

binils.com  
Anna University, Polytechnic, Schools  
SEMESTER III

S. NO.	COURSE CODE	COURSE TITLE	CATE GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
<b>THEORY</b>								
1.	MA3354	Discrete Mathematics	BSC	3	1	0	4	4
2.	CS3351	Digital Principles and Computer Organization	ESC	3	0	2	5	4
3.	CS3352	Foundations of Data Science	PCC	3	0	0	3	3
4.	CS3301	Data Structures	PCC	3	0	0	3	3
5.	CS3391	Object Oriented Programming	PCC	3	0	0	3	3
<b>PRACTICALS</b>								
6.	CS3311	Data Structures Laboratory	PCC	0	0	3	3	1.5
7.	CS3381	Object Oriented Programming Laboratory	PCC	0	0	3	3	1.5
8.	CS3361	Data Science Laboratory	PCC	0	0	4	4	2
9.	GE3361	Professional Development <sup>§</sup>	EEC	0	0	2	2	1
<b>TOTAL</b>				<b>15</b>	<b>1</b>	<b>14</b>	<b>30</b>	<b>23</b>

<sup>§</sup> Skill Based Course

SEMESTER IV

S. NO.	COURSE CODE	COURSE TITLE	CATE GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
<b>THEORY</b>								
1.	CS3452	Theory of Computation	PCC	3	0	0	3	3
2.	CS3491	Artificial Intelligence and Machine Learning	PCC	3	0	2	5	4
3.	CS3492	Database Management Systems	PCC	3	0	0	3	3
4.	CS3401	Algorithms	PCC	3	0	2	5	4
5.	CS3451	Introduction to Operating Systems	PCC	3	0	0	3	3
6.	GE3451	Environmental Sciences and Sustainability	BSC	2	0	0	2	2
7.		NCC Credit Course Level 2 <sup>#</sup>		3	0	0	3	3 <sup>#</sup>
<b>PRACTICALS</b>								
8.	CS3461	Operating Systems Laboratory	PCC	0	0	3	3	1.5
9.	CS3481	Database Management Systems Laboratory	PCC	0	0	3	3	1.5
<b>TOTAL</b>				<b>20</b>	<b>0</b>	<b>10</b>	<b>30</b>	<b>22</b>

<sup>#</sup> NCC Credit Course level 2 is offered for NCC students only. The grades earned by the students will be recorded in the Mark Sheet, however the same shall not be considered for the computation of CGPA.

MA3354

DISCRETE MATHEMATICS

L T P C  
3 1 0 4

**COURSE OBJECTIVES:**

- To extend student's logical and mathematical maturity and ability to deal with abstraction.
- To introduce most of the basic terminologies used in computer science courses and application of ideas to solve practical problems.
- To understand the basic concepts of combinatorics and graph theory.
- To familiarize the applications of algebraic structures.
- To understand the concepts and significance of lattices and boolean algebra which are widely used in computer science and engineering.

**UNIT I LOGIC AND PROOFS 9+3**

Propositional logic – Propositional equivalences - Predicates and quantifiers – Nested quantifiers – Rules of inference - Introduction to proofs – Proof methods and strategy.

**UNIT II COMBINATORICS 9+3**

Mathematical induction – Strong induction and well ordering – The basics of counting – The pigeonhole principle – Permutations and combinations – Recurrence relations – Solving linear recurrence relations – Generating functions – Inclusion and exclusion principle and its applications.

**UNIT III GRAPHS 9+3**

Graphs and graph models – Graph terminology and special types of graphs – Matrix representation of graphs and graph isomorphism – Connectivity – Euler and Hamilton paths.

**UNIT IV ALGEBRAIC STRUCTURES 9+3**

Algebraic systems – Semi groups and monoids - Groups – Subgroups – Homomorphism's – Normal subgroup and cosets – Lagrange's theorem – Definitions and examples of Rings and Fields.

**UNIT V LATTICES AND BOOLEAN ALGEBRA 9+3**

Partial ordering – Posets – Lattices as posets – Properties of lattices - Lattices as algebraic systems – Sub lattices – Direct product and homomorphism – Some special lattices – Boolean algebra – Sub Boolean Algebra – Boolean Homomorphism.

**TOTAL: 60 PERIODS**

**COURSE OUTCOMES:**

At the end of the course, students would :

**CO1:**Have knowledge of the concepts needed to test the logic of a program.

**CO2:**Have an understanding in identifying structures on many levels.

**CO3:**Be aware of a class of functions which transform a finite set into another finite set which relates to input and output functions in computer science.

