CS 3108 Soft Skills (skilled course)

Theory and Practice : 3 hrs Internal marks: 50 External Marks: 50 Total Marks: 100

Course Objectives:

- To develop skills to communicate clearly.
- To aid students in building interpersonal skills.
- To enhance team building and time management skills.
- To inculcate active listening and responding skills.

Course Outcomes:

- Make use of techniques for self-awareness and self-development.
- Apply the conceptual understanding of communication into everyday practice.
- Understand the importance of teamwork and group discussions skills.
- Develop time management and stress management.

Learning Outcomes:

- Acquisition of etiquette and skills that an engineer requires.
- Students will develop the acumen for self-awareness and self-development.
- Students will be able to communicate unmistakably.
- Students will be able to tackle real-life challenges.

Introduction to Soft Skills: Communication – Verbal and Non Verbal Communication - Personal grooming (Etiquette, Attitude, Body Language), Posture, Gestures, Facial Expressions, Eye Contact, Space Distancing, Presentation Skills, Public Speaking, Just a Minute (JAM) sessions, Adaptability.

Goal Setting and Time Management: Immediate, Short term, Long term, Smart Goals, Strategies to Achieve goals, Types of Time, Identifying Time Wasters, Time Management Skills, Stress Busters.

Leadership and Team Management: Qualities of a Good Leader, Team Dynamics, Leadership Styles, Decision Making, Problem Solving, Negotiation Skills.

Group Discussions: Purpose (Intellectual ability, Creativity, Approach to a problem, Tolerance), Group Behaviour, Analysing Performance.

Job Interviews: Identifying job openings, Covering Letter and CVs / Resumes, Interview (Opening, BodyAnswer Q, Close-Ask Q), Telephone Interviews, Types of Questions.

Reference Books:

1. Krannich, Caryl, and Krannich, Ronald L. Nail the Resume! Great Tips for Creating Dynamite Resumes. United States, Impact Publications, 2005.
2. Hasson, Gill. Brilliant Communication Skills. Great Britain: Pearson Education, 2012
3. Prasad, H. M. How to Prepare for Group Discussion and Interview. New Delhi: Tata McGrawHill Education, 2001.
4. Pease, Allan. Body Language. Delhi: Sudha Publications, 1998.
5. Rizvi, Ashraf M. Effective Technical Communication: India, McGraw-Hill Education. 2010
6. Thorpe, Edgar & Showick Thorpe. Winning at Interviews. 2nd Edition. Delhi: Dorling Kindersley, 2006.

SCHEME & SYLLABUS For All Semesters
(With effect from 2022-23 Admitted Batch)

B.Tech & B.Tech + M.Tech
(Computer Science and Engineering)

Department of Computer Science & Systems Engineering

Andhra University College of Engineering (A)

Andhra University

Guidelines for Obtaining MINOR in Computer Science Engineering:

Students belonging to other departments have to complete the following courses to obtain MINOR degree in Computer Science & Engineering:

1. A student belonging to other department have to study CPNM and Python Programming as Compulsory courses and any two of the following 4 open electives as follows:

   OE I: Artificial Intelligence & Machine Learning

   OE II: Data Science

   OE III: Cyber Security and Digital Forensics

   OE IV: Database Management System

   OR

   One or two MOOCS courses from NPTEL related to Computer Science Engineering without repetition from subjects within the curriculum can be used in Lieu of any of the above Open electives.
2. The duration of NPTEL courses should **NOT** be less than 12 weeks.
3. The MOOCS course(s) chosen by students of other departments for obtaining a MINOR in CSE should be taken prior permission/ approval from the Chairperson – BoS of the department of **CS & SE, AUCE(A)**.

Guidelines for Obtaining HONORS in Computer Science Engineering:

1. The student shall earn additional 15 credits beyond 160 credits from the same branch/ department/ discipline registered for major degree.
   (i) The students having 7.0 CGPA without any backlog subjects will be permitted to register for HONORS.
   (ii) If a student is detained due to lack of attendance either in Major or in Honors, registration shall be cancelled.
   (iii) Honors is to be completed simultaneously with B. Tech program.

