

Laboratory Manual

(Version 10.0)

for

Python Programming - Lab

(MCA-166)

MCA - II Semester

Compiled by:

Dr. Saumya Bansal

(Assistant Professor, BVICAM, New Delhi)



**Bharati Vidyapeeth's
Institute of Computer Applications
and Management (BVICAM)**

A-4, Paschim Vihar, Rohtak Road, New Delhi-63

Visit us at: www.bvicam.in

Index

List of Abbreviations

Declaration

1. Vision of the Department	1
2. Mission of the Department	2
3. Programme Educational Objectives (PEOs)	3
4. Programme Outcomes (POs)	4
5. Institutional Policy for Students' Conduct	5
6. Learning Outcomes of Laboratory Work	6
7. Course/Lab Outcomes (COs)	6
8. Mapping of COs with POs	7
9. Course/Lab Description	8
10. Grading Policy	9
11. Lesson Plan	11
12. Assignments	12
13. Advanced Problems	14

List of Abbreviations

BTL	Bloom's Taxonomy Level
CE	Communication Efficacy
CICP	Conduct Investigations of Complex Computing Problems
CK	Computational Knowledge
CO	Course Outcome
DAC	Departmental Advisory Committee
DDS	Design and Development of Solutions
I&E	Innovation and Entrepreneurship
I&T	Individual & Team Work
IQAC	Internal Quality Assurance Cell
LLL	Life-Long Learning
MTU	Modern Tool Usage
PA	Problem Analysis
PE	Professional Ethics
PEO	Programme Educational Objective
PMF	Project Management and Finance
PO	Programme Outcome
SEC	Societal and Environmental Concern

Declaration

Department : Department of Computer Science and Applications

Course, Year and the Semester to which Lab is offered : MCA - I Year, II Semester

Name of the Lab Course : Python Programming Lab

Course Code : MCA-166

Version No. : 1.0

Name of Course/Lab Teacher(s) : Dr. Saumya Bansal

Laboratory Manual Committee : 1. Dr. Anupam Baliyan, Member
2. Dr. Ritika Wason, Member
3. Mrs. Tanya Pathak Garg, Member
4. Mr. Uttam Singh Bist, Member
5. Prof. P. S. Grover, Margdarshak
6. Mr. Amit Sharma, Alumni & Industry Expert
7. Mr. Manish Kumar, Concerned Subject Teacher, Convener

Approved by : DAC

Approved by : IQAC

Signature
(Course Teacher)

Signature
(Head of Department)

Signature
(IQAC Coordinator)

1. Vision of the Department

To become a Centre of excellence in the field of Computer Science and Applications, to contribute effectively in the rapidly changing global economy directed towards national development ensuring prosperity for the mankind.

2. Mission of the Department

- M₁** To become a centre of excellence in the field of Computer Science and Applications and produce professionals as per global industry standards.
- M₂** To foster innovation, entrepreneurial skills, research capabilities and bring all-round development amongst budding professionals.
- M₃** To promote analytical and collaborative life-long learning skills, among students and faculty members involving all stakeholders.
- M₄** To inculcate strong ethical values and professional behaviour while giving equal emphasis to social commitment and nation building.

3. Programme Educational Objectives (PEOs)

The PEOs for the MCA programme are as follows:

- PEO₁** Exhibit professional competencies and knowledge for being a successful technocrat.
- PEO₂** Adopt creative and innovative practices to solve real-life complex problems.
- PEO₃** Be a lifelong learner and contribute effectively to the betterment of the society.
- PEO₄** Be effective and inspiring leader for fellow professionals and face the challenges of the rapidly changing multi-dimensional, contemporary world.

4. Programme Outcomes (POs)

PO₁ Computational Knowledge (CK)

Demonstrate competencies in fundamentals of computing, computing specialisation, mathematics, and domain knowledge suitable for the computing specialisation to the abstraction and conceptualisation of computing models from defined problems and requirements.

PO₂ Problem Analysis (PA)

Identify, formulate, and analyze complex real-life problems in order to arrive at computationally viable conclusions using fundamentals of mathematics, computer sciences, management and relevant domain disciplines.

PO₃ Design and Development of Solutions (DDS)

Design efficient solutions for complex, real-world problems to design systems, components or processes that meet the specifications with suitable consideration to public health, and safety, cultural, societal, and environmental considerations.

PO₄ Conduct Investigations of Complex Computing Problems (CICP)

Ability to research, analyze and investigate complex computing problems through design of experiments, analysis and interpretation of data, and synthesis of the information to arrive at valid conclusions.