**CO4:**Be aware of the counting principles.

**CO5:**Be exposed to concepts and properties of algebraic structures such as groups, rings and fields.

**TEXT BOOKS:**

1. Rosen. K.H., "Discrete Mathematics and its Applications", 7<sup>th</sup> Edition, Tata McGraw Hill Pub. Co. Ltd., New Delhi, Special Indian Edition, 2017.
2. Tremblay. J.P. and Manohar. R, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill Pub. Co. Ltd, New Delhi, 30<sup>th</sup> Reprint, 2011.

**REFERENCES:**

1. Grimaldi. R.P. "Discrete and Combinatorial Mathematics: An Applied Introduction", 5<sup>th</sup> Edition, Pearson Education Asia, Delhi, 2013.
2. Koshy. T. "Discrete Mathematics with Applications", Elsevier Publications, 2006.
3. Lipschutz. S. and Mark Lipson., "Discrete Mathematics", Schaum's Outlines, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 3<sup>rd</sup> Edition, 2010.

**CS3351**

**DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION**

**L T P C**

**3 0 2 4**

**COURSE OBJECTIVES:**

- To analyze and design combinational circuits.
- To analyze and design sequential circuits
- To understand the basic structure and operation of a digital computer.
- To study the design of data path unit, control unit for processor and to familiarize with the hazards.
- To understand the concept of various memories and I/O interfacing.

**UNIT I**

**COMBINATIONAL LOGIC**

**9**

Combinational Circuits – Karnaugh Map - Analysis and Design Procedures – Binary Adder – Subtractor – Decimal Adder - Magnitude Comparator – Decoder – Encoder – Multiplexers - Demultiplexers

**UNIT II**

**SYNCHRONOUS SEQUENTIAL LOGIC**

**9**

Introduction to Sequential Circuits – Flip-Flops – operation and excitation tables, Triggering of FF, Analysis and design of clocked sequential circuits – Design – Moore/Mealy models, state minimization, state assignment, circuit implementation - Registers – Counters.

**UNIT III**

**COMPUTER FUNDAMENTALS**

**9**

Functional Units of a Digital Computer: Von Neumann Architecture – Operation and Operands of Computer Hardware Instruction – Instruction Set Architecture (ISA): Memory Location, Address and Operation – Instruction and Instruction Sequencing – Addressing Modes, Encoding of Machine Instruction – Interaction between Assembly and High Level Language.

**UNIT IV**

**PROCESSOR**

**9**

Instruction Execution – Building a Data Path – Designing a Control Unit – Hardwired Control, Microprogrammed Control – Pipelining – Data Hazard – Control Hazards.

**UNIT V MEMORY AND I/O**

**9**

Memory Concepts and Hierarchy – Memory Management – Cache Memories: Mapping and Replacement Techniques – Virtual Memory – DMA – I/O – Accessing I/O: Parallel and Serial Interface – Interrupt I/O – Interconnection Standards: USB, SATA

**45 PERIODS**

**PRACTICAL EXERCISES:**

**30 PERIODS**

1. Verification of Boolean theorems using logic gates.
2. Design and implementation of combinational circuits using gates for arbitrary functions.
3. Implementation of 4-bit binary adder/subtractor circuits.
4. Implementation of code converters.
5. Implementation of BCD adder, encoder and decoder circuits
6. Implementation of functions using Multiplexers.
7. Implementation of the synchronous counters
8. Implementation of a Universal Shift register.
9. Simulator based study of Computer Architecture

**COURSE OUTCOMES:**

**At the end of this course, the students will be able to:**

**CO1** : Design various combinational digital circuits using logic gates

**CO2** : Design sequential circuits and analyze the design procedures

**CO3** : State the fundamentals of computer systems and analyze the execution of an instruction

**CO4** : Analyze different types of control design and identify hazards

**CO5** : Identify the characteristics of various memory systems and I/O communication

**TOTAL: 75 PERIODS**

**TEXT BOOKS:**

1. M. Morris Mano, Michael D. Ciletti, "Digital Design : With an Introduction to the Verilog HDL, VHDL, and System Verilog", Sixth Edition, Pearson Education, 2018.
2. David A. Patterson, John L. Hennessy, "Computer Organization and Design, The Hardware/Software Interface", Sixth Edition, Morgan Kaufmann/Elsevier, 2020.

**REFERENCES:**

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, "Computer Organization and Embedded Systems", Sixth Edition, Tata McGraw-Hill, 2012.
2. William Stallings, "Computer Organization and Architecture – Designing for Performance", Tenth Edition, Pearson Education, 2016.
3. M. Morris Mano, "Digital Logic and Computer Design", Pearson Education, 2016.