To obtain Honors in CSE the following subjects are to be taken for obtaining Honors:

1) Large Language Model & Prompt Engineering
2) Reinforcement Learning
3) High Performance Computing
4) Social Media Analytics
5) Software Metrics
6) MOOCs for 3 credits of 12 weeks duration related to any one of the above courses. MOOCs may be treated as optional to obtain Honors degree in CSE. If a student completes MOOCs with good grade, then it may be considered as betterment for any of the above five Courses (1 to 5)

The above five courses and MOOCs may be completed or pursued during 3rd and 4th years of his/her study of B.Tech, B.Tech + M.Tech program

(With effect from 2022-23 AB)
B. Tech & B. Tech + M. Tech
(Computer Science and Engineering)
III year - I semester

Course Code	Category	Course Title	Hours per week		Internal Marks	External Marks	Total Marks	Credits
			L	P				
CS3101	PC	Data Communications & Computer Networks	4	0	30	70	100	3
CS3102	PC	Artificial Intelligence	4	0	30	70	100	3
CS3103	PC	Compiler Design	4	0	30	70	100	3
CS3104	PE	Professional Electives - I	4	0	30	70	100	3
CS3105	OE	Open Electives - I	4	0	30	70	100	3
CS3106	PC	Data Communications & Computer Networks Lab	0	3	50	50	100	1.5
CS3107	PC	Python Programming Lab	0	3	50	50	100	1.5
CS3108	SC	Soft Skills	1	2	50	50	100	2
CS3109	INT	Internship - I	0	0	50	50	100	2
Summer Internship 2 Months (Mandatory) after 2nd year (to be evaluated during III year I semester)								
Total Credits								**22**

CS3101 DATA COMMUNICATIONS & COMPUTER NETWORKS

Course Objectives:

- To study basics of data communication systems.
- To study the various types of transmission media.
- To study the various hardware concepts related to data communications.
- To make the students understanding of basic requirements of network hardware, software and its architecture.

Course Outcomes:

- Ability to understand concepts related to data communication hardware and its interface.
- Ability to understand concepts related to Signal encoding techniques and multiplexing.
- The student must be able to understand the concepts related to MAC sub layer.
- Understand the concepts related to network and transport layer.

SYLLABUS:

Introduction to Data Communications: A Communications Model, Network Models(OSI, TCP/IP models), Analog and Digital Data Transmission, Transmission Impairments.

Data Communication Interface: Asynchronous and Synchronous Transmission. Data Link Control Flow Control, Error Detection, Error Control, High-Level Data Link Control (HDLC).

Signal Encoding Techniques: Digital data to Digital signal, Digital to Analog Signal, Analog data to Digital Signal, and Analog Data to Analog signal.

Multiplexing: Frequency-Division Multiplexing, Synchronous Time-Division Multiplexing: Characteristics, Digital Carrier Systems Statistical Time-Division Multiplexing: Characteristics.

Medium Access Control Sublayer: Wireless LAN's:802.11Architecture and Protocol Stack, 802.11Frame structure.

Network Layer: Network Layer Design Issues, Shortest path routing algorithm, Congestion Control Algorithms, IP Protocol, IP Address.

Transport layer: Transport Service, Elements of Transport Protocols, TCP and UDP Protocols, Simple Network Management Protocol (SNMP).

Text Books:

1. Data Communications and Networking, Behrouz A Forouzan, Tata McGraw-Hill Co Ltd, Second Edition, ISBN: 0-07-049935-7

2. Computer Networks, Andrews S Tanenbaum, 5th Edition, Pearson Edu.

References:

1. Data and Computer Communications, Eighth Edition, William Stallings, Pearson Education, Inc.

CS3102 ARTIFICIAL INTELLIGENCE

Course Objectives:

- To learn about AI problems, techniques and their modelling as state space search, problem characteristics, Production System categories.
- To learn different uninformed and heuristic search strategies for solving AI problems with examples
- To learn theorem proving with predicate logic, resolution, rulebased inference with forward and backward chaining
- Inheritable knowledge representation using slot-filler structures and dealing with different forms of uncertain and implicit knowledge
- To introduce essential concepts of plan generation, Natural Language understanding and Expert Systems.

Course Outcomes:

- By the end of the course the student understands, applies, evaluates and creates AI solutions as they are
- able to characterize and model AI problems in a state space search framework and identify appropriate production system category to solve them
- able to understand and evaluate pros & cons of different heuristic search strategies and apply appropriate heuristic search for specific problem solving scenario.
- able to represent domain knowledge in the form of predicates / rules and applies logic and inference for deducing the validity of a given assertion.
- able to create problem specific slot-filler knowledge structures and apply statistical, fuzzy and non-monotonic reasoning methods aptly to solve real world problems involving any type of uncertainty.
- able to understand basic concepts and approaches to natural language processing, plan generation and expert system development.

