Study & Evaluation Scheme with Syllabus

For

B.Tech. Second Year

(Computer Science and Engineering, Computer Engg. & Information Technology)

On

Choice Based Credit System

(Effective from the Session: 2017-18)
## THIRD SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Subject Name</th>
<th>L-T-P</th>
<th>Th/Lab ESE</th>
<th>Sessional CT</th>
<th>Th/Lab TA</th>
<th>Total</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RAS301/ROE030, 032 to 037, 039</td>
<td>Mathematics-III/ Science Based OE</td>
<td>3-1-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>RVE301/ RAS302</td>
<td>Universal Human Values &amp; Professional Ethics / Environment &amp; Ecology</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>REC301</td>
<td>Digital Logic Design</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>RCS301</td>
<td>Discrete Structures &amp; Theory of Logic</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>RCS302</td>
<td>Computer Organization and Architecture</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>RCS303</td>
<td>Data Structures</td>
<td>3-1-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>RCS351</td>
<td>Digital Logic Design Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>RCS352</td>
<td>Discrete Structure &amp; Logic Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>RCS353</td>
<td>Computer Organization Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>RCS354</td>
<td>Data Structures Using C / Java Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>1000</strong></td>
<td><strong>24</strong></td>
</tr>
</tbody>
</table>

### Science Based Open Electives:

1. ROE030/040 Manufacturing Process
2. ROE032/042 Nano Science
3. ROE033/043 Laser System and Application
4. ROE034/044 Space Science
5. ROE035/045 Polymer Science & Technology
6. ROE036/046 Nuclear Science
7. ROE037/047 Material Science
8. ROE039/049 Applied Linear Algebra
## FOURTH SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Subject Name</th>
<th>L-T-P</th>
<th>Th/Lab ESE</th>
<th>Sessional CT</th>
<th>TA</th>
<th>Total</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ROE040, 042 to 047, 049/ RAS401</td>
<td>Science Based OE/ Mathematics-III</td>
<td>3-1-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>RAS402/ RVE401</td>
<td>Environment &amp; Ecology/ Universal Human Values &amp; Professional Ethics</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>REC40</td>
<td>Introduction to Microprocessor</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>RCS401</td>
<td>Operating Systems</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>RCS402</td>
<td>Software Engineering</td>
<td>3-0-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>RCS403</td>
<td>Theory of Automata and Formal Languages</td>
<td>3-1-0</td>
<td>70</td>
<td>20</td>
<td>10</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>RCS451</td>
<td>Operating Systems Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>RCS452</td>
<td>Software Engineering Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>RCS453</td>
<td>TAFL Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>RCS454</td>
<td>Python Language Programming Lab</td>
<td>0-0-2</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>1000</strong></td>
<td></td>
<td></td>
<td><strong>24</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Science Based Open Electives:

1. ROE030/040 Manufacturing Process
2. ROE032/042 Nano Science
3. ROE033/043 Laser System and Application
4. ROE034/044 Space Science
5. ROE035/045 Polymer Science & Technology
6. ROE036/046 Nuclear Science
7. ROE037/047 Material Science
8. ROE039/049 Applied Linear Algebra
RCS301: DISCRETE STRUCTURES & THEORY OF LOGIC

Unit-I
Set Theory: Introduction, Combination of sets, Multi sets, ordered pairs, Set Identities.
Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Order of relations.
Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions.
Natural Numbers: Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases.

Unit-II
Algebraic Structures: Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphism’s, Definition and elementary properties of Rings and Fields, Integers Modulo n.

Unit-III
Partial order sets: Definition, Partial order sets, Combination of partial order sets, Hasse diagram.
Lattices: Definition, Properties of lattices – Bounded, Complemented, Modular and Complete Lattice, Morphisms of lattices.
Boolean Algebra: Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra. Combinational and sequential Circuits

Unit-IV
Propositional Logic: Proposition, well formed formula, Truth tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference ,Natural Deduction.
Predicate Logic: First order predicate, well formed formula of predicate, quantifiers, Inference theory of predicate logic.