**CS3352**

**FOUNDATIONS OF DATA SCIENCE**

**L T P C**

**3 0 0 3**

**COURSE OBJECTIVES:**

- To understand the data science fundamentals and process.
- To learn to describe the data for the data science process.
- To learn to describe the relationship between data.

- To utilize the Python libraries for Data Wrangling.
- To present and interpret data using visualization libraries in Python

**UNIT I INTRODUCTION 9**

Data Science: Benefits and uses – facets of data - Data Science Process: Overview – Defining research goals – Retrieving data – Data preparation - Exploratory Data analysis – build the model– presenting findings and building applications - Data Mining - Data Warehousing – Basic Statistical descriptions of Data

**UNIT II DESCRIBING DATA 9**

Types of Data - Types of Variables -Describing Data with Tables and Graphs –Describing Data with Averages - Describing Variability - Normal Distributions and Standard (z) Scores

**UNIT III DESCRIBING RELATIONSHIPS 9**

Correlation –Scatter plots –correlation coefficient for quantitative data –computational formula for correlation coefficient – Regression –regression line –least squares regression line – Standard error of estimate – interpretation of  $r^2$  –multiple regression equations –regression towards the mean

**UNIT IV PYTHON LIBRARIES FOR DATA WRANGLING 9**

Basics of Numpy arrays –aggregations –computations on arrays –comparisons, masks, boolean logic – fancy indexing – structured arrays – Data manipulation with Pandas – data indexing and selection – operating on data – missing data – Hierarchical indexing – combining datasets – aggregation and grouping – pivot tables

**UNIT V DATA VISUALIZATION 9**

Importing Matplotlib – Line plots – Scatter plots – visualizing errors – density and contour plots – Histograms – legends – colors – subplots – text and annotation – customization – three dimensional plotting - Geographic Data with Basemap - Visualization with Seaborn.

**COURSE OUTCOMES:**

**At the end of this course, the students will be able to:**

CO1: Define the data science process

CO2: Understand different types of data description for data science process

CO3: Gain knowledge on relationships between data

CO4: Use the Python Libraries for Data Wrangling

CO5: Apply visualization Libraries in Python to interpret and explore data

**TOTAL:45 PERIODS**

**TEXT BOOKS**

1. David Cielen, Arno D. B. Meysman, and Mohamed Ali, “Introducing Data Science”, Manning Publications, 2016. (Unit I)
2. Robert S. Witte and John S. Witte, “Statistics”, Eleventh Edition, Wiley Publications, 2017. (Units II and III)
3. Jake VanderPlas, “Python Data Science Handbook”, O’Reilly, 2016. (Units IV and V)

**REFERENCES:**

1. Allen B. Downey, “Think Stats: Exploratory Data Analysis in Python”, Green Tea Press,2014.





## REFERENCES

1. Langsam, Augenstein and Tanenbaum, Data Structures Using C and C++, 2nd Edition, Pearson Education, 2015.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms", Fourth Edition, Mcgraw Hill/ MIT Press, 2022.
3. Alfred V. Aho, Jeffrey D. Ullman, John E. Hopcroft, Data Structures and Algorithms, 1st edition, Pearson, 2002.
4. Kruse, Data Structures and Program Design in C, 2nd Edition, Pearson Education, 2006.

**CS3391**

**OBJECT ORIENTED PROGRAMMING**

**L T P C**  
**3 0 0 3**

### COURSE OBJECTIVES:

- To understand Object Oriented Programming concepts and basics of Java programming language
- To know the principles of packages, inheritance and interfaces
- To develop a java application with threads and generics classes
- To define exceptions and use I/O streams
- To design and build Graphical User Interface Application using JAVA FX

### **UNIT I INTRODUCTION TO OOP AND JAVA**

**9**

Overview of OOP – Object oriented programming paradigms – Features of Object Oriented Programming – Java Buzzwords – Overview of Java – Data Types, Variables and Arrays – Operators – Control Statements – Programming Structures in Java – Defining classes in Java – Constructors- Methods -Access specifiers - Static members- Java Doc comments

### **UNIT II INHERITANCE, PACKAGES AND INTERFACES**

**9**

Overloading Methods – Objects as Parameters – Returning Objects –Static, Nested and Inner Classes. Inheritance: Basics– Types of Inheritance -Super keyword -Method Overriding – Dynamic Method Dispatch –Abstract Classes – final with Inheritance. Packages and Interfaces: Packages – Packages and Member Access –Importing Packages – Interfaces.

