BE(CSE) SEMESTER-III, IV with Effect from AY(21-22)

CSE: SEMESTER – III

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
TheoryCourses										
1	HS 103 ME	Operations Research	3	-	-	3	30	70	3	3
2	ES 306 EC	Basic Electronics	3	-	-	4	30	70	3	3
3	ES 303 EC	Digital Electronics	3	1	-	4	30	70	3	3
4	PC 301 CS	Data Structures and Algorithms	3	1	-	4	30	70	3	3
5	PC 302 CS	Discrete Mathematics	3	1	-	4	30	70	3	3
6	PC 303 CS	OOP using JAVA	3	1	-	4	30	70	3	3
Practical/ LaboratoryCourses										
7	PC 351 CS	Data Structures and Algorithms Lab	-	-	2	2	25	50	3	1
8	PC 352 CS	Advanced Computer Skills Lab	-	-	2	2	25	50	3	1
9	ES 351 EC	Basic Electronics Lab	-	-	2	2	25	50	3	1
10	PC 353 CS	OOP using JAVA Lab	-	-	2	2	25	50	3	1
Total			**18**	**4**	**08**	**30**	**280**	**620**		**22**

CSE: SEMESTER – IV

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	HS 104 EG	Effective Technical Communication in English	3	-	-	3	30	70	3	3
2	HS 105 CM	Finance and Accounting	3	1	-	4	30	70	3	3
3	BS 207 MT	Mathematics – III (Probability & Statistics)	3	-	-	3	30	70	3	3
4	ES 305 EC	Signals and Systems	3	-	-	3	30	70	3	3
5	PC 401 CS	Operating Systems	3	-	-	3	30	70	3	3
6	PC 402 CS	Computer Organization	3	1	-	4	30	70	3	3
7	PC 403 CS	Database Management Systems	3	1	-	4	30	70	3	3
Practical/ Laboratory Courses										
8	PC 451 CS	Computer Organization Lab	-	-	2	2	25	50	3	1
9	PC 452 CS	Operating Systems Lab	-	-	2	2	25	50	3	1
10	PC 453 CS	Database Management Systems Lab	-	-	2	2	25	50	3	1
Total			**21**	**3**	**06**	**30**	**285**	**640**		**24**

HS 103 ME **OPERATIONS RESEARCH**

Instruction: 3 periods per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*
Credits: 3

Objectives:

1. Use variables for formulating complex mathematical models in management science, industrial engineering and transportation models.
2. Use the basic methodology for the solution of linear programming problems
3. Understand the mathematical tools that are needed to solve optimization problems like Transportation models and Assignment models
4. Understand the replacement models with change in money value considering with time and without time.
5. Model a system as a queuing model and compute important performance measures

Outcomes: Student will be able to:

1. Prepare the students to have the knowledge of Linear Programming Problem in Operations
2. Research at the end students would be able to understand the concept and develop the models for different applications.
3. Make students understand the concept Replacement models at the end students would able to explain various features and applications of replacement models in real time scenario.
4. Prepare the students to understand theory of Game in operations research at the end students would able to explain application of Game theory in decision making for a conflict

UNIT – I

Introduction: Definition and Scope of Operations Research.

Linear Programming: Introduction, Formulation of linear programming problems, graphical method of solving LP problem, simplex method, maximization and minimization, Degeneracy in LPP, Unbounded and, Infeasible solutions.

UNIT – II

Duality: Definition, Relationship between primal and dual solutions, Economic Interpretation, Post optimal of sensitivity analysis, Dual Simplex Method..

UNIT – III

Transportation Models: Finding an initial feasible solution - North West corner method, least cost method, Vogel's Approximation method, Finding the optimal solution, optimal solution by stepping stone and MODI methods, Special cases in Transportation problems - Unbalanced Transportation problem.

Assignment Problems: Hungarian method of Assignment problem, Maximization in Assignment problem, unbalanced problem, problems with restrictions, travelling salesman problems.

UNIT – IV

Replacement Models: Introduction, replacement of items that deteriorate ignoring change in money value, replacement of items that deteriorate considering change in money value with time, replacement of items that fail suddenly - Individual replacement policy, Group replacement policy.

Game Theory: Introduction, 2 person zero sum games, Maximin - Minimax principle, Principle of Dominance, Solution for mixed strategy problems, Graphical method for 2 x n and m x 2 games.

UNIT – V

Sequencing Models: Introduction, General assumptions, processing n jobs through 2 machines, processing 'n' jobs through m machines, Processing 2 jobs through m machines

Queuing Theory: Introduction, single channel - Poisson arrivals - exponential service times with infinite population & finite population, Multi channel - poison arrivals - Exponential service times with infinite population.

