MKSSS's Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune)

Curriculum for UG Degree Course in BTech. Computer Engineering

(Academic Year: 2024-25 Onwards)

Second Year | Semester-IV

Course Code	Course Title	Teaching Scheme Hours / Week			Credits	Examination Scheme			Total Marks
		L	T	P		ISE	ESE	Pr/Or	
23PCCE401	Theory Of Computation	3	1	0	**4**	50	50	00	100
23PCCE402	Microprocessor and Microcontroller	2	0	0	**2**	25	25	00	50
23PCCE403	Operating Systems	3	0	0	**3**	50	50	00	100
23CEP401	Community Engagement Project	1	0	2	**2**	50	00	00	50
23MM401	Enterprise Information systems	3	1	0	**4**	50	50	00	100
23VSECCE401	Programming Skills Development Laboratory I	1	0	2	**2**	25	00	25	50
23EEM401	Entrepreneurship Development	3	1	0	**4**	50	50	00	100
23PCCE403L	Operating Systems Laboratory	0	0	2	**1**	25	0	25	50
	Total	**16**	**03**	**06**	**22**	**325**	**225**	**50**	**600**

APPROVED BY
Secretary Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

M.K.S.S.S. Cummins college of Engg. for Women ★
KARVENAGAR
Pune -52

APPROVED BY
Chairman Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

23PCCE401 THEORY OF COMPUTATION

Teaching Scheme	**Examination Scheme**
Lectures: 03 Hours/week	In Semester: 50 Marks
Tutorial : 01 Hours/week	End Semester: 50 Marks
	Credit: 4

Prerequisites

1. Data Structure
2. Discrete Mathematics and Statistics

Course Objectives

To facilitate the learners

1. Recall and understand the basics of mathematical concepts, formal languages and machines.
2. Understand and design different computational models like Finite Automata, Regular Expression, Context Free Grammar for a language.
3. Apply inter conversion between equivalent representations of a language.
4. Design appropriate computational models like Push Down Automata, Turing Machine for a given language.

Course Outcomes:

By taking this course, the learner will be able to

1. Apply the knowledge of basics of mathematics and logic for problem understanding, representation and solving.
2. Construct different computational models like Finite automata, Regular Expression and Context Free Grammar.
3. Evaluate capabilities of computational models by inter-conversion.
4. Design appropriate computational models to solve given problems using Push Down Automata and Turing Machine.

Unit I: Introduction to Formal Languages and Finite Automata

Finite and infinite set. Basic concepts of symbol, alphabet, Kleene Closure and positive Closure of Alphabet, Strings, Empty String, Substring of a string, Concatenation of strings, Formal Language Definition, Finite representation of languages. Concept of Basic Machine and Finite State Machine. Finite Automata (FA): Deterministic FA, Non-deterministic FA and Є-NFA Formal Definitions. Construction of FA (DFA, NFA, Є-NFA) - Transition Function and language

acceptance, Transition graph. Conversion of NFA with Є moves to NFA without Є moves, Conversion of NFA without Є moves to DFA, Direct Conversion of NFA with Є to DFA.
FA with output: Moore and Mealy machines -Definition, models, inter-conversion.

Unit II: Regular Expressions and Languages

Regular Expression (RE): definition and operators, Primitive Regular Expressions, Algebraic Laws of Regular Expressions, Languages Defined by Regular Expressions, Building Regular Expressions, Closure Properties of Regular Languages, Regular expression examples. Inter-conversion of RE and FA, Construction of FA equivalent to RE (RE to Є-NFA, Є-NFA to DFA). Construction of RE equivalent to FA using Arden's Theorem. Pumping Lemma for Regular languages, Limitations of FA. Application of RE in text search and replace.

Unit III: Introduction to Context Free Grammar and Languages

Grammar: Definition, representation of grammar. Context Free Grammar (CFG) - Definition, Derivation – Leftmost, Rightmost, sentential form, parse tree, ambiguous grammar and removing ambiguity from grammar, Simplification of CFG, Normal Forms - Chomsky normal form, Greibach normal form, Closure properties of Context Free Languages (CFL), Decision properties of CFL, Chomsky hierarchy. Regular grammar- Definition, left linear, right linear grammar, Applications of grammar - Palindromes, Parenthesis Match.