### **UNIT III EXCEPTION HANDLING AND MULTITHREADING**

**9**

Exception Handling basics – Multiple catch Clauses – Nested try Statements – Java's Built-in Exceptions – User defined Exception. Multithreaded Programming: Java Thread Model–Creating a Thread and Multiple Threads – Priorities – Synchronization – Inter Thread Communication- Suspending –Resuming, and Stopping Threads –Multithreading. Wrappers – Auto boxing.

### **UNIT IV I/O, GENERICS, STRING HANDLING**

**9**

I/O Basics – Reading and Writing Console I/O – Reading and Writing Files. Generics: Generic Programming – Generic classes – Generic Methods – Bounded Types – Restrictions and Limitations. Strings: Basic String class, methods and String Buffer Class.

**UNIT V      JAVAFX EVENT HANDLING, CONTROLS AND COMPONENTS**

**9**

JAVAFX Events and Controls: Event Basics – Handling Key and Mouse Events. Controls: Checkbox, ToggleButton – RadioButtons – ListView – ComboBox – ChoiceBox – Text Controls – ScrollPane. Layouts – FlowPane – HBox and VBox – BorderPane – StackPane – GridPane. Menus – Basics – Menu – Menu bars – MenuItem.

**COURSE OUTCOMES:**

On completion of this course, the students will be able to

**CO1:**Apply the concepts of classes and objects to solve simple problems

**CO2:**Develop programs using inheritance, packages and interfaces

**CO3:**Make use of exception handling mechanisms and multithreaded model to solve real world problems

**CO4:**Build Java applications with I/O packages, string classes, Collections and generics concepts

**CO5:**Integrate the concepts of event handling and JavaFX components and controls for developing GUI based applications

**TOTAL:45 PERIODS**

**TEXT BOOKS:**

1. Herbert Schildt, “Java: The Complete Reference”, 11<sup>th</sup> Edition, McGraw Hill Education, New Delhi, 2019
2. Herbert Schildt, “Introducing JavaFX 8 Programming”, 1<sup>st</sup> Edition, McGraw Hill Education, New Delhi, 2015

**REFERENCE:**

1. Cay S. Horstmann, “Core Java Fundamentals”, Volume 1, 11<sup>th</sup> Edition, Prentice Hall, 2018.

**CS3311**

**DATA STRUCTURES LABORATORY**

**L T P C**  
**0 0 3 1.5**

**COURSE OBJECTIVES:**

- To demonstrate array implementation of linear data structure algorithms.
- To implement the applications using Stack.
- To implement the applications using Linked list
- To implement Binary search tree and AVL tree algorithms.
- To implement the Heap algorithm.
- To implement Dijkstra’s algorithm.
- To implement Prim’s algorithm
- To implement Sorting, Searching and Hashing algorithms.

**LIST OF EXERCISES:**

1. Array implementation of Stack, Queue and Circular Queue ADTs
2. Implementation of Singly Linked List
3. Linked list implementation of Stack and Linear Queue ADTs
4. Implementation of Polynomial Manipulation using Linked list
5. Implementation of Evaluating Postfix Expressions, Infix to Postfix conversion



6. Implementation of Binary Search Trees
7. Implementation of AVL Trees
8. Implementation of Heaps using Priority Queues
9. Implementation of Dijkstra's Algorithm
10. Implementation of Prim's Algorithm
11. Implementation of Linear Search and Binary Search
12. Implementation of Insertion Sort and Selection Sort
13. Implementation of Merge Sort
14. Implementation of Open Addressing (Linear Probing and Quadratic Probing)

**TOTAL:45 PERIODS**

**COURSE OUTCOMES:**

At the end of this course, the students will be able to:

- CO1:** Implement Linear data structure algorithms.
- CO2:** Implement applications using Stacks and Linked lists
- CO3:** Implement Binary Search tree and AVL tree operations.
- CO4:** Implement graph algorithms.
- CO5:** Analyze the various searching and sorting algorithms.

**CS3381**

**OBJECT ORIENTED PROGRAMMING LABORATORY**

**L T P C**  
**0 0 3 1.5**

**COURSE OBJECTIVES:**

- To build software development skills using java programming for real-world applications.
- To understand and apply the concepts of classes, packages, interfaces, inheritance, exception handling and file processing.
- To develop applications using generic programming and event handling

**LIST OF EXPERIMENTS:**

1. Solve problems by using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
2. Develop stack and queue data structures using classes and objects.
3. Develop a java application with an Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club funds. Generate pay slips for the employees with their gross and net salary.
4. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shape.
5. Solve the above problem using an interface.
6. Implement exception handling and creation of user defined exceptions.

7. Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.
8. Write a program to perform file operations.
9. Develop applications to demonstrate the features of generics classes.
10. Develop applications using JavaFX controls, layouts and menus.
11. Develop a mini project for any application using Java concepts.

**Lab Requirements: for a batch of 30 students**

Operating Systems: Linux / Windows

Front End Tools: Eclipse IDE / Netbeans IDE

**TOTAL: 45 PERIODS**

**COURSE OUTCOMES:**

On completion of this course, the students will be able to

**CO1** : Design and develop java programs using object oriented programming concepts

**CO2** : Develop simple applications using object oriented concepts such as package, exceptions

**CO3**: Implement multithreading, and generics concepts

**CO4** : Create GUIs and event driven programming applications for real world problems

**CO5**: Implement and deploy web applications using Java

CS3361

DATA SCIENCE LABORATORY

L T P C  
0 0 4 2

**COURSE OBJECTIVES:**

- To understand the python libraries for data science
- To understand the basic Statistical and Probability measures for data science.
- To learn descriptive analytics on the benchmark data sets.
- To apply correlation and regression analytics on standard data sets.
- To present and interpret data using visualization packages in Python.

**LIST OF EXPERIMENTS:**

1. Download, install and explore the features of NumPy, SciPy, Jupyter, Statsmodels and Pandas packages.
2. Working with Numpy arrays
3. Working with Pandas data frames
4. Reading data from text files, Excel and the web and exploring various commands for doing descriptive analytics on the Iris data set.
5. Use the diabetes data set from UCI and Pima Indians Diabetes data set for performing the following:
  - a. Univariate analysis: Frequency, Mean, Median, Mode, Variance, Standard Deviation, Skewness and Kurtosis.
  - b. Bivariate analysis: Linear and logistic regression modeling
  - c. Multiple Regression analysis
  - d. Also compare the results of the above analysis for the two data sets.

6. Apply and explore various plotting functions on UCI data sets.
  - a. Normal curves
  - b. Density and contour plots
  - c. Correlation and scatter plots
  - d. Histograms
  - e. Three dimensional plotting
7. Visualizing Geographic Data with Basemap

**LIST OF EQUIPMENTS :(30 Students per Batch)**

**Tools:** Python, Numpy, Scipy, Matplotlib, Pandas, statmodels, seaborn, plotly, bokeh

**Note: Example data sets like: UCI, Iris, Pima Indians Diabetes etc.**

**TOTAL: 60 PERIODS**

**COURSE OUTCOMES:**

**At the end of this course, the students will be able to:**

- CO1:** Make use of the python libraries for data science
- CO2:** Make use of the basic Statistical and Probability measures for data science.
- CO3:** Perform descriptive analytics on the benchmark data sets.
- CO4:** Perform correlation and regression analytics on standard data sets
- CO5:** Present and interpret data using visualization packages in Python.

CS3452

THEORY OF COMPUTATION

L T P C

3 0 0 3

**COURSE OBJECTIVES:**

- To understand foundations of computation including automata theory
- To construct models of regular expressions and languages.
- To design context free grammar and push down automata
- To understand Turing machines and their capability
- To understand Undecidability and NP class problems

**UNIT I AUTOMATA AND REGULAR EXPRESSIONS**

**9**

Need for automata theory - Introduction to formal proof – Finite Automata (FA) – Deterministic Finite Automata (DFA) – Non-deterministic Finite Automata (NFA) – Equivalence between NFA and DFA – Finite Automata with Epsilon transitions – Equivalence of NFA and DFA- Equivalence of NFAs with and without  $\epsilon$ -moves- Conversion of NFA into DFA – Minimization of DFAs.

**UNIT II REGULAR EXPRESSIONS AND LANGUAGES**

**9**

Regular expression – Regular Languages- Equivalence of Finite Automata and regular expressions – Proving languages to be not regular (Pumping Lemma) – Closure properties of regular languages.

**UNIT III CONTEXT FREE GRAMMAR AND PUSH DOWN AUTOMATA**

**9**

Types of Grammar - Chomsky's hierarchy of languages -Context-Free Grammar (CFG) and Languages – Derivations and Parse trees – Ambiguity in grammars and languages – Push Down Automata (PDA): Definition – Moves - Instantaneous descriptions -Languages of pushdown automata – Equivalence of pushdown automata and CFG-CFG to PDA-PDA to CFG – Deterministic Pushdown Automata.