Government of Karnataka

DEPARTMENT OF COLLEGIATE AND TECHNICAL EDUCATION

Programme	Computer Science and Engineering	**Semester**	III
Course Code	20CS31P	**Type of Course**	Programme Core
Course Name	**Python Programming**	**Contact Hours**	8 hours/week 104 hours/semester
Teaching Scheme	L:T:P :: 3:1:4	**Credits**	6
CIE Marks	60	**SEE Marks**	40

1.Rationale

Computer programming is the core of the computer science and strong fundamentals of programming can give competitive edge in this technology driven world. It not only instils coding skills but also enhances problem solving ability. Python is one of the programming languages which is versatile and feature rich yet simple and easy to learn, has applications in various domains. Python programming sets the basis for further study of web development, data science, IoT, machine learning etc.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Install the latest version of python distribution and configure it for an appropriate workspace as needed for a given project.
CO-02	Write a program by selecting python constructs needed to solve a given problem and then code, execute, test and debug the program to obtain the desired result.
CO-03	Demonstrate how a program can be optimized by using modular programming approach.
CO-04	Identify and resolve both syntactical and semantic errors in a given code snippet.

3. Course Content

Week	**CO**	**PO**	**Lecture (Knowledge Criteria)**	**Tutorial (Activity Criteria)**	**Practice (Performance Criteria)**
			3 hours/week	1 hour/week	4 hours/week (2 hours/batch twice in a week)
1	1,2	1,4	**Fundamental Concepts:** brief history; features; applications of python; python distributions; versions; python IDEs; Python interpreter; Execution of python programs, debugging python code; Indentation, Comments; best practices for python programming; Character set; tokens; keywords, variables, naming rules for variables, Assignment,	**Refer Table 1**	1. Setup python environment 2. Executing python: explore different ways to run python program 3. debug python code
2	2,4	1,2,4	**Basics I/O operations** Input- input (), raw_input() ; output – print (), formatting output. **Datatypes**		1. Code, execute and debug programs that a) Use i/o statements

			Scalar type: Numeric (int, long, float, complex), Boolean, bytes, None; Type casting **Operators** Arithmetic, Comparison/Relational, Logical/Boolean, Bitwise; string operators; Expressions and operator precedence		b) Evaluate expressions and displays formatted output c) Evaluate expressions to examine the operator precedence 2. Identify and resolve syntactic and semantic issues in the given code snippet
3	2,4	1,2,4	**Control Flow: Conditional blocks** If statement: general format; Multiway branching; Sufficient examples;		1. Identify and Code, execute and debug programs using conditional statements. 2. Identify and resolve syntactic and semantic issues in the given code snippet
4	2,4	1,2,4	**Control Flow: Loops** While loop: general format; examples For loop: general format, examples. Range();nesting loops and conditional statements; Controlling loop execution: Break, continue, pass statements;		1. Code, execute and debug programs using loops. 2. Code, execute and debug programs using loops and conditional statements 3. Identify and resolve syntactic and semantic issues in the given code snippet
5	2,4	1,2,4	**Data Collections** Concept of mutability Set – features, declaration, initialization, operations, comprehension; Tuple-features; declaration, initialization, basic operations; indexing; slicing; built in functions; Nested tuples;		1. Code, execute and debug programs to perform following ▪ set operations ▪ set comprehension 2. Code, execute and debug programs to perform following ▪ basic operations on tuples ▪ tuple indexing and slicing 3. Identify and resolve syntactic and semantic issues in the given code snippet
6	2,4	1,2,4	**List** features; declaration, initialization, basic operations; indexing; List iterations; Slicing; built in functions; Nested Lists; Comprehensions; Applications		1. Write code snippet to perform following on List ▪ basic operations on List ▪ indexing and slicing ▪ comprehension

					2. Identify and resolve syntactic and semantic issues in the given code snippet
7	2,4	1,2,4	**Dictionary** features; declaration, initialization, basic operations; indexing; adding and removing keys, iterating through dictionaries; built in functions; Comprehensions; Applications		1. Code, execute and debug programs to perform basic operations on Dictionary 2. Code, execute and debug programs to perform Dictionary indexing Iterating comprehension 3. Identify and resolve syntactic and semantic issues in the given code snippet
8	2,4	1,2,4	**Arrays and Strings** Arrays: features; create, initialize, indexing, traversal, manipulation; Strings: create, assign, indexing, built in functions;		1. Code, execute and debug programs to perform string manipulation 2. Code, execute and debug programs to perform array manipulation 3. Identify and resolve syntactic and semantic issues in the given code snippet
9	2,3,4	1,2,4	**Functions** Need of function; types; define function, calling function, function arguments; return and yield; None keyword; Scope of variables; Recursion; anonymous functions; sufficient examples;		1. Code, execute and debug programs to solve the given problem using built in functions 2. Code, execute and debug programs to solve the given problem by defining a function 3. Code, execute and debug programs to solve the given problem using recursion 4. Define anonymous function and code to solve the given problem 5. Identify and resolve syntactic and semantic issues in the given code snippet
10	2,3,4	1,2,4	**Modules and Packages**		1. Create Modules and Packages

			Why modules? Module creation; Importing modules; Module Namespace;
			Packages: basics; path setting; Package __init__.py Files;
			Commonly used modules: Math, random; Emoji;
					2. Code, execute and debug programs using built in modules