Introduction to Optimization Techniques: Single objective & Multi objective optimization Techniques like G.A, NSGA, P.Q.O & MPSO Techniques.

Suggested Readings:

1	Hamdy, A. Taha, "Operations Research-An Introduction", Sixth Edition, Prentice Hall of India Pvt. Ltd.,1997
2	S.D. Sharma, Operations Research, Kedarnath, Ramnath & Co., Meerut,2009
3	J.B. Gupta, "Utilization of Electric Power and Electric Traction" S.K. Kataria & Sons Publications, 2010 Hrvey M. Wagner, Principles of Operations Research, Second Edition, Prentice Hall of India Ltd., 1980.
4	V.K. Kapoor, Operations Research, S. Chand Publishers, New Delhi,2004
5	R. Paneer Selvam, Operations Research, Second Edition, PHI Learning Pvt. Ltd., New Delhi,2008.

ES 306 EC **BASIC ELECTRONICS**

Instruction: 3+1 *periods per week* *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*
Credits: 3

Course Objectives

1. To understand the characteristics of diodes and transistor configurations
2. To understand the design concepts of biasing of BJT and FET
3. To understand the design concepts of feedback amplifiers and oscillators
4. To study the design concepts of OP Amp and data converters

Course Outcomes

The student will be able to:

1. Study and analyse the rectifiers and regulator circuits.
2. Study and analyse the performance of BJTs, FETs on the basis of their operation and working.
3. Ability to analyse & design oscillator circuits.
4. Ability to analyse different logic gates & multi-vibrator circuits.
5. Ability to analyse different data acquisition systems

UNIT – I
PN Junction Diode: Characteristics, Half wave rectifier, Full wave rectifier, filters, ripple, regulation, TIF and efficiency, Zener diode and Zener diode regulators. CRT construction and CRO applications.
UNIT – II
Transistors: BJT construction and working, modes of operation, configurations of BJT (CB, CE, CC), small signal h-parameter model of CE, CE amplifier analysis. Construction and working of JFET, V-I characteristics of JFET.
UNIT – III
Feedback concepts: Types of negative feedback – modification of gain, bandwidth, input and output impedances, applications. **Oscillators:** RC Phase shift, Wein bridge, LC and crystal Oscillators (Qualitative treatment only).
UNIT – IV
Operational Amplifier: OP-AMP Block diagram, Ideal OP-AMP, DC and AC Characteristics, Inverting and Non-Inverting Amplifiers, Adder/Subtractor, Integrator, Differentiator. **Logic gate circuits -** Introduction to Digital systems- AND, NAND, NOR, XOR gates, Binary half adder, full adder.
UNIT – V
Data Acquisition Systems: Construction and Operation of transducers- Strain guage LVDT, Thermo couple, Instrumentation systems. **Data Converters**: R-2R Ladder DAC, Successive approximation and Flash ADC.

Suggested Readings:

1	Robert Boylestad L. and Louis Nashelsky, Electronic Devices and Circuit Theory, PHI,2007.
2	Helfrick Dand DavidCooper,ModernElectronicInstrumentationandMeasurementsTechniques, 1st Edition, Prentice Hall of India, 2006.
3	Salivahanan, Suresh Kumar and Vallavaraj, Electronic Devices and Circuits, 2nd Edition, Tata McGraw-Hill,2010.

ES 303 EC **DIGITAL ELECTRONICS**

Instruction: 3+ 1 periods per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*
Credits: 3

Course Objectives

1. To learn the principles of digital hardware and support given by it to the software.
2. To explain the operation and design of combinational and arithmetic logic circuits.
3. To design hardware for real world problems.

Course Outcomes

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

1. Understand the deign process of digital hardware, use Boolean algebra to minimize the logical expressions and optimize the implementation of logical functions.
2. Understand the number representation and design combinational circuits like adders, MUXetc.
3. Design Combinational circuits using PLDS and write VHDL code for basic gates and combinational circuits.
4. Analyse sequential circuits using flip-flops and design registers, counters.
6. Represent a sequential circuit using Finite State machine and apply state minimization techniques to design a FSM