SYLLABUS:

Introduction to Artificial Intelligence: Artificial Intelligence, AI Problems, AI Techniques, Defining the Problem as a State Space Search, Production Systems, control strategies, Uninformed search using BFS and DFS, Heuristic search, Problem Characteristics, Production system categories for AI problem solving

Heuristic Search Techniques: Issues in The Design of Search Programs, Generate-And- Test, Hill Climbing and its variants, Best-First Search, A* Algorithm, Problem Reduction, AO*Algorithm, Constraint Satisfaction, Means-Ends Analysis.

Knowledge Representation using Predicate Logic and Rules: Propositional Logic, First Order Predicate Logic: Representing Instance and is-a Relationships, Computable Functions and Predicates, Resolution in propositional logic, Resolution in predicate logic with Clause form, Unification & Resolution algorithm, Question answering, Procedural Vs Declarative Knowledge, Logic programming with Prolog, Forward Vs Backward Reasoning and combining them, Matching Techniques, Matching with variables, RETE Algorithm, Conflict resolution

Reasoning under Uncertainty: Introduction to Non-Monotonic Reasoning, Logics for Non monotonic reasoning, Depth first search with Dependency-directed backtracking, Justification based Truth Maintenance

System, Statistical Reasoning: Bayes Theorem for probabilistic inference, Certainty Factors and Rule-Based Systems, Bayesian Belief Networks, Dempster Shafer Theory, Fuzzy Logic

Structured Representations of Knowledge: Semantic Nets, representing non-binary predicates, Partitioned Semantic Nets, Frames as sets and instances, Slots as full-fledged Objects, Property Inheritance through tangled hierarchies, Conceptual Dependency, Conceptual Dependency Graphs, Scripts, examples in natural language understanding, merits and demerits of strong slot filler structures.

Natural Language Processing: Steps in Natural Language Processing, Syntactic Processing and Augmented Transition Nets, Semantic Analysis and grammars, Discourse and pragmatic processing; Planning: Components of a Planning System, Goal Stack Planning, Non-linear Planning using Constraint Posting, Hierarchical Planning, Reactive Systems.

Experts Systems: Overview of an Expert System, Applications of expert systems, Components of an Expert Systems, Expert system development, Types of Expert Systems: Rule Based, Frame Based, Neural Network based, Black Board Architectures, Case studies of successful expert systems, Expert System Shells, Knowledge Acquisition and Validation Techniques.

Text Books:

1. Artificial Intelligence, Elaine Rich and Kevin Knight, Tata McGraw -Hill Publications 2. Introduction To Artificial Intelligence & Expert Systems, Patterson, PHI publications References:

2. Artificial Intelligence, George F Luger, Pearson Education Publications 2. Artificial Intelligence : A modern Approach, Russell and Norvig, Prentice Hall

CS3103 COMPILER DESIGN

Course objectives:

- To explain the basic understanding of grammars and language definition and introducing various phases of designing a compiler.
- To make the student to understand the concepts underlying the design and implementation of language processors and its mechanisms.
- To extend the knowledge of parser by parsing LL parser and LR parser.
- To enrich the knowledge in various phases of compiler ant its use, code optimization techniques, loop optimization techniques, machine code generation, and use of symbol table.

Course outcomes:

- Ability to understand grammars, language definitions and various phases of designing a compiler.
- Ability to understand Language processors and different parsers.
- Ability to learn the new code optimization techniques to improve the performance of a program in terms of speed & space.
- Ability to do usage of registers in efficient manner during the program execution.
- Ability to acquire the knowledge of modern compiler & its features.

SYLLABUS:

Introduction: Introduction to Compilers and Language processors, Programming Language basics, Structure & Different Phases of a Compiler, Review of Compiler Structure, Structure of Optimizing Compilation, Compiler construction tools, Boot strapping, Cross compilers.

Finite Automata & Lexical Analysis: Introduction to Lexical Analysis, Lexical Analyzers, Approaches to design Lexical Analysers, Language for specifying lexical analyzers, Introduction to Finite automata, Regular Expressions & Languages, Recognition of Tokens, Transition Diagrams, Implementation of lexical analyzers, Lexical Analyzer Generator LEX.

