

Name of Institute: Indus Institute of Technology & Engineering

Name of Faculty: Mr. Darshan Solanki

Course code: CE0525

Course name: Programming for Scientific Computing (Python)

Pre-requisites: - Knowledge of Object Oriented language will be useful.

Credit points: 4

Offered Semester: V

Course coordinator

Full name: Mr. Darshan Solanki

Department with sitting location: CSE dept, 4th floor Bhanwar Building.

Telephone: -

Email: darshansolanki.ce@indusuni.ac.in

Consultation times:

Monday to Friday - 03:00 pm to 05:00 pm & Working Saturday

Course lecturer

Full name: Ms. Zalak Trivedi, Ms. Madhvi Bera, Hinal Shah

Department with sitting location: CSE and EL dept, 4th floor and 2nd floor Bhanwar Building.

Telephone: -

Email: madhvibera.ce@indusuni.ac.in, zalaktrivedi.ce@indusuni.ac.in, hinalshah.el@indusuni.ac.in

Consultation times:

Monday to Friday - 03:00 pm to 05:00 pm & Working Saturday

Students will be contacted throughout the session via mail with important information relating to this course.

Course Objectives:

1. Importance of Python as a scientific computing tool which directly leads to employ-ability .
2. To learn how to design and develop Python applications.
3. Skill development to apply mutable and immutable types.
4. To learn how to design object-oriented concepts in python.
5. Development of GUI based applications for entrepreneurship.
6. To learn how to build and package Python modules for re-usability.

Course Outcomes (CO)

At the end of this subject, students should be able to:

1. Work with the Python standard libraries
2. Implement mutability for various elements of python
3. Develop GUI based projects
4. Design Networking configuration for chatting applications
5. Implement Scientific Computing
6. Solve Real world problems using python programming

Course Outline

UNIT-I	[12 hours]
--------	------------

Basic elements of python, Branching, looping, Strings and Input, Iteration, Functions, Recursion, Global variables, Modules, Files, Structured Objects, Mutability: Strings, Tuples, Lists, Sets, Dictionaries, Functions as Objects, Mutability and Higher-Order Functions.

UNIT-II	[12 hours]
---------	------------

Object-Oriented Programming, Abstract Data Types and Classes, Encapsulation and Information Hiding, Simple Algorithms and Data structures, Regular Expressions – REs, Networking, Multithreading in Python.

UNIT-III	[12 hours]
----------	------------

Array computing and curve plotting, vectors and higher-dimensional arrays, matrices, numPy, sciPy and Matplotlib, Plotting using PyLab, Chat Application, Graphics and GUI Programming – Drawing using Turtle, Tkinter.

UNIT-IV	[12 hours]
---------	------------

Python Pandas - Data alignment, aggregation, summarization, computation and analysis with Pandas. Scientific computation using python - Statistical data analysis, image processing, Basics of Web development(Introduction to frameworks flask, tensor flow).

Method of delivery

Chalk and Board, PowerPoint presentation

Study time

3 hrs theory, 2 Hrs practical

CO-PO Mapping (PO: Program Outcomes)

Engineering Graduates will be able to:

PO1 Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3 Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4 Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6 The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Mapping of CO with PO's

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	2	-	-	-	-	-	-	-	-	-	-	-
CO 2	3	1	1	-	-	-	-	-	-	-	-	-
CO 3	3	3	2	-	-	-	-	-	-	-	-	-
CO 4	2	1	-	-	-	-	-	-	-	-	-	-
CO 5	3	3	2	-	-	-	-	-	-	-	-	-
CO 6	3	3	2	-	-	-	-	-	-	-	-	-

Bloom's Taxonomy and Knowledge retention (For reference)