Unit IV: Push Down Automata

Push Down Automata (PDA): Definition, Notations, Transition Table form, Types of PDA (Deterministic PDA and Non Deterministic PDA), acceptance by final state, acceptance by empty stack, Construction of PDA (DPDA, NPDA), Instantaneous Description of PDA. Equivalence of PDA and CFG - Grammar to PDA conversion, Applications of PDA - parsing.

Unit V: Turing Machine and Computability Theory

Turing machine (TMs): Formal Definition, TM Instantaneous Description, Transition Function, Languages of TM, Turing Machine and halting, Deterministic Turing Machines (DTM), Construction of DTM. Universal Turing Machine (UTM), Church-Turing hypothesis, Comparison between FA, PDA and TM. Turing Machine Halting Problem. TM's as acceptors, Recognizing Languages with TM's. Recursive and Recursively Enumerable Languages. Decidable Problems and Undecidable Problems, Church-Turing Thesis. Reducibility: Undecidable Problems that are recursively enumerable, A Simple Undecidable.

Text Books

1. Hopcroft J., Motwani R., Ullman J., "Introduction to Automata Theory, Languages andComputations", Third edition, 2008, Pearson Education Asia. ISBN: 9788131720479.
2. Michael Sipser, "Introduction to The Theory of Computation", Third edition, 2017 Thomson Course Technology, ISBN: 9781131525296.

Reference Books

1. Daniel Cohen., "Introduction to Computer Theory", Second edition, 2011, Wiley Publications (India) ISBN: 9788126513345.
2. H.R. Lewis, C. H. Papadimitriou, "Elements of the Theory of Computation", Second edition, 2006, Prentice Hall Inc. ISBN: 8131703878.
3. John C Martin. "Introduction to Language and Theory of Computation", Third edition, 2012, Tata McGraw- Hill, ISBN: 978007660489.
4. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
5. Vivek Kulkarni, "Theory of Computation", Oxford University Press, ISBN 0-19-808458

e-Books

1. https://cglab.ca/~michiel/TheoryOfComputation/TheoryOfComputation.pdf
2. https://www.cs.virginia.edu/~robins/Sipser_2006_Second_Edition_Problems.pdf

Mooc Course - Theory of Computation - IIT Kanpur

https://nptel.ac.in/courses/106104148

Suggestive List of Tutorials

1. Design of Finite state machine.
2. Design Deterministic Finite Automata.
3. NFA design and NFA to DFA conversion.
4. Design of Regular Expression from Language.
5. Converting RE to NFA with null moves and then NFA with null moves to NFA without null moves.
6. Formal language and CFG interconversion.
7. Simplification / standardization of CFG to Normal Forms.
8. Design of Push Down Automata.
9. Design of Turing Machine.

23PCCE402 MICROPROCESSOR AND MICROCONTROLLER

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hours/week	In-Semester: 25 Marks
	End-Semester: 25 Marks
	Credits: 2

Prerequisites

1. Digital Systems and Computer Organization

Course Objectives

To facilitate the learners to-

1. Understand the basic architecture and programming of Intel Pentium processor.
2. Understand the advanced features of the Intel Pentium processor.
3. Understand the architecture and programming of an Arduino microcontroller.
4. Understand the basic concepts of Embedded Systems and Internet of Things (IoT).

Course Outcomes

By taking this course, the learner will be able to -

1. Illustrate the knowledge of basic Intel Pentium processor.
2. Construct simple assembly language programs using x86 instructions.
3. Identify the role and function of Arduino microcontroller components.
4. Relate the concepts of microprocessor and microcontroller to embedded systems and Internet of Things applications.

Unit I: Intel Pentium Microprocessor Architecture

Introduction to Intel Pentium Architecture in detail, Instruction execution pipeline and stages, Memory management using segmentation and paging, Introduction to various modes of Pentium (Real, Protected, Virtual), Interrupts.

