

CST 283	Python for Machine Learning	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is a programming course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in *Machine Learning*. The objective of the course is to provide learners an insight into Python programming, and develop programming skills to manage the development of software systems. It covers programming environment, important instructions, data representations, intermediate level features, Object Oriented Programming and file data processing of Python. This course lays the foundation to develop web applications, Machine Learning, and Artificial Intelligence-based applications and tools, Data Science and Data Visualization applications.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO1	Write, test and debug Python programs (Cognitive Knowledge level: Apply)
CO2	Illustrate uses of conditional (if, if-else, if-elif-else and switch-case) and iterative (while and for) statements in Python programs (Cognitive Knowledge level: Apply)
CO3	Develop programs by utilizing the modules Lists, Tuples, Sets and Dictionaries in Python (Cognitive Knowledge level: Apply)
CO4	Implement Object Oriented programs with exception handling (Cognitive Knowledge level: Apply)
CO5	Write programs in Python to process data stored in files by utilizing the modules Numpy, Matplotlib, and Pandas (Cognitive Knowledge level: Apply)

Mapping of course outcomes with program outcomes

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

Abstract POs defined by National Board of Accreditation

#PO	Broad PO	#PO	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (<i>Marks in percentage</i>)	Test 2 (<i>Marks in percentage</i>)	End Semester Examination (<i>Marks in percentage</i>)
Remember	20	20	20
Understand	35	35	35
Apply	45	45	45
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks
 Continuous Assessment Test : 25 marks
 Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

SYLLABUS

Module I

Programming Environment and Python Basics:

Getting Started with Python Programming - Running code in the interactive shell, Editing, Saving, and Running a script. Using editors - IDLE, Jupyter. The software development process - Case Study.

Basic coding skills - Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, Working with numeric data, Type conversions, Comments in the program. Input, Processing, and Output. Formatting output. How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module.

Module II

Building Python Programs:

Control statements - Selection structure (if-else, switch-case), Iteration structure (for, while), Testing the control statements, Lazy evaluation. Functions - Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, Named arguments, Main function, Working with recursion, Lambda functions. Strings and number systems - String function, Handling numbers in various formats.

Module III

Data Representation:

Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Work with dates and times. Dictionaries - Dictionary

functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup. Case Study - Data Structure Selection.

Module IV

Object Oriented Programming:

Design with classes - Objects and Classes, Methods, Instance Variables, Constructor, Accessors and Mutators. Structuring classes with Inheritance and Polymorphism. Abstract Classes. Exceptions - Handle a single exception, handle multiple exceptions.

Module V

Data Processing:

The *os* and *sys* modules. Introduction to file I/O - Reading and writing text files, Manipulating binary files. NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization. Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data.

Text Books:

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017

Reference Books:

1. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
2. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
3. David M.Baezly, Python Essential Reference. Addison-Wesley Professional; 4/e, 2009.
4. Charles Severance. Python for Informatics: Exploring Information,
5. <http://swcarpentry.github.io/python-novice-gapminder/>

Sample Course Level Assessment Questions

Course Outcome1(CO1): What is type conversion? How is it done in Python?

Course Outcome 2(CO2): Write a Python program which takes a positive integer *n* as input and finds the sum of cubes all positive even numbers less than or equal to the number.

Course Outcome 3(CO3): Given is a list of words, *wordlist*, and a string, *name*. Write a Python function which takes *wordlist* and *name* as input and returns a tuple. The first element of

the output tuple is the number of words in the *wordlist* which have *name* as a substring in it. The second element of the tuple is a list showing the index at which the *name* occurs in each of the words of the *wordlist* and a 0 if it doesn't occur.

Course Outcome 4(CO4): Write a Python program to implement the addition, subtraction, and multiplication of complex numbers using classes. Use constructors to create objects. The input to the program consist of real and imaginary parts of the complex numbers.

Course Outcome 5(CO5): Given a file “auto.csv” of automobile data with the fields *index*, *company*, *body-style*, *wheel-base*, *length*, *engine-type*, *num-of-cylinders*, *horsepower*, *average-mileage*, and *price*, write python code to

- 1) Clean and Update the CSV file
- 2) Print total cars of all companies
- 3) Find the average mileage of all companies
- 4) Find the highest priced car of all companies.

