. Start the Python interpreter and type help('print') to get information about the print statement.
2. If you run a 10 kilometer race in 43 minutes 30 seconds, calculate your average time per mile and your average speed in miles per hour using Python Calculator. (Hint: there are 1.61 kilometers in a mile).
3. Write a function to calculate the square of first n natural numbers.
4. Write a function that draws a grid like the following:

```
+ - - - + - - - +
|       |       |
```



5. Write a function that takes four parameters—a, b, c and n—and then checks to see if Fermat’s theorem,  $a^n + b^n = c^n$ , holds. If n is greater than 2 and it turns out to be true then the program should print, “Holy smokes, Fermat was wrong!” Otherwise the program should print, “No, that doesn’t work.”
6. Write a function that takes a string argument and returns true if it is a palindrome and False otherwise.
7. A number, a, is a power of b if it is divisible by b and a/b is a power of b. Write a function that takes parameters a and b and returns True if a is a power of b.
8. Write a recursive function to calculate the factorial of a given number.
9. Write a function that takes a string as a parameter. Calculate the length of a string without using len function. Print the length concatenated with the string and aligned towards the extreme right of the output screen.
10. ROT13 is a weak form of encryption that involves “rotating” each letter in a word by 13 places. To rotate a letter means to shift it through the alphabet, wrapping around to the beginning if necessary, so 'A' shifted by 3 is 'D' and 'Z' shifted by 1 is 'A'. Write a function that takes a string and an integer as parameters, and then returns a new string that contains the letters from the original string “rotated” by the given amount. Use the built-in functions ord, which converts a character to a numeric code, and chr, which converts numeric codes to characters.
11. Write a function that takes a nested list of integers and add up the elements from all of the nested lists.
12. Write a function called middle that takes a list and returns a new list that contains all but the first and last elements. So middle ([1, 2, 3, 4]) should return [2, 3].
13. Write a program to print the keys of the dictionary and their values in an alphabetical order.
14. Write a function that takes any number of arguments and returns their sum.
15. Write a program that reads words.txt and prints only the words with more than 20characters (not counting white space).

**Assessment/ Examination Scheme:**

Theory L/T (%)	Lab/Practical/Studio (%)	Total (%)
75	25	100

**Theory Assessment (L&T):**

Continuous Assessment/Internal Assessment					End Term Examination
Components (Drop down)	Attendance	Mid-Term	Project	Viva	EE
Weightage (%)	5	10	10	5	70

**Lab/ Practical/ Studio Assessment:**

Continuous Assessment/Internal Assessment						End Term Examination	
Components (Drop down)	Attendance	Mid-Term	Lab Record	Continuous Assessment	Viva	Practical	Viva
Weightage (%)	5	10	10	10	5	40	20

**Text:**

- A Byte of Python by Swaroop C H, published by ebsshelf Inc., 2013.
- Think Python: How to think like a Computer Scientist by Allen Downey, published by Green Tea Press, May 2013. ISBN: 0971677506

**References:**

- Fundamentals of Python by Kenneth A. Lambert, published by Cengage Learning, 2011. ISBN-13: 978-1111822705
- Exploring Python by Timothy A. Budd, published by McGraw Hill, (c) 2010. ISBN-13: 978-0073523378
- Python Essential Reference, 4th Edition by David M. Beazley, published by Addison-Wesley. ISBN-13: 978-0672329784