Syntax Analysis: Syntactic Specification of Programming Languages, Context Free Grammars & Languages, Introduction to Parsers. Top-down parsing techniques: Brute force parsing, Recursive Descent Parsing, Predictive Parsing, Bottom–up Parsing: Shift reduce parsing, Operator parsing, LR (k) parsing.

Semantic Analysis and Intermediate Code Generation: Semantic Actions, Syntax Directed Translations, Translation on the parse Tree, Implementation of Syntax Directed Translator, Intermediate Codes, Syntax Directed translation to Postfix code, Syntax Trees, Intermediate Code Generation, Three Addr5ess Code-Translation of Expressions, Type Checking & Type Conversions.

Code Optimization: Principal sources of Code Optimization, Loop Optimization, Basic Blocks & Flow Graphs, DAG Representation of Basic Blocks, Applications of DAG, Local Optimization, Unreachable Code Elimination, Dead Code Elimination, Data Flow Analysis, Data Flow Equations & Computations, Peep-Hole Optimization.

Machine Dependent Optimizations, Overview of Informal Compiler Algorithm Notation (ICAN), If Simplification, Loop Simplification, Loop Inversion, Branch Optimization and Prediction

Code Generation and Code Scheduling: Issues in Code Generation, Input to Code Generator, Instruction Selection, Register Allocation, Simple Target Machine Model, Program and Instruction Costs, Register allocation & Assignments, Code Generation Algorithm, Code Generators, Optimal Code Generation for Expressions, Code Generation From DAG.

Symbol Tables, Runtime Environment and Error Handling: Contents of a Symbol Table, Data Structures for Symbol Tables; Run time Environments, Implementation of a simple Stack allocation, Heap Management, Block Structured Languages; Error Detection & Recovery, Lexical Phase Errors, Syntactic & Semantic Errors, Error Handling Routines.

Text Books:

1. Principles of Compiler Design by Aho.D. Ullman, Lam and Ravi Sethi, Pearson Education Second Edition
2. Advanced Compiler Design and Implementation, Steven Muchnic, Elsevier Publications.

Reference Books:

1. Compiler Construction by Kenneth. C. Louden, Vikas Pub. House.
2. Compiler Design, A.A. Pentambekar, Technical Publications
3. Modern Compiler Design, Grune.D, Van Reeuwijk K, Bal H.E, Jacobs C J H, Lan gendoen K, Springer,

CS3106 DATA COMMUNICATIONS & COMPUTER NETWORKS LAB

Course Objectives:

- This course provides students with hands on training regarding the design, troubleshooting, modelling and evaluation of computer networks.
- To study the various hardware concepts related to data communications
- To make the students understand the basic requirements of network hardware, software and its architecture.
- To understand the various Error detection & control techniques
- To learn Socket programming techniques

Course Outcomes:

The students will learn:

- About networking concepts and connecting systems
- To setup Local Area Network using packet tracer software
- To experiment in a real tested networking environment, network design and troubleshooting topics and tools
- Simulator Error control and flow control techniques
- To write socket program and client server applications.

SYLLABUS: Module I: Packet tracer software

1. Study of different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.
2. Connect the computers in Local Area Network.
3. Study of basic network command Network configuration commands.
4. Configure a Network topology using packet tracer software.

Module II: Network simulator (NS)

1. Implementation of Error Detection/Error Correction Techniques
2. Implementation of Stop and Wait Protocol and sliding window
3. Implementation and study of Goback-N and selective repeat protocols
4. Implementation of High-Level Data Link Control
5. Study of Socket Programming and Client-Server model using Java
6. Write a socket program for Echo/Ping/Talk commands using Java
7. Study of Network simulator (NS) and simulation of Congestion Control Algorithms

CS3107 PYTHON PROGRAMMING LAB

Course Objectives:

- familiarize students with key data structures in Python including lists and dictionaries and apply them in context of searching, sorting, text and file handling
- introduce students to calculation of statistical measures using Python such as measures of central tendency, correlation
- familiarize students with important Python data related libraries such as Numpy and Pandas and use them to manipulate arrays and data frames
- introduce students to data visualization in Python through creation of line plots, histograms, scatter plots, box plots and others
- Implementation of basic machine learning tasks in Python including pre-processing data, dimensionality reduction of data using PCA, clustering, classification and cross-validation.

