

19IT211 PYTHON PROGRAMMING

Hours Per Week :

L	T	P	C
2	-	4	4

Total Hours :

L	T	P
30	-	60

WA/RA	SSH/HSH	CS	SA	S	BS
5	40	5	20	5	2

PREREQUISITE COURSES: Programming for Problem Solving - I, II; Data Structures

COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed at offering the fundamental concepts of Python scripting language to the students. It starts with the basics of Python programming and deals with lists, dictionaries, functions, exceptions and files. The objective of this course is to enable the students to develop the applications using the concepts of Python.

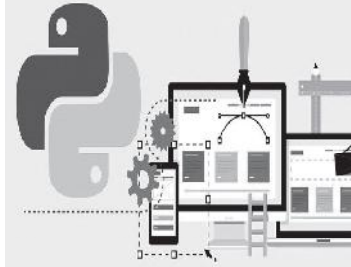
COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand different python programming concepts and apply them to develop programs.	1
2	Analyze the usage of different data structures for practical and contemporary applications for solving a given problem.	1,2
3	Develop functional, reliable and user friendly Python programs for a given problem application.	3
4	Design solutions using the concepts of object oriented programming paradigm.	2
5	Develop programs to solve data stored in files.	4

SKILLS:

- ✓ Identify suitable data types and data structures required for an application
- ✓ Design structured and Object oriented programming solutions
- ✓ Design reliable applications for a given problem



SOURCE:

<https://blog.internshala.com/2018/02/learn-python-programming-gamers-and-coders-of-the-world-unite/>

UNIT - I**L- 7**

INTRODUCTION: History of Python, Features of Python, Python Installation on Windows & LINUX, Installing python packages via PIP, Running python commands using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation; Basic data types - integers, booleans etc.

OPERATORS AND EXPRESSIONS: Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators; Expressions and order of evaluations.

UNIT - II**L- 7**

CONTROL STRUCTURES-CONDITIONAL CONTROL STRUCTURES: if, elif, else; Loop control structures: for, while, for... else, while..else, nested lo ops, break, continue, pass.

Python Data Structures: Lists, Tuples, Dictionary: Creation, Accessing, Basic operators and methods.

UNIT - III**L-6**

OTHER DATA STRUCTURES: Strings- creation, accessing, operators, methods; Sets- creation, accessing, operators, methods; List Comprehensions.

Functions - Defining functions, Calling functions, Passing arguments, Keyword arguments, Default arguments, Variable-length arguments, Anonymous functions (lambda), fruitful Functions (Function Returning Values), Scope of the variables in a function - global and local Variables.

UNIT - IV**L- 5**

MODULES: Creating modules, import statement, from. Import statement, name spacing.

Error and Exceptions Difference between an error and Exception, Handling Exception, Try except block, Raising Exceptions, User defined exceptions.

File processing: Reading and Writing Files- Creating a New File- Writing to a File- Reading Files as Text, Opening and Closing files, Reading and writing, tell (), seek(), rename ().

UNIT - V**L- 5**

OBJECT ORIENTED PROGRAMMING IN PYTHON: Classes, 'self variable' Methods, Constructor Method, Inheritance, Overriding Methods, Datahiding.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS

TOTAL HOURS: 60

1. a) Installation of python and relevant packages in windows.
 b) Installation of python and relevant packages in Linux.
 c) Practice Execution of python statements in REPL(shell).
2. a) Implement a python program to display all the python keywords and display each of them in separate lines.
 b) Develop a python program to read two integers and perform all possible arithmetic operations on those two numbers.
3. a) Develop a program to accept three numbers as command line arguments and find biggest, smallest and average of those three numbers.
 b) Implement a python program to find first n Prime Numbers.
 c) Implement a program that prints the decimal equivalents of $1/2$, $1/3$, $1/4$, . . . , $1/n$.
 d) Implement a python program to read n and find sum of even and odd numbers.
4. Write python code to achieve the following
 - a) to remove vowels in the given string using control transfer statements.
 - b) to count number of uppercase and lowercase letters in the given string.
 - c) to remove all punctuation characters from given string.
5. Implement python code to illustrate the following on Lists and Tuples
 - a) Creation b) Accessing elements c) apply operators d) Usage of different methods
6. Implement python code to illustrate the following on Sets and Dictionary
 - a) Creation b) Accessing elements c) apply operators d) Usage of different methods
7. a) Implement python code to illustrate the following
 - i) Positional arguments ii) Keyword arguments
 - iii) Default arguments iv) Variable length arguments
 b) Implement a function to find n^{th} Fibonacci number.
 c) Develop a recursive function to find the factorial of a given number.
 d) Implement function to compute GCD, LCM of two numbers (use Lambda function).
8. a) Develop a python code to handle the following built-in exceptions
 - i) ValueError ii) ZeroDivisionError iii) TypeError iv) NameError
 b) Implement python code to handle multiple exceptions.
 c) Implement Python code to raise an exception.
9. a) Implement python code to read contents of a file and write the contents to another file.
 b) Create a class called Student and perform operations such as display, Calculate percentage, add, delete and modify student data.
 c) Design python code to depict the following oops concepts:
 - i) Datahiding ii) Inheritance iii) Overriding

10. Develop python code to calculate the following statistical parameters using python 'numpy'.
- a) Mean b) Harmonic Mean c) Meadian d) Mode
- e) Standard Deviation f) Variance g) Percentile
11. Design python code to illustrate the following plots using 'matplotlib' package
- a) Line plot b) Bar plot c) Histogram d) Scatter Plot
12. Implement python program for the following problems on Pandas DataFrame
- a) Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
                    'Matthew', 'Laura', 'Kevin', 'Jonas'],
            'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
            'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
            'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

- b) Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.
- c) Write a Pandas program to select the specified columns and rows from a given data frame.
- d) Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2.
- e) Write a Pandas program to count the number of rows and columns of a DataFrame.
- f) Write a Pandas program to change the name 'James' to 'Adhvik' in name column of the DataFrame.

TEXT BOOKS:

1. Vamsi Kurama, "Python Programming: A Modern Approach", 1st edition, Pearson Publishers, 2018.
2. Mark Lutz, "Learning Python", 2nd edition, O'Reilly Media, 2003.

REFERENCE BOOKS:

1. Allen Downey, "Think Python", 2nd edition, Green Tea Press, 2016.
2. Ashok Namdev Kamthane, Amith Ashok Kamthane, " Programming and Problem Solving With Python", 1st Edition, McGraw Hill Education, 2016.
3. W.J. Chun, "Core Python Programming" , 3rd Edition, Pearson publishers, 2013.