Unit II: Assembly Language Programming

Addressing Modes and Instruction set of Pentium, Data types, Introduction to Assembly Language Programming, Use of system calls, Code snippets.

Unit III: Introduction to Microcontroller

Microprocessor vs Microcontroller, Introduction to Arduino Microcontroller, Components and Programming for Arduino, Introduction to Arduino Simulator, Basics of Interfacing - General hardware interfacing like LED's, Sensors etc.

Unit IV: Embedded Systems and IoT

Introduction to Embedded Systems, System on Chip concept, Components, Applications, Introduction to Internet of Things, Characteristics, Vision of IoT, IoT Ecosystem, Applications.

Manuals

1. Pentium Architecture – Intel Manual.
2. Arduino UNO microcontroller Datasheet.

Text Books

1. James Antonakos, "The Pentium Microprocessor", Pearson Education, Revised 2nd Edition, 2006, ISBN: 9788177582765.
2. Sivarama P. Dandamudi, "Introduction to Assembly Language Programming For Pentium and RISC Processors", Springer, 2nd Edition, 2003.
3. Arshdeep Bahga, Vijay Madisetti, "Internet of Things: A Hands-On Approach", Universities Press 2014, ISBN: 9788173719547.

Reference Books

1. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, 3rd Edition, 2008.
2. Rajkamal, "Internet of Things: Architecture and Design Principles", 2018, McGraw Hill Education (India) Private Limited, ISBN: 9789352605224.
3. Raj Kamal, "Embedded Systems- Architecture, Programming and Design", 2017, 3rd Edition, McGraw Hill Education (India) Private Limited, ISBN :978-9332901490.

Web References

1. NPTEL series – https://onlinecourses.nptel.ac.in/noc20_ee42/preview
2. Course on Arduino Microcontroller Programming- https://onlinecourses.swayam2.ac.in/aic20_sp04/preview

23PCCE403 OPERATING SYSTEMS

Teaching Scheme	Examination Scheme
Lectures: 3 Hours/Week	In Semester: 50 Marks
	EndSemester: 50 Marks
	Credits: 3

Prerequisites

1. Digital Systems and Computer Organization.
2. Data Structure.

Course Objectives

To facilitate the learner to

1. Understand basic concepts of Operating Systems.
2. Understand process life-cycle and CPU scheduling algorithms.
3. Apply memory management strategies.
4. Understand file System concepts, protection and security.
5. Learn operating systems for managing resources such as I/O, CPU, memory etc.
6. Understand Inter-process Communication and deadlock concepts.

Course Outcomes

By taking this course, the learner will be able to

1. Build the basic knowledge of operating systems.
2. Make use of the CPU Scheduling Algorithms and Memory Management strategies.
3. Apply the file management concepts along with protection and security features.
4. Make use of the knowledge of storage devices for disk management.
5. Apply the concepts of Inter-process Communication.

Unit I: Introduction to Operating Systems

Introduction to Operating System (OS), Functions of OS, Types of OS, OS Concepts: Process, Files, Shell and its types, System calls, Kernel and its types, Introduction to Virtual Machine, Case Study of introduction to UNIX Operating System.

UnitII: Process and CPU Scheduling

Process Concept, Operations On Processes : Creation and Termination, States Transition and Context Switching, Scheduling Criteria, Scheduling Algorithm, First-Come First-Serve (FCFS), Shortest Job First (SJF), Round-Robin (RR), Introduction to Threads and Benefits, Comparison of Thread and Process, Case Study of UNIX Process Management.

Unit III: Memory Management

Contiguous and Non-Contiguous Memory, Swapping, Paging, Segmentation, Comparison of Paging and Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms- First-In First-Out (FIFO), Least Recently Used (LRU), Optimal, Trashing, Case Study of UNIX memory management.

Unit IV: File and Directory Management

File Concepts, File Attributes, File Operations, File Types, File Sharing, File Structure, Allocation Methods, Types of Users, Access Modes, Directory Overview, Types of Directories, Introduction to Protection and Security, Case Study of UNIX File Structure.