PO₅ Modern Tool Usage (MTU)

Create, select, adapt and apply appropriate technologies and tools to a wide range of computational activities while understanding their limitations.

PO₆ Professional Ethics (PE)

Ability to perform professional practices in an ethical way, keeping in mind cyber regulations & laws, responsibilities, and norms of professional computing practices.

PO₇ Life-Long Learning (LLL)

Ability to engage in independent learning for continuous self-development as a computing professional.

PO₈ Project Management and Finance (PMF)

Ability to apply knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects in multidisciplinary environments.

PO₉ Communication Efficacy (CE)

Ability to effectively communicate with the technical community, and with society at large, about complex computing activities by being able to understand and write effective reports, design documentation, make effective presentations, with the capability of giving and taking clear instructions.

PO₁₀ Societal and Environmental Concern (SEC)

Ability to recognize and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities applicable to professional computing practices.

PO₁₁ Individual & Team Work (I&T)

Ability to work in multi-disciplinary team collaboration both as a member and leader as per need.

PO₁₂ Innovation and Entrepreneurship (I&E)

Ability to apply innovation to track a suitable opportunity to create value and wealth for the betterment of the individual and society at large.

5. Institutional Policy for Students' Conduct

The following guidelines shall be followed:-

- 5.1 All the students in their introductory Lab. shall be assigned a system, which shall be their workplace for the complete semester. Students can store records of all their Lab. assignments on their individual workstations.
- 5.2 Introductory Lab. shall include an introduction to the appropriate software/tool, followed by a basic Introductory Assignment having Practice Questions. All the students are expected to complete this assignment within a week time, as the same shall be assessed through a lab. test.
- 5.3 Each week the instructor, in parallel to respective topics covered in the theory lecture, shall assign a set of practical problems to the students in form of Assignments (A, B, C,). The problems in these assignments shall be divided into two parts. The first set of Problems shall be compulsory for all the students and its record need to be maintained in the Practical File, having prescribed format, as given in Appendix-A. All the students should get the weekly assignment checked and signed in the Practical File by the respective teacher in the immediate succeeding week. The second set of problems are Advanced Problems and shall be optional. Student may solve these advanced problems for their further practice.
- 5.4 Cellular phones, pagers, CD players, radios and similar devices are prohibited in the classrooms, laboratories and examination halls.
- 5.5 Laptop-size computers/Tablets may be used in lectures for the purpose of taking notes or working on team-projects.
- 5.6 The internal practical exam shall be conducted towards the end of the semester and shall include the complete set of Lab exercises conducted as syllabus. However, students shall be assessed on continuous basis through

overall performances in regular lab. tests, both announced and surprise and viva-voce.

- 5.7 The respective faculty shall prepare and submit sufficient number of practical sets of computing problems to the Dean (Examinations), atleast two weeks prior to the actual exam. It is the responsibility of the faculty to ensure that a set should not be repeated for more than 5 students in a given batch.
- 5.8 The exam shall be of 3 hours duration where the student shall be expected to implement solutions to his/her assigned set of problems on appropriate software tools in the lab.
- 5.9 Once implemented, student shall also appropriately document code implemented in the assigned answer sheets, which shall be submitted at the end of the examination. All the students shall also appear for viva-voce examination during the exam.
- 5.10 Co-operate, Collaborate and Explore for the best individual learning outcomes but copying or entering into the act of plagiarism is strictly prohibited.

6. Learning Outcomes of Laboratory Work

The student shall demonstrate the ability to:

- Verify and implement the concepts and theory learnt in class.
- Code and use Software Tools to solve problems and present their optimal solutions.
- Apply numerical/statistical formulas for solving problems/questions.
- Develop and apply critical thinking skills.
- Design and present Lab as well as project reports.

- ☑ Apply appropriate methods for the analysis of raw data.
- ☑ Perform logical troubleshooting as and when required.
- ☑ Work effectively as a member of a team in varying roles as need be.
- ☑ Communicate effectively, both oral and written.
- ☑ Cultivate ethics, social empathy, creativity and entrepreneurial mindset.