11	2,3,4	1,2,4	**NumPy** Brief about NumPy module; NumPy arithmetic functions; NumPy array manipulation functions; NumPy statistical functions; **Pandas** Introduction, series, data frame; Create dataframes; formatting data; fundamental data frame operations;		1. Code, execute and debug programs using NumPy module. 2. Code, execute and debug programs using series. 3. Code, execute and debug programs using dataframes. 4. Identify and resolve syntactic and semantic issues in the given code snippet

12	2,3,4	1,2,4	**Files**
			Concept; features; file operations;
			Opening Files; Closing Files; Writing to Files;
			Reading to Files; File methods;
			Working with files using data frame.
					1. write code snippet to perform following operations on different types of files
					▪ read file
					▪ write to file.
					2. Write code to perform file operations using dataframes on different file types.
					3. Identify and resolve syntactic and semantic issues in the given code snippet

13	2,3,4	1,2,4	**Error and Exception Handling:** Python errors; exceptions: built in, user defined. How to catch exceptions? Raising exceptions;		1. Integrate exception handling into above code 2. Write code snippet to raise exceptions 3. Identify and resolve syntactic and semantic issues in the given code snippet
Total in hours			**39**	**13**	**52**

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl no	Activity
1	1. Compare and contrast excel and python 2. Identify various python IDEs and identify differences between them.
2	1. Identify use cases like reading student name and contact details and display in a required format 2. Compare and contrast input () and raw_input() and identify its appropriate use.

	3. Identify use of operators and develop algorithms to solve the same 4. Compare and contrast different types of operators
3	1. Identify use cases that involve decision making and develop algorithms to solve the same 1. Identify common syntactical errors when using control flow statements
4	1. Identify use cases that involve iteration and develop algorithms to solve the same 2. Compare and contrast different types of loops 2. Identify common syntactical errors when using loops
5	1. Identify use cases and solve them using sets 2. Identify use cases and solve them using tuples 3. Identify common syntactical errors when working with sets and tuples
6	1. Identify use cases and solve them using List 2. Identify common syntactical errors when working with List 3. Reimplement built in list functions
7	1. Identify use cases and solve them using dictionary 2. Reimplement built in dictionary functions 3. Identify common syntactical errors when working with dictionary
8	1. Identify use cases and solve them using arrays 2. Reimplement built in string functions 3. Identify common syntactical errors when working with arrays and strings
9	1. Optimize previously written programs by using modular programming approach
10	1. Identify and present pros and cons of modules and packages 1. Explore and present python built in modules.
11	1. Identify the applications of Pandas 2. Perform data analysis using Pandas module on a dataset such as .
12	1. Identify use cases on files concept and develop algorithms to solve the same 2. Explore regular expressions and present how they can be used for file manipulation
13	3. Compare and contrast error and exception. 4. Rewrite the programs using exceptions if needed

4. CIE and SEE Assessment Methodologies

Sl. No	**Assessment**	**Test Week**	**Duration** In minutes	**Max marks**	**Conversion**
1.	CIE-1 Written Test	5	80	30	Average of three tests 30
2.	CIE-2 Written Test	9	80	30	
3	CIE-3 Written Test	13	80	30	
4.	CIE-4 Skill Test-Practice	6	180	100	Average of two skill tests reduced to 20
5	CIE-5 Skill Test-Practice	12	180	100	
6	CIE-6 Portfolio continuous evaluation of Activity through Rubrics	1-13		10	10
				Total CIE Marks	60
	Semester End Examination (Practice)		180	100	40
				Total Marks	**100**

5. Format for CIE written Test

Course Name	**Python Programming**		Test	I/II/III	Sem	III/IV
Course Code	**20CS31P**		Duration	80 Min	Marks	30
Note: Answer any one full question from each section. Each full question carries 10 marks.						
Section	Assessment Questions		Cognitive Levels	Course Outcome	Marks	
I	1					

	2				
II	3				
	4				
III	5				
	6				
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, Cognitive level and course outcomes.					

6. Rubrics for Assessment of Activity (Qualitative Assessment)

Sl. No.	Dimension	Beginner	Intermediate	Good	Advanced	Expert	Students Score
		2	4	6	8	10	
1		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	8
2		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	6
3		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	2
4		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	2
	Average Marks= (8+6+2+2)/4=4.5						**5**

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

7. Reference:

Sl. No.	Description
1	Core python programming, Wesley J. Chun Publisher: Prentice Hall PTR
2	Fluent Python by Luciano Ramalho
3	https://www.softcover.io/read/e4cd0fd9/conversational-python
4	https://realpython.com/
5	https://www.python-course.eu/
6	https://www.datacamp.com/
7	https://www.w3schools.com/

8. CIE Skill Test and SEE Scheme of Evaluation

SL. No.	Particulars/Dimension	Marks
1	Develop an algorithmic solution for the given problem statement based on the documentation of each of the steps involved, including input, output and logic.	20
2	Write program for the above given problem choosing relevant python constructs.	20
3	Code, execute, test and debug the above program.	30
4	Demonstrate how your program has solved the given problem In the event of, a student fails to get the desired result (with no syntactical and least semantic errors), the examiner shall use viva voce to assess the student's problem-solving and python programming skills	20
5	Portfolio evaluation based on aggregate of all practice sessions	10
	Total Marks	**100**

9. Equipment/software list with Specification for a batch of 20 students

Sl. No.	Particulars	Specification	Quantity
1	Python 3.8		20
2	Editor such as iPython, Jupyter, spider, PyCharm , google CoLab		20
3	Computers		20