Unit V: I/O Management and Disk Scheduling

I/O Devices, I/O Buffering, Disk Scheduling - First Come-First Serve (FCFS), SCAN, Circular SCAN (C- SCAN), Shortest Seek Time First (SSTF).

Unit VI: Inter-Process Communication (IPC)

Critical Section Problem, Semaphores, Classical Problems of Synchronization, Deadlocks, Methods of Handling Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

Text Books

1. Silberschatz, Galvin, Gagnes, "Operating System Concepts", John Wiley & Sons, (8/e), ISBN: 9971- 51-388-9.
2. William Stallings, "Operating System-Internals and Design Principles ", Prentice Hall India, (5/e) ISBN: 81-297-0 1 094-3.
3. Andrew S. Tanenbaum, "Modern Operating Systems", Prentice Hall India, (4/e), ISBN: 81-203- 2063.

Reference Books

1. Evi Nemeth, Garth Snyder, Tren Hein, Ben Whaley, “Unix and Linux System Administration Handbook”, (4/e), ISBN: 978-81-317-6177-9. (2011).
2. Milan Milenkovic, “Operating Systems”, TMH, (2/e), ISBN: 0-07-044700-4.
3. Maurice J. Bach, “The Design of the Unix Operating System”, Pearson Education, ISBN: 81-7758- 770-6.

Web References

1. NPTEL: Computer Science and Engineering: Introduction to Operating Systems: https://nptel.ac.in/courses/106/106/106106144/
2. NPTEL: Computer Science and Engineering: Operating System Fundamentals: https://nptel.ac.in/courses/106/105/106105214/

MKSSS's Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune)

23PCCE403L OPERATING SYSTEMS LABORATORY

Teaching Scheme

Practical: 2 Hours/Week

Examination Scheme

In Semester: 25 Marks
End Semester: 25 Marks
Credit: 1

Prerequisites

1. Programming Skills in Java Language.
2. Data Structure Laboratory.

Course Objectives

To facilitate the learner to

1. Understand the fundamentals of Operating Systems.
2. Learn the concepts of Process management and CPU scheduling.
3. Apply the concepts of Operating System for Memory management.
4. Understand the operations performed by the Operating System as a resource manager.
5. Understand the communication among the processes.

Course Outcomes

After completion of the course, students will be able to

1. Implement basic UNIX/Linux Commands and Shell commands.
2. Make use of different CPU scheduling algorithms.
3. Apply Memory Management algorithms.
4. Implement various disk scheduling algorithms.
5. Explore the Inter-Process Communication concepts.

Preamble

Operating Systems Laboratory (PCCCE403L) is designed to understand and implement the fundamental concepts of Operating Systems. Motivation here is that students should be able to apply various Unix/Linux commands and shell commands. Also students should make use of various Operating system services algorithms. Students can use Java/Python/C programming languages to implement these assignments. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it enhances the concepts and logic of the solution. Students will be encouraged to solve open ended problems in different operating systems. Faculty will appropriately adopt assignments on similar lines as the examples shown here.

Group A assignments are on Unix/Linux commands, shell commands and simulation of schedulers, memory and I/O management algorithms. Group B assignments are on Inter-process communication problems using Semaphores and Group C assignment is on case study for different Operating Systems and its services.

Suggestive List of Assignments:

Group A : (Mandatory)

1. Demonstration of Installation of Linux Operating System.
2. Exploration of Unix/Linux (File, Directory and Process commands) and shell programming commands.
3. Simulation of the scheduling algorithms. For example: First Come First Serve (FCFS), Shortest Remaining Time First (SRTF), Round-Robin (RR), Shortest Job First (SJF).
4. Simulation of memory allocation strategies. For example: First Fit, Best Fit and Worst Fit.
5. Simulation of Page replacement algorithms. For example: First-In-First-Out (FIFO), Least Recently Used (LRU), Optimal page replacement.
6. Simulation of Disk scheduling algorithms. For example: First Come First Serve (FCFS), SCAN, Circular – SCAN (C-SCAN), Shortest Seek Time First (SSTF).

