Lingaya's Vidyapeeth

Deemed-to-be-University u/s 3 of UGC Act 1956, Government of India
NAAC ACCREDITED
Approved by MHRD / AICTE / PCI / BCI / COA / NCTE
Nachauli, Jasana Road, Faridabad- 121002 (Haryana)
Website: www.lingayasvidyapeeth.edu.in | Ph: 0129-2598200-05

COURSE PLAN & COURSE DATA SHEET

PROGRAM: BCA/BSc(CS)	DEGREE:
COURSE: Python Programming	SEMESTER: VI CREDITS: 3
COURSE CODE: BCA-308/BCS-308 REGULATION:	COURSE TYPE: CORE
COURSE AREA/DOMAIN:	CONTACT HOURS: 3 hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY): BCA-358/BCS-358	LAB COURSE NAME (IF ANY): Python Programming Lab

PROGRAM EDUCATIONAL OBJECTIVES:

SYLLABUS:

UNIT	DETAILS	HOURS
I	**Introduction to Python:** History, features & benefits of Python, Structure of a Python Program, Identifiers and keywords, concept of variable, memory allocation for variable, data types in python, conversion functions, Operators (Arithmetic Operator, Relational, Logical or Boolean operator, Assignment, Operator, ,Bit wise operator, membership operator).Input and output function, Control Statements (Looping- while Loop, for Loop, Loop Control, Conditional Statement-if...else, Difference between break, continue and pass).	8
II	**Data Structures & Function:** String, lists tuples, sets, Dictionary data structure, built-in library function, method and operation on these data structure .Python Date & Time, Defining Function, type of function arguments (Required arguments, keyword arguments, default arguments, Variable-length arguments), pass by reference Vs pass by value, Concept of recursion, Lambda functions, scope of a variable, global Vs local variable, Python modules & packages, import statement, dir(), globals(), locals() and reload() functions, collections.	10
III	**Python Object Oriented Programming:** Introduction to object oriented programming, Concept of abstraction, encapsulation, class, object and instances. Creating classes, _init_() method, creating instance object, class attributes, UML class diagrams, access specifiers in python, instance method Vs class method Vs static method,. Inheritance & polymorphism, overriding and overloading methods, overloading operators, relationships, Programming using Oops support.	9
IV	**Python File handling, exception handling & concurrency:** opening & closing files ,file access modes, file object attributes, reading and writing files, Manipulating file pointer using seek and tell. Programming using file operations. Exception handling in python	6

V	**Python Database Integration, Data visualization, Statistical data analysis:** MySQL/Oracle Database connection using python. Creating database tables, SELECT, INSERT, UPDATE, and DELETE operation, performing commit, rollback operation, error handling Programming using database connections, Python data visualization, statistical data analysis using python.	7
	TOTAL HOURS	40

Teacher Centric Approach **TC1: Chalk and Talk, TC2: PPT, TC3: Video Lectures TC4: Blended learning**
Learner Centric Approach: **LC1: Assignment. LC2: Mini project. LC3: Quiz/Class test. LC 4: Seminar on recent trends.** **LC5: Group Task. LC6: Others**

DETAILED SESSION PLAN

Lecture session/ Number	Topics to be covered	CO addressed	Teacher Centric Approach	Learner Centric Approach	References	Relevance with POs and PSOs
1	**UNIT-1 Introduction to Python:** History, features & benefits of Python, Structure of a Python Program	CO1	TC1, TC2	LC2	T1/W2	
2	Identifiers and keywords, Concept of variable, Memory allocation for variable, data types in python	CO1	TC1, TC2	LC2	T1/W2	
3	conversion functions, Operators	CO1	TC1, TC2	LC2	T1/W2	
4	Operators, Input and output function	CO1	TC1, TC2	LC2	T1/W2	
5	Conditional Statement- if...else	CO1	TC1, TC2	LC2	T1/W2	
6	Difference between break, continue and pass	CO1	TC1, TC2	LC2	T1/W2	
7	Control Statements (Looping- while Loop,)	CO1	TC1, TC2	LC2	T1/W2	