Course Outcomes:

- After completion of the course the student should be able to:
- implement searching, sorting and handle text and files using Python data structures such as lists and dictionaries
- calculate statistical measures using Python such as measures of central tendency, correlation
- use Python data related libraries such as Numpy and Pandas and create data visualizations
- implement basic machine learning tasks pre-processing data, compressing data, clustering, classification and cross-validation.

SYLLABUS:

1. Python Programs on lists & Dictionaries

2. Python Programs on Searching and sorting

3. Python Programs on Text Handling

4. Python Programs on File Handling

5. Python Programs for calculating Mean, Mode, Median, Variance, Standard Deviation

6. Python Programs for Karl Pearson Coefficient of Correlation, Rank Correlation

7. Python Programs on NumPy Arrays, Linear algebra with NumPy

8. Python Programs for creation and manipulation of Data Frames using Pandas Library

9. Write a Python program for the following.

   - Simple Line Plots,

- Adjusting the Plot: Line Colours and Styles, Axes Limits, Labelling Plots,
- Simple Scatter Plots,
- Histograms,
- Customizing Plot Legends,
- Choosing Elements for the Legend,
- Boxplot
- Multiple Legends,
- Customizing Colorbars,
- Multiple Subplots,
- Text and Annotation,
- Customizing Ticks

10. Python Programs for Data pre-processing: Handling missing values, handling categorical data, bringing features to same scale, selecting meaningful features

11. Python Program for Compressing data via dimensionality reduction: PCA

12. Python Programs for Data Clustering

13. Python Programs for Classification

14. Python Programs for Model Evaluation: K-fold cross validation.

Reference Books:

1. Core Python Programming, Second Edition, Wesley J. Chun, Prentice Hall
2. Chris Albon,―Machine Learning with Python Cookbook-practical solutions from pre-processing to Deep learning‖, O'REILLY Publisher,2018
3. Mark Summerfield, Programming in Python 3--A Complete Introduction to the Python Language, Second Edition, Additson Wesley
4. Phuong Vo.T.H , Martin Czygan, Getting Started with Python Data Analysis, Packt Publishing Ltd
5. Armando Fandango, Python Data Analysis, Packt Publishing Ltd
6. Magnus VilhelmPersson and Luiz Felipe Martins, Mastering Python Data Analysis, Packt Publishing Ltd
7. Sebastian Raschka&VahidMirjalili, ―Python Machine Learning‖, Packt Publisher, 201

CLOUD COMPUTING
PROFESSIONAL ELECTIVE
SYLLABUS

Course Objectives:

- To define Cloud Computing and expose the students to the frontier areas of Cloud Computing.
- To provide an in-depth and comprehensive knowledge of the Cloud Computing fundamental issues, technologies, applications and implementations.
- To introduce various levels of services that can be achieved by cloud.
- To gain knowledge on virtualization techniques.
- To understand the working methodology of existing clouds, such as, Amazon, Google and Azure.

Course Outcomes:

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing.
- Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, laaS, public cloud, private cloud, hybrid cloud, etc.
- Classify the Levels of Virtualization and mechanism of tools and Analyze Cloud Architectures.
- Assess Control Storage Systems.
- Get an idea and set up Private Clouds.

SYLLABUS:

History of Computing Paradigms: Overview of Distributed Computing, Cluster Computing, Grid Computing, Ubiquitous Computing, Peer-to-Peer Computing. Distributed System Models and Enabling Technologies

Introduction to Cloud Computing: Cloud Computing and Service Models- Public, Private, and Hybrid Clouds, Cloud Ecosystem and Enabling Technologies, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

Virtual Machines and Virtualization: Implementation Levels of Virtualization, VMM Design Requirements and Providers, Virtualization Support at the OS Level, Middleware Support for Virtualization, Virtualization Structures/Tools and Mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management

Public Cloud Platform-Architectures and Programming: Google App Engine (GAE). Amazon Web Services (AWS), Microsoft Windows Azure. Service Oriented Architecture: REST, Publish Subscribe Model

Storage Systems: Storage Models, File Systems, and Databases, Distributed File Systems, General Parallel File System, Google File Systems, Apache Hadoop, Locks and Chubby: A Locking Service, Transaction Processing and NoSQL Databases, Big Table, Mega Store

Case Studies: The Grep The Web Application, Aneka Application of Maya Rendering Case Study

Text Books:

1. Kai Hwang, Geoffrey C. Fox, Jack K. Dongarra, Distributed and Cloud Computing: From parallel processing to Internet of Things, Morgen Kauffmann 2013.
2. Cloud Computing Theory and Practice, Dan C. Marinescu.