Group B: **(Any Two)**

7. A single processor system has three resource types X, Y and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column alloc denotes the number of units of each resource type allocated to each process, and the column request denotes the number of units of each resource type requested by a process in order to complete execution. Which of these processes will finish LAST?

	Alloc			**Request**		
	X	Y	Z	X	Y	Z
P0	1	2	1	1	0	3
P1	2	0	1	0	1	2
P2	2	2	1	1	2	0

8. An operating system uses the banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y and Z to three processes P0, P1 and P2. The table given below presents the current system state. Here, the Allocation matrix shows the

current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

	Allocation			**Max**		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in safe state. Consider the following independent requests for additional resources in the current state-

REQ1: P0 requests 0 units of X, 0 units of Y and 2 units of Z

REQ2: P1 requests 2 units of X, 0 units of Y and 0 units of Z

Check whether these requests will be granted or not?

9. Consider a situation where a group of people are sharing a common file. In this case, if one person tries editing the file, no other person should be reading or writing at the same time, otherwise changes will not be visible to him/her. However, if some person is reading the file, then others may read it at the same time. Write a program to implement Reader-Writer problem using semaphores.

10. Consider there are two types of processes, the producer and the consumer that share a common fixed-size buffer. The producer's job is to generate data, put it into the buffer, and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer), one piece at a time. Given the common fixed-size buffer, the task is to make sure that the producer can't add data into the buffer when it is full and the consumer can't remove data from an empty buffer. Accessing memory buffers should not be

allowed to producers and consumers at the same time. Implement producer-consumer problem using Semaphores.

11. Write a program to implement operations on processes using fork and join system calls.

Group C

12. Case study of various Operating systems services. (Example: Android, RTOS, Linux, IOS, Windows etc.)

23MM401 ENTERPRISE INFORMATION SYSTEMS

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 Hours/week	In-Semester: 50 Marks
Tutorial: 1 Hour/week	End-Semester: 50 Marks
	Credits: 4

Course Objectives

To facilitate the learners to-

1. Understand the basics of business models.
2. Learn operational and strategic systems and its applicability.
3. Explore enterprise systems social aspect.
4. Understand various advancements in information systems.

Course Outcomes

By taking this course, the learner will be able to -

1. Apply business models to a given product/scenario.
2. Make use of operational and strategic systems to business process automation.
3. Understand the corporate social responsibility perspective to the enterprise system.
4. Apply ESG concepts to information systems.
5. Illustrate various trends in Enterprise Information Systems.

Unit I: Introduction to Enterprise Information Systems, Challenges and Issues

Evolution of Enterprise Information Systems (EIS), Enterprise data model, different types of business models, challenges and issues involved, Enterprise Business Process Model, Categories of business processes, Types of systems: operational system and strategic system, Examples of Enterprise systems: Enterprise resource planning, Supply chain management, Customer relationship management, enterprise intelligence etc.Case study on B2B governance.

Unit II: Business Process Automation

Concept of business plan, comparison of business model and business plan, Factors affecting business process automation, Benefits of Automating Business Process, importance of EIS in business process automation, steps involved in Implementing Business Process Automation, Business process modeling notations, Diagrammatic Representation of Business Processes Enterprise Risk management and its benefits
Case study: Automation of Employee Attendance.

Unit III: Sustainability Issues

Sustainable information systems, Corporate Social Responsibility (CSR) and EIS, social and ethical issues in EIS, sustainable CSR, social enterprises in CSR.

Unit IV: Environmental, Social and Governance (ESG) Issues

Concept of ESG, ESG standards and their evolution, CSR to ESG , Enterprise architecture for ESG, Sustainable enterprise architecture, ESG and risk management.

Unit V: Trends, Tools and Technologies

Enterprise Application Interface (EAI), Artificial Intelligence enabled Customer Relationship Management, Digital India Initiative, Case study on EIS-5G impact, Use of information systems in areas like sales, finance.

Text books

1. Board Of Studies, The Institute Of Chartered Accountants Of India, "Enterprise Information Systems".
2. Information Resources Management Association, USA, "Enterprise Information Systems: Concepts, Methodologies, Tools and Applications".
3. Graham Friend and Stefan Zehle, "Guide To Business Planning".