8	Doubt Class			LC2		
9	UNIT-2 **Data Structures & Function:** String, lists data structure, built-in method and operation on string and list data structure	CO2	TC1, TC2	LC2	T1/W2	
10	Tuples, sets data structure, built-in method and operation on tuple data structure	CO2	TC1, TC2	LC2	T1/W2	
11	Dictionary data structure, built-in method and operation on dictionary data structure	CO2	TC1, TC2	LC2	T1/W2	
12	Python Date & Time	CO2	TC1, TC2	LC2	T1/W2	
13	Defining Function, type of function arguments (Required arguments, keyword arguments, default arguments, Variable-length arguments)	CO3	TC1, TC2	LC2	T1/W2	
14	Function argument: pass by reference Vs pass by value	CO3	TC1, TC2	LC2	T1/W2	
15	Concept of recursion, Lambda functions,	CO3	TC1, TC2	LC2	T1/W2	

	scope of a variable, global Vs local variable					
16	Python modules & packages	CO3	TC1, TC2	LC2	T1/W2	
17	import statement, dir(), globals(), locals() and reload() functions, collections.	CO3	TC1, TC2	LC2	T1/W2	
18	Doubt Class			LC2		
19	Unit-3 **Python Object Oriented Programming:** Basic Concepts	CO4	TC1, TC2	LC2	T1/W2	
20	Introduction to object oriented programming, Concept of abstraction, encapsulation, class, object and instances	CO4	TC1, TC2	LC2	T1/W2	
21	Creating classes, creating instance object, class attributes	CO4	TC1, TC2	LC2	T1/W2	
22	_init_() method	CO4	TC1, TC2	LC2	T1/W2	
23	access specifiers in python, instance method Vs class method Vs static method	CO4	TC1, TC2	LC2	T1/W2	

24	Inheritance and its type & overloading operators, relationships, Programming using Oops support.	CO4	TC1, TC2	LC2	T1/W2	
25	Polymorphism, overriding methods	CO4	TC1, TC2	LC2	T1/W2	
26	Polymorphism, overloading methods	CO4	TC1, TC2	LC2	T1/W2	
27	Doubt Class			LC2		
28	Unit-4 **Python File handling, exception handling & concurrency:** opening & closing files ,file access modes	CO5	TC1, TC2	LC2	T1/W2	
29	file object attributes, reading and writing files, Manipulating file pointer using seek and tell.	CO5	TC1, TC2	LC2	T1/W2	
30	Programming using file operations	CO5	TC1, TC2	LC2	T1/W2	

31	Exception handling in python	CO5	TC1, TC2	LC2	T1/W2	
32	Exception handling in python	CO5	TC1, TC2	LC2	T1/W2	
33	Doubt class			LC2		
34	Unit-5 **Python Database Integration, Data visualization, Statistical data analysis:** MySQL/Oracle Database connection using python. Creating database tables	CO5	TC1, TC2	LC2	T1/W2	
35	SELECT, INSERT, UPDATE, and DELETE operation	CO5	TC1, TC2	LC2	T1/W2	

36	performing commit, rollback operation,	CO5	TC1, TC2	LC2	T1/W2	
37	error handling Programming using database connections	CO5	TC1, TC2	LC2	T1/W2	
38	Python data visualization	CO5	TC1, TC2	LC2	T1/W2	
39	Statistical data analysis using python.	CO5	TC1, TC2	LC2	T1/W2	
40	Revision class					

TEXT/REFERENCE BOOKS/ WEB LINK:

T/ R/W	
T1	John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
R1	T. Budd, Exploring Python, TMH, 1st Ed, 2011
R2	Allen Downey, Jeffrey Elkner, Chris Meyers ,How to think like a computer scientist :Learning with Python,Freelyavailableonline.2012
W1	Python Tutorial (tutorialspoint.com)
W2	NPTEL: Programming in Python

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
BCA-101	**Computer Programming**	**Concepts of C language**	**I**

COURSE OBJECTIVES:

1	To understand why Python is a useful scripting language for developers.
2	To learn how to design and program Python applications
3	To learn how to use lists, tuples, and dictionaries in Python programs.
4	To learn how to identify Python object types.
5	To learn how to use indexing and slicing to access data in Python programs.
6	To define the structure and components of a Python program.
7	To learn how to write loops and decision statements in Python.
8	To learn how to write functions and pass arguments in Python.
9	To learn how to build and package Python modules for reusability.
10	To learn how to read and write files in Python.
11	To learn how to design object-oriented programs with Python classes.

12	To learn how to use class inheritance in Python for reusability.
13	To learn how to use exception handling in Python applications for error handling.

COURSE OUTCOMES:

S.NO	DESCRIPTION	PO(1..12) MAPPING	PSO(1..3) MAPPING
CO 1	Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements.	PO1,PO2,PO3,PO5,PO6	PSO1
CO 2	Determine the methods to create and manipulate Python programs by utilizing the data structures like lists, dictionaries, tuples and sets	PO1,PO2,PO3,PO5,PO6	POS1,PSO2
CO 3	Express proficiency in the handling of strings and functions	PO1,PO2,PO5,PO6,PO8,PO12	PSO1,PSO2,PSO3
CO 4	Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism as used in Python	PO1,PO2,PO4,PO5,PO6,PO9,PO12	PSO1,PSO3
CO 5	Identify the commonly used operations involving file systems and regular expressions., Connectivity with database	PO1,PO2,PO4,PO5,PO6,PO8,PO9,PO12	PSO1,PSO3
COURSE OVERALL PO/PSO MAPPING:			