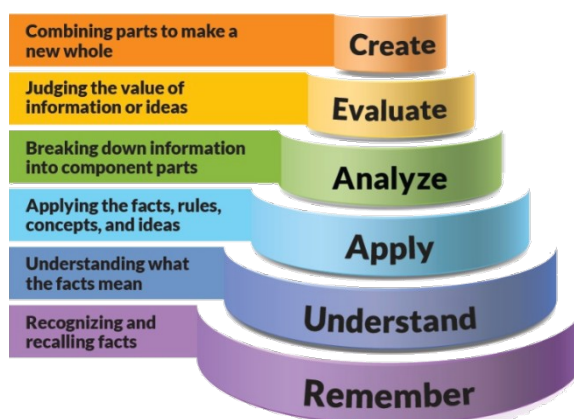


Figure 1: Bloom's Taxonomy

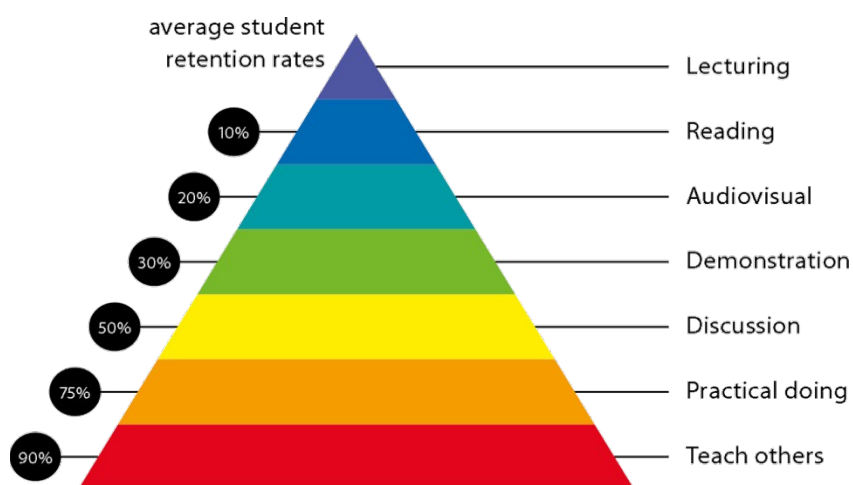


Figure 2: Knowledge retention

Graduate Qualities and Capabilities covered

General Graduate Qualities	Specific Department of CSE Graduate Capabilities
Informed Have a sound knowledge of an area of study or profession and understand its current issues, locally and internationally. Know how to apply this knowledge. Understand how an area of study has developed and how it relates to other areas.	1 Professional knowledge, grounding & awareness
Independent learners Engage with new ideas and ways of thinking and critically analyze issues. Seek to extend knowledge through ongoing research, enquiry and reflection. Find and evaluate information, using a variety of sources and technologies. Acknowledge the work and ideas of others.	2 Information literacy, gathering & processing
Problem solvers Take on challenges and opportunities. Apply creative, logical and critical thinking skills to respond effectively. Make and implement decisions. Be flexible, thorough, innovative and aim for high standards.	4 Problem solving skills
Effective communicators Articulate ideas and convey them effectively using a range of media. Work collaboratively and engage with people in different settings. Recognize how culture can shape communication.	5 Written communication
	6 Oral communication
	7 Teamwork
Responsible Understand how decisions can affect others and make ethically informed choices. Appreciate and respect diversity. Act with integrity as part of local, national, global and professional communities.	10 Sustainability, societal & environmental impact

Practical work:

1	<ol style="list-style-type: none"> 1. Demonstrate functions and all categories of functions in C/C++/java/python. 2. Demonstrate arrays (Insertion, Deletion, arithmetic operations). 3. Perform Addition, Multiplication for Vectors and Matrices. 4. Practical based on Strings (Length finding, change specific character, palindrome, concatenation) 5. Create a menu driven program to show various operators supported by python. 6. Write a python program to find out if a given number is even or odd using a user defined function. 7. 1) Addition of first 15 numbers using loop. 2) Addition of any 15 numbers using a loop. 8. Write a python program to check if the entered year is leap year or not. 9. Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them. 10. Write a Python program to print the calendar of a given month and year. 11. Write a Python program to check whether a specified value is contained in a group of values. Test Data: c. -> [1, 5, 8, 3]: True -1 -> [1, 5, 8, 3]: False 12. Write a Python program to get OS name, platform and release information. 	SIMPLE PROGRAMS
2	<ol style="list-style-type: none"> 1. Write a function to find out x^y. Function should find out the square of x in case of default argument passing. 2. Write a function to find out the factorial of a given number. I) without recursion II) with recursion 3. Write a program to find out Fibonacci series using recursion and function as an object. 	USER DEFINED FUNCTIONS (RECURSION, ARGUMENT PASSING, SCOPING)