Reference books

1. Belanger et.al."Information Systems for Business: An Experiential Approach" Edition 4.0
2. David T. Bourgeois, "Information Systems for Business and Beyond".
3. Lorenz M. Hilty, Eberhard K. Seifert, René Treibert, "Information systems for sustainable development".
4. Marlon Dumas Marcello La Rosa Jan Mendling Hajo A. Reijers, "Fundamentals of Business Process Management".

Web references

1. **NPTEL:** https://onlinecourses.nptel.ac.in/noc20_mg60/preview

Suggestive list of tutorials

- System requirement analysis for the given business use case.
- Develop a digital marketing and sales strategy for the given scenario.
- Generate suitable data model for a given scenario for example chair manufacturing for elderly people.

- Group discussion on data modeling best practices.
- Generate a business plan design for a given scenario (9 steps of designing business plan) eg. for a new/refurbished laptop launch.
- Group discussions on social and ethical issues in various types of business models.
- Develop an enterprise architecture by identifying its components like business architecture, information system architecture, technology architecture.
- Develop a product design strategy in operation management for a given situation eg. product design strategy for the use of mobile technology for older people.
- Suggest a suitable business model for a given scenario such as adjusting the business model for a major targeted customer chunk.
- Demonstration on CRM, healthcare enterprise such as OMEGA financial, salesforce.

MKSSS's Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune)

23VSECCE401 PROGRAMMING SKILLS DEVELOPMENT LABORATORY I

Teaching Scheme	**Examination Scheme**
Lecture: 1 Hrs/week	In semester: 25 Marks
Practical: 2 Hrs/week	End semester: 25 Marks
	Credit: 4

Prerequisites:

1. Programming Skills in C Language.
2. Programming Skills in Java Language.

Course Objectives

To facilitate the learners to

1. Understand the fundamentals of Python programming.
2. Learn the multi paradigm features of Python.
3. Make use of the data handling and visualization concepts in Python.
4. Learn the Python database connectivity

Course Outcomes

By taking this course, the learner will be able to

1. Make use of the various programming constructs in Python.
2. Implement the multi paradigm features of Python.
3. Implement data handling and visualization in Python.
4. Build a database application using Python.

UNIT I: Introduction to Python

Input-output statements, comments, formatting the output, execution of a Python program. Variables (concept of dynamic typing), control statements, string handling, defining functions, returning multiple values from a function. Datatypes in Python: Built-in datatypes, List, Tuple, Range, Sets, Dictionary, and their functions.

UNIT II: Object Oriented and Functional Programming in Python

Object Oriented Programming: Defining a class with data members, constructor, and methods. The "self" variable, Creating objects, Class variables, Inheritance, Abstract classes, and Interfaces, Exception handling.

Functional Programming: Introduction to the functional programming paradigm, Programming without side effects. Treating functions as first class arguments, map(), filter() and reduce() in Python.

UNIT III: Data Handling and Visualization

Reading datasets from a .csv or .xlsx file using the pandas library. Data analysis using the numpy library. Graphical representation of data using Matplotlib and seaborn libraries in Python. Reading and manipulating image files using the PIL and open-cv libraries of Python.

UNIT IV: Python's Database Connectivity and Graphical User Interface

Concept of database driver, connection, result-set, statement, and prepared statement in Python. Creating a graphical User Interface using the tkinter library. Concept of root window, title, Frame and the different widgets supported by Python to create a graphical User Interface.

Suggestive List of Assignments

The following list of assignments is suggestive, and the faculty is encouraged to define the problem statements on the similar lines so that the students can implement maximum features of the Python programming language.
Group A assignments are based on the control structures and datatypes in Python and the object-oriented features of Python.
Group B assignments are based on file handling, database connectivity and building graphical user interface in Python.
Group C assignment is an open-ended assignment where students will define a problem statement for a real-world application and implement it as a mini project. This is a group activity to be done in teams of 3-4 students.