Unit-V
Trees: Definition, Binary tree, Binary tree traversal, Binary search tree.
Graphs: Definition and terminology, Representation of graphs, Multi graphs, Bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring.
Recurrence Relation & Generating function: Recursive definition of functions, Recursive algorithms, Method of solving recurrences.
Combinatorics: Introduction, Counting Techniques, Pigeonhole Principle

References :
4. R. P. Grimaldi, Discrete and Combinatorial Mathematics, Addison Wesley,
10. Deo N., “Graph Theory with Applications to Engineering and Computer Science”, PHI Learning Private Limited, Delhi India
Unit-I
Functional units of digital system and their interconnections, buses, bus architecture, types of
buses and bus arbitration. Register bus and memory transfer, Processor organization, general
register organization, stack organization and addressing modes, Look ahead carry adders.
Multiplication: Signed operand multiplication, Booths algorithm and array multiplier.
Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit
design.

Unit-II
Instruction types, formats, instruction cycles and sub cycles (fetch, execute etc) , micro-
operations, execution of a complete instruction, Hardwire and micro-programmed control:
micro-programme sequencing, concept of horizontal and vertical microprogramming.

Unit-III
Basic concept and hierarchy, semiconductor RAM memories, 2D & $2^{1/2}$D memory
organization. ROM memories, Cache memories: concept and design issues & performance,
address mapping and replacement, Auxiliary memories: magnetic disk, magnetic tape and
optical disks, Virtual memory: concept implementation.

Unit-IV
Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts
and exceptions, Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct
Memory Access., I/O channels and processors, Serial Communication: Synchronous &
asynchronous communication, standard communication interfaces.

Unit - V
Architectural Classification Schemes, Flynn’s & Feng’s Classification, Performance
Metrics and Measures, Speedup Performance Laws , Pipelining and Memory Hierarchy
Basic and Intermediate Concepts, Linear and Nonlinear Pipeline Processors, Optimization
of Cache Performance.

Reference Books:
Integrated Approach” 2nd Edition
Unit - I
Abstract Data Types (ADT), Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations.

Unit – II

Unit – III

Unit – IV

Unit – V
Searching: Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting.
Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees.
Hashing: Hash Function, Collision Resolution Strategies
Storage Management: Garbage Collection and Compaction.

References:
12. Adam Drozdek “ Data Structures and Algorithm in Java”, Cengage Learning

RCS351: DIGITAL LOGIC DESIGN LAB

Objective: To understand the digital logic and create various systems by using these logics.

1. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.
2. Implementation of the given Boolean function using logic gates in both SOP and POS forms.
3. Verification of state tables of RS, JK, T and D flip-flops using NAND & NOR gates.
4. Implementation and verification of Decoder/De-multiplexer and Encoder using logic gates.
5. Implementation of 4x1 multiplexer using logic gates.
6. Implementation of 4-bit parallel adder using 7483 IC.
7. Design, and verify the 4-bit synchronous counter.
8. Design, and verify the 4-bit asynchronous counter.

RCS352: DISCRETE STRUCTURE & LOGIC LAB

Understanding of mathematical computation software such as Mapple, Prolog to experiment the followings:

1. Working of Computation software
2. Discover a closed formula for a given recursive sequence vice-versa
3. Recursion and Induction: Practice of proof techniques
4. Practice of various set operations
5. Testing of set operating using software
6. Counting
7. Combinatorial equivalence
8. Permutations and combinations
9. Difference between structures, permutations and sets
10. Implementation of a recursive counting technique
11. N digit binary sequences not having adjacent 1’s
12. Probability simulation
13. The Birthday problem
14. Poker Hands problem
16. Comparison of theoretical probability with experimental probability
17. Baseball: Binomial Probability
18. Basketball: One and one
19. Expected value problem
20. Binary relations
RCS353: COMPUTER ORGANIZATION LAB

EXPERIMENTS:-
1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER and Implementing 4x1 and 8x1 MULTIPLEXERS.
4. Verify the excitation tables of various FLIP-FLOPS.
5. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
6. Design of an 8-bit ARITHMETIC LOGIC UNIT.
7. Design the data path of a computer from its register transfer language description.
8. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
9. Write an algorithm and program to perform matrix multiplication of two n * n matrices on the 2-D mesh SIMD model, Hypercube SIMD Model or multiprocessor system.
10. Study of Scalability for Single board Multi-board, multi-core, multiprocessor using Simulator