	4. Demonstrate Function Scoping.	
3	<ol style="list-style-type: none"> Write a Python program to check whether a specified value is contained in a group of values using lambda function. Test Data: 3 -> [1, 5, 8, 3] : True -1 -> [1, 5, 8, 3] : False Write a Python program to find whether a given string ends with a given character using Lambda. Prime or not prime. Input: L= [3,4,6,9,11] Output: L= [P, NP, NP, NP, P] using map function. Write a python program to find out even numbers from a list using filter (). 	HIGHER ORDER FUNCTIONS (LAMDA, MAP, FILTER)
4	An interactive program where one module asks numbers from the user and the second module performs at least three arithmetic operations on them.	MODULES
5	<ol style="list-style-type: none"> A program to count the number of words, number of lines, occurrence of particular word, occurrence of particular character, number of blank spaces in a text file. A program to read a string from the user and append it into a file. A program to copy the contents of one file into another. A program to read a text file and print all the numbers present in the text file. A program to append the contents of one file to another file. A program to read a file and capitalize the first letter of every word in the file. 	FILE HANDLING
6	<ol style="list-style-type: none"> Write a Python program which accepts a sequence of comma-separated numbers from the user and generate a list and a tuple with those numbers. 30. Write a Python program to display the first and last colours from the following list. [orange, purple, red, yellow, blue] Write a Python program to concatenate all elements in a list into a string and return it. 	Collection Object Programs:

	<p>5. Write a Python program to print out a set containing all the colours from color_list_1 which are not present in color_list_2.</p> <p>Test Data: color_list_1 = set (["White", "Black", "Red"]) color_list_2 = set (["Red", "Green"])</p> <p>Expected Output: {'Black', 'White'}</p> <p>6. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are squares of keys.</p> <p>Sample Dictionary: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10:100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}</p> <p>7. Write a menu driven program to implement the following methods on List. (1) Create (2) Update particular element of list (3) Append to the list (4) Delete whole list (5) Delete particular element (6) Sort the list (7) Find length</p> <p>8. Write a menu driven program to implement the following methods on set. (1) Create (2) Update particular element of set (3) Append to the set (4) Delete whole set (5) Delete particular element (6) Sort the set (7) Find length</p> <p>9. Write a menu driven program to implement following methods on Dictionary. (1) Create (2) Update particular element of dictionary (3) Append to the dictionary (4) Delete whole dictionary (5) Delete particular element (6) Sort the dictionary (7) Find length</p> <p>10. Do following: 1) Define and Call user defined function for n numbers one by one. 2) Check if each element is Even or Odd and Print it. (Use Class Variable) 3) Print List of even numbers and odd numbers.</p> <p>11. Print total number of Even numbers and Total number of Odd Numbers.</p> <p>Example: 1) function(21), function(22), function(35),function(36),function(40).</p> <p>2) 21 is odd, 22 is even, 35 is odd,36 is even, 40 is even.</p> <p>3) Even={22,36,40} Odd={21,35} total number of even = 3, total number of odd=2</p>	
7	1. Implement following inheritance:	Object- Orientation Programs:

	<p>(1)Single (2) Multiple (3) Multilevel (4) Hybrid</p> <p>2. Demonstrate Overriding and methods to overcome.</p>	
8	<p>1. Implement following searching techniques:</p> <p>(1) Linear (2) Binary</p> <p>2. Implement following sorting algorithms.</p> <p>(1) Selection (2) Merge (3) Tim</p> <p>3. Implement a Singly Linked List. (Insert, Display, Delete).</p>	Data Structures Programs:
9	<p>1. Implement following Regular Expressions.</p> <p>1) Write a Python Program that searches a string to see if it starts with "The " and ends with "Indus".</p> <p>2) Write a Python Program that returns a match where the string contains a white space character.</p> <p>3) Write a Python program that matches a string that has an a followed by three 'b'.</p> <p>4) Write a Regular Expression to find Words or Strings having three characters and with 'm' as first character.</p> <p>5) Write a regular expression to retrieve all words starting with "a".</p> <p>6) Write a regular expression to retrieve all words starting with a numeric digit.</p> <p>7) Write a regular expression to retrieve all words having 5 character length.</p> <p>8) Write a regular expression to retrieve all words with 3,4 or 5 character length. [a-z]{3,5}</p> <p>9) Write a regular expression to retrieve all single digits from a string.</p> <p>10) Write a regular expression to retrieve the last word from the string.</p> <p>11) Write a regular expression to retrieve all words starting with 'an' or 'ak'.</p> <p>12) Write a regular expression to retrieve DOB from the string.</p>	Regular Expressions Programs:
10	<p>1. Implement Chat Application.</p>	Networking and Multithreading