Group A (Mandatory)

1. Assignment to implement the Collatz conjecture for a given positive integer n.

2. Assignment to explore Lists and Strings in Python.
   - Create a list of numbers and
     - Append an element to the list.
     - Insert an element in the given position.
     - Remove an element from the list.
     - Remove all odd numbers from the list.
     - Print the largest and smallest number from the list.
     - Print the second largest number from the list.
     - Concatenate two lists.

  - Reverse the list.
  - Replace every element by its square.
  - Print the numbers from the list which are from the given range.
  - Create a copy of the given list.
  - Find the difference between two lists.
  - Remove repeating elements from the list.
  - Use the list as a matrix and perform addition of 2 matrices.
- Consider a list of words and find the words which are longer than 'n'

3. Assignment to explore Dictionaries in Python.
Part 1 - Make use of Dictionaries to create a contact list (store the contact name and contact number). Perform validations like: phone number should start with 9/8/7 and phone number should be of 10 digits. Perform operations like search, add, delete and modify a contact.
Part 2 - Simultaneously create a log record to keep track of the operations on the contact list. (record the date and time of the update along with the operation – add/delete/modify). Hint: Make use of list to store the log records.

4. Assignment to explore tuples in Python.
Read data from the excel file and print it on the Python console.

5. Assignment to implement Object oriented Programming in Python
Part 1 – Create a class "CAR" with suitable attributes. Create a list of objects and perform following operations – add, modify, delete, search, and display all cars.
Part 2 – Create a derived class "SUV" with suitable attributes. Create a list of objects and perform following operations – add, modify, delete, search and display all SUVs.

Group B (any 3)

1. Choose any dataset and perform the following:
    a. Load the dataset into a Pandas DataFrame
    b. Perform exploratory data analysis to understand the structure and characteristics of the dataset.
    c. Generate appropriate visualizations to understand the correlation between the attributes and to understand the patterns in the data.

2. Assignment to implement Database connectivity in Python.
   Access database in MySQL table and perform following operations on a table – Display, search, add a record, delete a record, Update a record.

3. Write a lambda function for mathematical operations like addition, subtraction, multiplication and square and use the map() function to apply each lambda function to a list of numbers.

4. Write a Python program to print the first n rows of Pascal's Triangle.

5. Assignment to build graphical user interface in Python.
   Use Tkinter to develop applications like:
   - Currency converter/ Weight converter (Kgs to lbs)
   - Text editor
   - Calculator

6. Assignment to implement the file handling in Python.

Perform file operations: create a file, read from a file, write in a file, close the file, copy contents of one file into another.

7. Implement the following operations on Strings (make use of in-built functions wherever possible):
   a. Find the length.
   b. Reverse the string.
   c. Concatenation of two strings.
   d. Compare two strings.
   e. Check if Substring is present.
   f. Convert the string to upper case.
   g. Convert the string to lower case.
   h. Count the number of occurrences of a character in a string.
   i. Count the number of occurrences of a substring in the string.
   j. Replace a character in the string.
   k. Print the position of a character in the string.
   l. Accept a sentence from the user and compute the length of each word in that sentence.

Group C: Mini Project

Students can choose any problem statement of their choice and build Python application for the same.

For example:

1. Employee management system.
2. Ticket booking system.
3. Rock-Paper-Scissor game.
4. Snakes game.
5. Memory game.

Text Books

1. Kenneth. A. Lambert, “Fundamentals of Python First Programs”, Cengage, 2nd Edition, 2019.
2. Vamsi Kurama, “Python Programming: A Modern Approach”, Pearson, 1stEdition, 2017.

Reference Books

1. Gowrishankar.S, Veena A, “Introduction to Python Programming”, CRC Press, Paperback Edition, 2019.
2. Y. Daniel Liang, “Introduction to Programming Using Python”, Pearson, Paperback Edition, 2017.
3. Dr. R. Nageshwara Rao, “Core Python Programming”, DreamTech Press, 2nd Edition, 2018.

e-Resources

1. https://www.tutorialspoint.com/python3/python_tutorial.pdf.