20MCA131	PROGRAMMING LAB	CATEGORY	L	T	P	CREDIT
		PRACTICAL	0	1	3	2

Preamble: This course introduces a basic step towards program writing and develops the logical ability and problem-solving skill using Python Programming Language. Students are able to do testing and debugging of code written in Python.

Prerequisite: None

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understands basics of Python Programming language including input/output functions, operators, basic and collection data types
CO 2	Implement decision making, looping constructs and functions
CO 3	Design modules and packages - built in and user defined packages
CO 4	Implement object-oriented programming and exception handling.
CO 5	Create files and form regular expressions for effective search operations on strings and files.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2	2	1	2							
CO 2	3	3	3	2	2							
CO 3	3	3	3	3	3						1	
CO 4	3	3	3	3	3						1	
CO 5	3	3	3	3	3						1	

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember (K1)			
Understand (K2)			
Apply (K3)	10	10	10
Analyse (K4)	10	10	10
Evaluate (K5)	10	10	10
Create(K6)	20	20	20

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	50	50	3 hours

Continuous Internal Evaluation Pattern:

Maximum Marks: 50	
Attendance	15%
Maintenance of daily lab record and GitHub management	20%
Regular class viva	15%
Timely completion of day to day tasks	20%
Tests/Evaluation	30%

End Semester Examination Pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE like PyCharm, PyDev…
2. Display future leap years from current year to a final year entered by user.
3. List comprehensions:
   (a) Generate positive list of numbers from a given list of integers
   (b) Square of N numbers
   (c) Form a list of vowels selected from a given word
   (d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)
4. Count the occurrences of each word in a line of text.
5. Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.
6. Store a list of first names. Count the occurrences of 'a' within the list
7. Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both
8. Get a string from an input string where all occurrences of first character replaced with '$', except first character.
   [eg: onion -> oni$n]
9. Create a string from given string where first and last characters exchanged. [eg: python -> nythop]
10. Accept the radius from user and find area of circle.
11. Find biggest of 3 numbers entered.
12. Accept a file name from user and print extension of that.
13. Create a list of colors from comma-separated color names entered by user. Display first and last colors.
14. Accept an integer n and compute n+nn+nnn.
15. Print out all colors from color-list1 not contained in color-list2.
16. Create a single string separated with space from two strings by swapping the character at position 1.
17. Sort dictionary in ascending and descending order.

18. Merge two dictionaries.
19. Find gcd of 2 numbers.
20. From a list of integers, create a list removing even numbers.

Course Outcome 2 (CO2)

1. Program to find the factorial of a number
2. Generate Fibonacci series of N terms
3. Find the sum of all items in a list
4. Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.
5. Display the given pyramid with step number accepted from user.

   Eg: N=4

   1

   2 4

   3 6 9

   4 8 12 16
6. Count the number of characters (character frequency) in a string.
7. Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’
8. Accept a list of words and return length of longest word.
9. Construct following pattern using nested loop

   *

   * *

   * * *

   * * * *

   * * * * *

   * * * *

   * * *

   * *

   *
10. Generate all factors of a number.

11. Write lambda functions to find area of square, rectangle and triangle.

Course Outcome 3(CO3):

1. Work with built-in packages
2. Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

Course Outcome 4 (CO4):

1. Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.
2. Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.
3. Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.
4. Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.
5. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Course Outcome 5 (CO5):

1. Write a Python program to read a file line by line and store it into a list.
2. Python program to copy odd lines of one file to other
3. Write a Python program to read each row from a given csv file and print a list of strings.
4. Write a Python program to read specific columns of a given CSV file and print the content of the columns.
5. Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

Syllabus:

Input, Output and Import Functions, Operators, Data Types, Decision Making & Loops, Functions, Modules and Packages, File Handling, Object Handling, Exception Handling, Regular Expressions

Reference Books

1. Wesley J. Chun, "***Core Python Applications Programming***", 3rd Edition , Pearson Education, 2016

2. Charles Dierbach, "***Introduction to Computer Science using Python***", Wiley, 2015

3. Jeeva Jose, "***Taming Python by Programming***", Khanna Publishers, New Delhi, 2018

4. Downey, A. et al., "***How to think like a Computer Scientist: Learning with Python***", John Wiley, 2015

Web References

1. https://archive.org/details/MIT6.00SCS11
2. https://www.coursera.org/course/pythonlearn
3. http://www.learnerstv.com/Free-Computer-Science-Video-lectures-ltv163-Page1.htm
4. https://www.coursera.org/learn/python-databases

Course Contents and Lab Schedule

Topic	No. of hours
1. Input, Output and Import Functions	3
2. Operators	5
3. Data Types	6
4. Decision Making & Loops	6
5. Functions	5
6. Modules and Packages	6
7. File Handling	5
8. Object Handling	5
9. Exception Handling	2
10. Regular Expressions	4