	2. Demonstrate multithreading.	Programs:															
11	1. Create a tkinter GUI to implement a calculator. 2. Create a dropdown list to select a city from the given list of cities. 3. Write a tkinter code to place an image/picture in the window. 4. Create Registration window.	GUI Using Tkinter:															
12	1. Using the Turtle module in python, Draw Square, Rectangle, Star and explore various attributes with it (ex: background color, line fill, border color, etc...)	GUI Using Turtle:															
13	1. a) Extract all odd numbers from the array using the “where” clause. b) Replace all odd numbers in array with -1 c) Convert a 1D array to a 2D array with 2 rows and 5 columns. d) Get the common items between array1 and array2 Input: array1 = [1,2,3,2,3,4,3,4,5,6] array2 = [7,2,10,2,7,4,9,4,9,8] Desired Output: array([2, 4]) e) Perform Matrix multiplication on 2 matrices. f) Using numpy and matplotlib/pylab generate bar plots for appropriate data.	Numpy															
14	1. Create a DataFrame fruits that looks like this: <table border="1" data-bbox="311 1608 877 1675"> <thead> <tr> <th></th><th>Apples</th><th>Bananas</th></tr> </thead> <tbody> <tr> <td>0</td><td>30</td><td>21</td></tr> </tbody> </table> 2. Create a dataframe fruit_sales that matches the diagram below: <table border="1" data-bbox="292 1787 949 1888"> <thead> <tr> <th></th><th>Apples</th><th>Bananas</th></tr> </thead> <tbody> <tr> <td>2017 Sales</td><td>35</td><td>21</td></tr> <tr> <td>2018 Sales</td><td>41</td><td>34</td></tr> </tbody> </table> 3. Create a variable ingredients with a Series that looks like:		Apples	Bananas	0	30	21		Apples	Bananas	2017 Sales	35	21	2018 Sales	41	34	
	Apples	Bananas															
0	30	21															
	Apples	Bananas															
2017 Sales	35	21															
2018 Sales	41	34															

	Flour 4 cups Milk 1 cup Eggs 2 large Spam 1 can Name: Dinner, dtype: object 4. Download file name: winemag-data-130k-v2.csv and perform the following. 1) Display row number 25 2) Display column number 13 3) Display rows where country name = France 4) Display records where province=Michigan and taster_name=Alexander Peartree 5) Explore describe() for the generated dataframe. 5. To display any Sports Scoreboard generate appropriate plot (line/bar/scatter, etc..)	
15	59. Basic Program of Flask. 60. Basic Program of TensorFlow.	Miscellaneous:

Lecture/Tutorial times

Lecture	Tuesday	11.45 AM to 12.40 PM
Lecture	Wednesday	2.15 PM to 3.10 AM
Lecture	Thursday	1.35 PM to 2.25 PM
Practical (C1)	Wednesday	10.50 AM to 12.40 PM
Practical (C2)	Thursday	9.00 AM to 10.50 PM

Attendance Requirements

The University norms state that it is the responsibility of students to attend all lectures, tutorials, seminars and practical work as stipulated in the course outline. Minimum attendance requirement as per university norms is compulsory for being eligible for semester examinations.