RCS354: DATA STRUCTURE USING C / JAVA LAB

Program in C or C++ for following:
1. To implement addition and multiplication of two 2D arrays.
2. To transpose a 2D array.
3. To implement stack using array.
4. To implement queue using array.
5. To implement circular queue using array.
6. To implement stack using linked list.
7. To implement queue using linked list.
8. To implement circular queue using linked list.
9. To implement binary tree using linked list.
10. To implement binary search tree using linked list.
11. To implement tree traversals using linked list.
12. To implement BFS using linked list.
13. To implement DFS using linked list.
14. To implement Linear Search.
15. To implement Binary Search.
16. To implement Bubble Sorting.
17. To implement Selection Sorting.
18. To implement Insertion Sorting.
19. To implement Merge Sorting.
20. To implement Heap Sorting.
UNIT I
Introduction to Microprocessor, Microprocessor architecture and its operations, Memory, Input & output devices, Logic devices for interfacing, The 8085 MPU, Example of an 8085 based computer, Memory interfacing.

UNIT II
Basic interfacing concepts, Interfacing output displays, Interfacing input devices, Memory mapped I/O, Flow chart symbols, Data Transfer operations, Arithmetic operations, Logic Operations, Branch operation, Writing assembly language programs, Programming techniques: looping, counting and indexing.

UNIT III
Additional data transfer and 16 bit arithmetic instruction, Arithmetic operations related to memory, Logic operation: rotate, compare, counter and time delays, Illustrative program: Hexadecimal counter, zero-to-nine, (module ten) counter, generating pulse waveforms, debugging counter and time delay, Stack, Subroutine, Restart, Conditional call and return instructions, Advance subroutine concepts, The 8085 Interrupts, 8085 vector interrupts.

UNIT IV
Program: BCD-to-Binary conversion, Binary-to-BCD conversion, BCD-to-Seven segment code converter, Binary-to-ASCII and ASCII-to-Binary code conversion, BCD Addition, BCD Subtraction, Introduction to Advance instructions and Application, Multiplication, Subtraction with carry.

UNIT V
8255 Programmable peripheral interface, interfacing keyboard and seven segment display, 8254 (8253) programmable interval timer, 8259A programmable interrupt controller, Direct Memory Access and 8237 DMA controller.
Introduction to 8086 microprocessor: Architecture of 8086 (Pin diagram, Functional block diagram, Register organization).

References:
Unit – I

Unit – II

Unit – III

Unit – IV
Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

Unit – V
I/O Management and Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, File system implementation issues, File system protection and security.

References:
2. Andrew S. Tanenbaum, “Modern Operating System”, PHI Learning
RCS402: SOFTWARE ENGINEERING

Unit-I: Introduction

Unit-II: Software Requirement Specifications (SRS)


Unit-III: Software Design

Unit-IV: Software Testing

Unit-V: Software Maintenance and Software Project Management

References:
2. Pankaj Jalote, Software Engineering, Wiley
8. P Fleeger, Software Engineering, Macmillan Publication
RCS403: THEORY OF AUTOMATA AND FORMAL LANGUAGES

Unit – I
Introduction; Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill-Nerode Theorem

Unit – II
Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen’s Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages. Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

Unit – III
Context free grammar (CFG) and Context Free Languages (CFL): Definition, Examples, Derivation, Derivation trees, Ambiguity in Grammar, Inherent ambiguity, Ambiguous to Unambiguous CFG, Useless symbols, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure proper ties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs.