7. Course/Lab Outcomes (COs)

- CO1** Demonstrate program creation in Python through usage of appropriate constructs and OOPs concepts. (BTL2)
- CO2** Apply the concepts of data structures and string functions in python program. (BTL3)
- CO3** Apply the concepts of file handling and exception handling. (BTL3)
- CO4** Evaluate and visualize the data using appropriate python libraries. (BTL5)
- CO5** Develop GUI based applications with database connectivity in Python. (BTL6)

8. Mapping of CO's with PO's

Table 1: Mapping of CO's with PO's

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓		✓							
CO2	✓	✓	✓	✓	✓							
CO3	✓	✓	✓	✓	✓							
CO4	✓	✓	✓	✓	✓	✓				✓		
CO5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

9. Course/Lab Description

Course (Lab) Title	: Python Programming Lab
Course (Lab) Code	: MCA-166
Credits	: 01
Pre-requisites	: NA
Academic Session	: January to June
Contact Hours/Week	: 02 (01 Lab of 02 Hours/Week)
Internal Assessment	: 40 Marks
External Assessment	: 60 Marks

10. Grading Policy

Item	Points	Marks	Remarks
Programme Execution Test / Presentation / Group Discussion	20	20	Closed Book/Open Book
Practical File / Project Report	10	10	Closed Book
Viva-Voce	10	10	Closed Book
External End-Term Examinations	60	60	Closed Book (conducted and evaluated by the University)
Total		100	

11. Lesson Plan

Week No.	Lab No.	Topics/Concepts to be Covered
1.	1.	Basics of Python
2.	2.	Functions in Python
3.	3.	Concept of Recursion
4.	4.	Concept of tuple and list
5.	5.	Dictionary problems

Week No.	Lab No.	Topics/Concepts to be Covered
6.	6.	Concept of Strings
7.	7.	Use of in-built function
8.	8.	Blend of recursion and strings
9.	9.	Concept of files
10.	10.	Matplotlib library
11.	11.	Creating GUI using Python
12.	12.	Revision of all concepts

12. Lab Problems

Assignment Set:

Problems:

- P1* Implement Python Script to generate first N natural numbers.
- P2* By considering the terms in the Fibonacci sequence whose values do not exceed 1000, find the sum of the even-valued terms.
- P3* Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.
- P4* Write a function `cumulative_product` to compute cumulative product of a list of numbers.
- P5* Write a function `reverse` to reverse a list. Without using the `reverse` function.
- P6* Define a function which generates Fibonacci series up to n numbers using RECURSION.
- P7* With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.
- P8* Write a program to count the numbers of characters in the string and store them in a dictionary data structure.
- P9* Remove spaces from a string using recursion.
- P10* Write a program to compute the number of characters, words and lines in a file.
- P11* Write a Python class which has two methods `get_String` and `print_String`.

get_String accept a string from the user and print_String print the string in upper case.

- P12* Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.
- P13* Write a Python class to reverse a string word by word : Input string : 'hello Python' Expected Output : 'Python hello'
- P14* Write a function that finds the sum of first n terms of following series: $1 - x^2 / 2! + x^4 / 4! - x^6 / 6! + \dots \dots x^n / n!$
- P15* Write a program to compare two files and display total number of lines in a file.
- P16* Every time a vote is cast the name of the candidate is appended to the data structure. Print the names of candidates who received maximum vote in lexicographical order and if there is a tie print lexicographically smaller name.
- P17* Write a program to determine whether a given string has balanced parenthesis or not.
- P18* Implement a python script to check the element is in the list or not by using Linear search & Binary search.
- P19* Implement a python script to arrange the elements in sorted order using Bubble, Selection, Insertion and Merge sorting techniques.
- P20* Create a menu driven program to perform various matrices operations.
- P21* Draw different graphs and plots in Python using Matplotlib Library.
- P22* Write a python code for simple GUI calculator using Tk
- P23* Write a python program to perform various database operations (create, insert, delete, update)

13. Advanced Problems

Assignment Set:

Problems:

- A1** Write a function that takes a number as an input parameter and returns the corresponding number name in words. For example: if input is 452, the function should return "Four Five Two". Use a dictionary for mapping digits to their string representation.
- A2** Given a list of integers with duplicate elements in it. Create a new list that contains the elements which appear more than one.
- A3** Write a function that reads a file and copies only alternative lines to another file. Alternative lines copied should be the odd numbers lines. Handle all exceptions that can be raised.
- A4** Write a Python program that accepts a space separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.
- A5** Write a python program to merge two list into one sorted in ascending order. Eg
L1= [1,5,9] L2 = [2,4,6], Output = [1,2,4,5,6,9]

Appendix - A: Index of Lab File

Week No.	Lab. Ex. No.	Detailed Description of the Lab Exercise	Outcome Mapping		Page No./Link of Online Document	Signature of Teacher with Date
			CO	BTL		
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						