Reference Books:

1. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
2. R. Nageswara Rao, "Core Python Programming", dreamtech
3. Wesley J. Chun. "Core Python Programming - Second Edition", Prentice Hall

4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, “Data Structures and Algorithms in Python”, Wiley
5. Kenneth A. Lambert, “Fundamentals of Python – First Programs”, CENGAGE Publication
6. Luke Sneeringer, “Professional Python”, Wrox
7. “Hacking Secret Ciphers with Python”, Al Sweigart,
URL-<https://inventwithpython.com/hacking/chapters>

Online Courses:

1. <https://www.youtube.com/watch?v=N4mEzFDjqtA>
2. <https://www.youtube.com/watch?v=hnXIRVZ0EyU>
3. <https://www.youtube.com/watch?v=tKTZoB2Vjuk>

ASSESSMENT GUIDELINES

CIE - Theory (60 Marks)	CIE - Practical (60 Marks)
Mid Semester Exam : 40 Marks	Mini Project + Viva : 20 +10 Marks
Regularity+Class Performance : 5 Marks	Practical Performance : 20 Marks
Assignment/ Quiz : 15 Marks	Lab File : 10 Marks
ESE-Theory- 40 Marks	ESE-Practical-40 Marks
Total: 100 Marks	Total: 100 Marks

Your final course mark will be calculated from the following:

SUPPLEMENTARY ASSESSMENT

Students who receive an overall mark less than 40% in internal component or less than 40% in the end semester will be considered for supplementary assessment in the respective components (i.e internal component or end semester) of semester concerned. Students must make themselves available during the supplementary examination period to take up the respective components (internal component or end semester) and need to obtain the required minimum 40% marks to clear the concerned components.

Practical Work Report/Laboratory Report:

A report on the practical work is due the subsequent week after completion of the class by each group.

Late Work

Late assignments will not be accepted without supporting documentation. Late submission of the reports will result in a deduction of -% of the maximum mark per calendar day

Format

All assignments must be presented in a neat, legible format with all information sources correctly referenced. **Assignment material handed in throughout the session that is not neat and legible will not be marked and will be returned to the student.**

Retention of Written Work

Written assessment work will be retained by the Course coordinator/lecturer for two weeks after marking to be collected by the students.

University and Faculty Policies

Students should make themselves aware of the University and/or Faculty Policies regarding plagiarism, special consideration, supplementary examinations and other educational issues and student matters.

Plagiarism - Plagiarism is not acceptable and may result in the imposition of severe penalties. Plagiarism is the use of another person's work, or idea, as if it is his or her own - if you have any doubts at all on what constitutes plagiarism, please consult your Course coordinator or lecturer. Plagiarism will be penalized severely.

Do not copy the work of other students.

Do not share your work with other students (except where required for a group activity or assessment).

Course schedule (subject to change)

(Mention quiz, assignment submission, breaks etc as well in the table under the Teaching Learning Activity Column)

	Week #	Topic & contents	CO Addressed	Teaching Learning Activity (TLA)
	Week 1	Basic elements of python, Branching, looping, Strings and Input, Iteration	I	Chalk & Board, Discussion
	Week 2	Functions, Recursion, Global variables	I	Presentation, Chalk & Board
	Week 3	Modules, Files	I	Presentation, Chalk & Board
	Week 4	Structured Objects, Mutability: Strings, Tuples, Lists, Sets, Dictionaries	II	Presentation, Chalk & Board
	Week 5	Functions as Objects, Mutability and Higher-Order Functions	II	Presentation, Chalk & Board
	Week 6	Object-Oriented Programming	II	Model presentation
	Week 7	Abstract Data Types and Classes, Encapsulation and Information Hiding	II	Presentation, Chalk & Board, Demonstration
	Week 8	Simple Algorithms and Data structures	II	Presentation, Chalk & Board, Demonstration
	Week 9	Regular Expressions – REs ,Networking, Multithreading in Python	II	Presentation, Chalk & Board
	Week 10	Array computing and curve plotting	II	Presentation, Chalk & Board
	Week 11	Vectors and higher-dimensional arrays, matrices, numPy, sciPy and Matplotlib, Plotting using PyLab	III	Presentation, Chalk & Board
	Week 12	Chat Application, Graphics and GUI Programming – Drawing using Turtle, Tkinter	III	Presentation, Chalk & Board
	Week 13	Python Pandas - Data alignment, aggregation, summarization, computation and analysis with Pandas	IV	Presentation, Chalk & Board

Week 14	Scientific computation using python - Statistical data analysis, image processing	VI	Presentation, Chalk & Board
Week 15	Basics of Web development(Introduction to frameworks flask, tensor flow)	VI	Presentation, Chalk & Board

COMPUTER ENGINEERING DEPARTMENT COURSE DEPENDANCY CHART