COURSE OUTCOMES VS POs MAPPING (DETAILED; HIGH: 3; MEDIUM:2; LOW:1**):**

S.NO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1	1		1	1							3		
CO.2	2	1	1		3	1							1	2	
CO.3	1	1			1	1		1				1	1	1	1
CO.4	1	1		1	1	1			1			1	1		1
CO.5	1	1		1	1	1		1	1			1	1		1

** For Entire Course, PO & PSO Mapping*

POs & PSO REFERENCE:

PO 1	Engineering Knowledge	PO7	Environment & Sustainability	PSO1	To equip the students with theoretical and implementation knowledgebase in all the latest areas of Computer Science & Engineering for a

					successful career in software industries, pursuing higher studies, or entrepreneurial establishments.
PO 2	Problem Analysis	PO8	Ethics	PSO2	To nurture the students with the critical thinking abilities for better decision making by offering them a socially acceptable solutions to real life problems through computing paradigm.
PO 3	Design & Development	PO9	Individual & Team Work	PSO3	To nurture the students with the comprehensive analytical and design abilities by offering them techno-commercially feasible solutions of real business problems through computing.
PO 4	Investigations	PO10	Communication Skills		
PO 5	Modern Tools	PO11	Project Mgt. & Finance		
PO 6	Engineer & Society	PO12	Life Long Learning		

COs VS POs MAPPING JUSTIFICATION:

S.NO	PO/PSO MAPPED	LEVEL OF MAPPING	JUSTIFICATION
Cxxx.1			
Cxxx.2			
Cxxx.3			
Cxxx.4			
Cxxx.5			
Cxxx*			

GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSION REQUIREMENTS, POs & PSOs:

SNO	DESCRIPTION	PROPOSED ACTIONS
1		
2		
3		
4		
5		

PROPOSED ACTIONS: TOPICS BEYOND SYLLABUS/ASSIGNMENT/INDUSTRY VISIT/GUEST LECTURER/NPTEL ETC

 # TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	Read CSV file
2	How to fill Null entries in CSV file

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

☐ CHALK & TALK	☐ STUD. ASSIGNMENT	☐ WEB RESOURCES	☐ NPTEL/OTHERS
☐ LCD/SMART BOARDS	☐ STUD. SEMINARS	☐ ADD-ON COURSES	☐ WEBNIARS

ASSESSMENT METHODOLOGIES-DIRECT

☐ ASSIGNMENTS	☐ STUD. SEMINARS	☐ TESTS/MODEL EXAMS	☐ UNIV. EXAMINATION

☐ STUD. LAB PRACTICES	☐ STUD. VIVA	☐ MINI/MAJOR PROJECTS	☐ CERTIFICATIONS
☐ ADD-ON COURSES	☐ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

☐ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	☐ STUDENT FEEDBACK ON FACULTY (TWICE)
☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	☐ OTHERS

 # INNOVATIONS IN TEACHING/LEARNING/EVALUATION PROCESSES:

1. **Technology Integration:** Embrace and integrate technology tools in the classroom to enhance the learning experience. This can include interactive whiteboards, educational apps, virtual reality, and online collaboration platforms. Utilizing technology allows for more dynamic and interactive lessons, catering to diverse learning styles.
2. **Personalized Learning Paths:** Implement personalized learning approaches that cater to individual student needs and pace of learning. Adaptive learning platforms and data analytics can help tailor educational content, assignments, and assessments based on the strengths and weaknesses of each student, promoting a more customized learning experience.
3. **Active Learning Strategies:** Move away from traditional lecture-based approaches and incorporate active learning strategies. This involves engaging students in hands-on activities, group discussions, problem-solving exercises, and real-world projects. Active learning fosters critical thinking, collaboration, and practical application of knowledge.
4. **Blended Learning Models:** Adopt blended learning models that combine face-to-face instruction with online resources. This allows for flexibility in learning, enabling students to access materials at their own pace outside the classroom. Flipped classrooms, where students learn new concepts online and engage in discussions and activities during class, are an example of a blended learning approach.
5. **Assessment Innovation:** Rethink assessment methods to go beyond traditional exams and quizzes. Explore alternative forms of assessment, such as project-based assessments, portfolios, presentations, and peer assessments. Additionally, incorporate formative assessments and feedback throughout the learning process to help students track their progress and make improvements.

Prepared by
(Ms. Komal Malsa)

Approved by
(HOD)

 # Additionally, the details to be compiled separately by the Departmental Coordinator for the entire Department.