References:

1. Cloud Computing: A Hands-On Approach, Arshdeep Bagha & Vijay Madisetti, University Press, 2022 Edition
2. Cloud Computing: A Practical Approach Anthony T. Velte Toby J.Velte, Ph.D. Robert Elsenpeter
3. Cloud Computing Bible, Barrie Sosinsk
4. Cloud Computing Course (nptel.ac.in)

PYTHON PROGRAMMING
OPEN ELECTIVE
SYLLABUS

Course Objectives:

- To develop skills on procedural oriented and object oriented programming in Python.
- To understand key data structures in Python and apply different data wrangling techniques using Python.
- To understand data related libraries such as Numpy and Pandas.
- To understand concepts of data analysis using Python libraries
- To introduce exploratory data analysis using Matplotlib data visualization concepts

Course Outcomes:

After completion of the course the student should be able to:

- Acquire knowledge on basic Python programming and usage of associated Python Libraries
- Acquire knowledge on usage of Object oriented Programming concepts in Python.
- Acquire knowledge on pre-processing data, compressing data, clustering, classification and cross-validation.
- Acquire knowledge on Data analysis with Pandas
- Acquire Knowledge on implementing Data Visualization techniques in Python.

SYLLABUS:

Introduction to Python: Rapid Introduction to Procedural Programming, Data Types: Identifiers and Keywords, Integral Types, Floating Point Types Strings: Strings, Comparing Strings, Slicing and Striding Strings, String Operators and Methods, String formatting with str, format, Collections Data Types: Tuples, Lists, Sets, dictionaries, Iterating and copying collections, Introduction to PIP

Control Structures & Functions: Python Control Structures, Functions and OOP:Control Structures and Functions: Conditional Branching, Looping, Exception Handling, Custom Functions, Python Library Modules: random, math, time, os, shutil, sys, glob, re, statistics, creating a custom module

Object Oriented Programming: Object Oriented Concepts and Terminology, Custom Classes, Attributes and Methods, Inheritance and Polymorphism, Using Properties to Control Attribute Access, File Handling: Writing and Reading Binary Data, Writing and Parsing Text Files

NumPy Arrays and Vectorized Computation: NumPy arrays, Array creation, Indexing and slicing, Fancy indexing, Numerical operations on arrays, Array functions, Data processing using arrays, Loading and saving data, Saving an array, Loading an array, Linear algebra with NumPy, NumPy random numbers

Data Analysis with Pandas: An overview of the Pandas package, The Pandas data structure-Series, The DataFrame, The Essential Basic Functionality: Reindexing and altering labels , Head and tail, Binary operations, Functional statistics , Function application Sorting, Indexing and selecting data, Computational tools, Working with Missing Data, Advanced Uses of Pandas for Data Analysis - Hierarchical indexing, The Panel data

Data Analysis Application Examples: Data munging, Cleaning data, Filtering, Merging data, Reshaping data, Data aggregation, Grouping data

Data Visualization: The matplotlib API primer-Line properties, Figures and subplots, Exploring plot types-Scatter plots, Bar plots, Histogram plots, Legends and annotations, Plotting functions with Pandas

Text Books

1. Programming in Python 3: A Complete Introduction to Python Language, Mark Summerfield, Second Edition, Addison-Wesley Publications

2. Python: End-to-End Data Analysis Learning Path, Module 1: Getting Started with Python Data Analysis , Phuong VothiHong , Martin Czygan, , Packt Publishing Ltd

Reference Books

1. Learning Python, 5th Edition, Mark Lutz, Orielly Publications

2. Python for Data Analysis, Wes McKinney, Orielly Publications

3. How to Think Like a Computer Scientist: Learning with Python 3 Documentation 3rd Edition, Peter Wentworth, Jeffrey Elkner, Allen B. Downey, Chris Meyers

4. Core Python Programming, Second Edition, Wesley J. Chun, Prentice Hall

5. Python Cookbook – Recipes for Mastering Python 3,3rdEdition, David Beazley, Brian K. Jones, Oreilly