Unit – IV
Push Down Automata (PDA): Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA

Unit – V
Turing machines (TM): Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computerof Integer functions, Universal TM, Church’s Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TMs. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory

References:
1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation”, Pearson Education.
5. Malviya ,AK "Theory of Computation and Application ” BPaperback Publications
7. K. Krithivasan and R. Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.
RCS451: OPERATING SYSTEMS LAB

1. To implement CPU Scheduling Algorithms
   - FCFS
   - SJF
   - SRTF
   - PRIORITY
   - ROUND ROBIN

2. Simulate all Page Replacement Algorithms
   - FIFO
   - LRU

3. Simulate Paging Technique of Memory Management
   Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

RCS452: SOFTWARE ENGINEERING LAB

For any given case/problem statement do the following:
1. Prepare a SRS document in line with the IEEE recommended standards.
2. Draw the use case diagram and specify the role of each of the actors. Also state the precondition, post condition and function of each use case.
3. Draw the activity diagram.
4. Identify the classes. Classify them as weak and strong classes and draw the class diagram.
5. Draw the sequence diagram for any two scenarios.
6. Draw the collaboration diagram.
7. Draw the state chart diagram.
8. Draw the component diagram.
9. Perform forward engineering in java. (Model to code conversion)
10. Perform reverse engineering in java. (Code to Model conversion)
11. Draw the deployment diagram.

RCS453: TAFL Lab

Understanding of software like JFLAP for experimenting with formal languages
1. Deterministic Finite Automata (DFA)
2. Nondeterministic Finite Automata (NFA)
3. Conversion of NFA to DFA
4. DFA Minimization
5. DFA to regular grammar conversion
6. DFA to regular expression conversion
7. Combining automata
8. Regular expression to DFA conversion
9. Mealy and Moore machine
10. Pushdown automata
11. Single tape Turing machine
12. Multi-tape Turing machine
13. Context free grammars (CFG) with single symbols
14. CFG with multiple symbols
15. LL Parsing
16. LR Parsing
17. Regular expressions
18. Regular pumping lemma
19. Context free pumping lemma
20. CFG to Chomsky Normal form transformation

RCS454: PYTHON LANGUAGE PROGRAMMING LAB

Write a Python program to:

1. Demonstrate the working of ‘id’ and ‘type’ functions
2. To find all prime numbers within a given range.
3. To print ‘n’ terms of Fibonacci series using iteration.
4. To demonstrate use of slicing in string
5. a. To add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.
   Sample String : 'abc'
   Expected Result : 'abcing'
   Sample String : 'string'
   Expected Result : 'stringly'
   b. To get a string from a given string where all occurrences of its first char have been changed to 'S', except the first char itself.
6. a. To compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.
   b. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.
7. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.
8. To demonstrate use of list & related functions
9. To demonstrate use of Dictionary & related functions
10. To demonstrate use of tuple, set & related functions
11. To implement stack using list
12. To implement queue using list
13. To read and write from a file
14. To copy a file
15. To demonstrate working of classes and objects
16. To demonstrate class method & static method
17. To demonstrate constructors
18. To demonstrate inheritance
19. To demonstrate aggregation/composition
20. To create a small GUI application for insert, update and delete in a table using Oracle as backend and front end for creating form

The lab experiments for this course have to ensure that the following concepts of PYTHON LANGUAGE are covered during lab classes:

Installing Python; basic syntax, interactive shell, editing, saving, and running a script, the concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; reading input from console, writing to console, comments in the program; understanding error messages; Conditions, Boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while);

String manipulations: subscript operator, indexing, slicing a string; other functions on strings: string module, strings and number system, format functions: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers

Lists, tuples, sets, and dictionaries: basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries, Array in Python

Regular Expressions: re modules, match function, search function, modifiers and patterns

Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments. Program structure and design. Recursive functions, scope and global statements, Lambda expressions, Importing Modules, math Module & Random Modules, creating own module.

Exception Handling: Exceptions, except clause, try and finally clause user defined exceptions

File Handling: manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file;

Simple Graphics: “turtle” module; simple 2d drawing - colors, shapes; digital images, image file formats. Graphical user interfaces: event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames

Database: cx_ Oracle module, Connections, Executing Queries, calling procedure and functions, Using GUI to access Database

Object Oriented Programming: Concept of OOP: Abstraction, Encapsulation, Inheritance, and Polymorphism in Python, classes, objects, attributes and methods; defining classes; design with classes, constructors and destructors, inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; aggregation and composition

The reference books for this lab course are suggested as below:

5. Mark Pilgrim, “Dive into Python”, Apress
7. Y. Daniel Liang “Introduction to Programming using Python” Pearson