UNIT – I
Design Concepts: Digital Hardware, Design process, Design of digital hardware. Introduction to logic circuits – Variables and functions, Logic gates and networks. Boolean algebra, Synthesis using gates, Design examples. Optimized implementation of logic functions using K-Map and Quine-McCluskey Tabular method
UNIT – II
Number Representation: Addition and Subtraction of signed and unsigned numbers.Combinational circuit building blocks: Half adder, Full adder, Multiplexers. Decoders. Encoders. Code converters, BCD to 7-segment converter, Arithmetic comparator circuits.
UNIT – III
Design of combinational circuits using Programmable Logic Devices (PLDs): General structure of a Programmable Array Logic (PAL), Programmable Logic Arrays(PLAs), Structure of CPLDs and FPGAs, 2- input and 3-input lookup tables (LUTs).Introduction to Verilog HDL: Verilog code for basic logic gates, adders, decoders.
UNIT – IV
Sequential Circuits: Basic Latch, Gated SR Latch, gated D Latch, Master-Slave edge triggered flip-flops, T Flip-flop, JK Flip-flop, Excitation tables. Registers, Counters, Verilog code for flip-flops
UNIT – V
Synchronous Sequential Circuits: Basic Design Steps, Finite State machine(FSM) representation using Moore and Mealy state models, State minimization, Design of FSM for Sequence Generation and Detection, Algorithmic State Machine charts.

Suggested Readings:

1	Moris Mano and Michael D CIletti, Digital Design, Pearson, fourth Edition,2008.
2	ZviKohavi, Switching and Finite Automata Theory, 3rd ed., Cambridge University Press-New Delhi, 2011.
3	Samir Palnitkar, "Verilog HDL A Guide to Digital Design and Synthesis," 2nd Edition, Pearson Education,2006.

Course Code	Course Title						Core/Elective
PC301CS	**DATA STRUCTURES AND ALGORITHMS**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
-	L	T	D	P	**30**	**70**	**3**
	3	1	-	-			

Objectives:

1. To develop proficiency in the specification, representation, and implementation of abstract data types and data structures.
2. To discuss the linear and non-linear data structures and their applications
3. To introduce the creation, insertion and deletion operations on binary search trees and balanced binary search trees.
4. To introduce various internal sorting, searching techniques and their time complexities

Outcomes:

After completing this course, the student will be able to:

1. Understand the importance of abstract data type and implementing the concepts of data structure using abstract data type.
2. Evaluate an algorithm by using algorithmic performance and measures.
3. Distinguish between linear and non-linear data structures and their representations in the memory using array and linked list.
4. Apply the suitable data structure for a real world problem and think critically for improvement in solutions.
5. Determine the suitability of the standard algorithms: Searching, Sorting and Traversals

UNIT – I
Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis of an algorithm- Time and Space Complexity, Asymptotic Notations. **Arrays:** Arrays - ADT, Polynomials, Sparse matrices, Strings-ADT, PatternMatching.
UNIT – II
Stacks and Queues: Stacks, Stacks using Arrays, Stacks using dynamic arrays, Evaluation of Expressions – Evaluating Postfix Expression, Infix to Postfix. **Queues:** Queues ADT, operations, Circular Queues, Applications
UNIT – III
Linked Lists: Singly Linked Lists and Chains, Linked Stacks and Queues, Polynomials, Operations for Circularly linked lists, Equivalence Classes, Sparse matrices, Doubly Linked Lists. **Hashing:** Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques
UNIT – IV
Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps, Binary Search trees (BST) : Definition, Searching an element, Insertion into a BST, Deletion from a BST. **Efficient Binary Search Trees:** AVL Trees: Definition, Searching an element, Insertion into a AVL

UNIT – V
Graphs: Graph Abstract Data Type, Elementary Graph operations (DFS and BFS), Minimum Cost Spanning Trees (Prim's and Kruskal's Algorithms). **Sorting and Searching:** Insertion sort, Quick sort, Best computing time for Sorting, Merge sort, Heap sort, shell sort, Sorting on Several Keys, List and Table Sorts, Summary of Internal Sorting, Linear and Binary Search algorithms.

Suggested Books:

1. Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals of Data structures in C, 2nd Edition (2008), Universities Press

Reference Books:

1. Mark A Weiss, Data Structures and Algorithm Analysis In C, Second Edition (2002), Pearson
2. Kushwaha D. S and Misra A.K, Data structures A Programming Approach with C, Second Edition (2014), PHl.
3. Gilberg R. F and Forouzan B. A, Data structures: A Pseudocode Approach with C, Second Edition (2007), Cengage Leaming
4. Tanenbaum A. M , Langsam Y. Augenstein M. J, Data Structures using C, Second Edition (2008), Pearson.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, Third Edition (2009), MIT Press
6. Yedidyah Langsam , Moshe J. Augenstein ,Aaron M. Tenenbaum, Data Structures Using C and C++ , Second Edition (2009), PHI

Proposed for the academic years 2020-2024

DISCRETE MATHEMATICS

PC 302 CS

Instruction: 3 +1 periods per week *Duration of SEE: 3 hours*

CIE: 30 marks *SEE: 70 marks*

Credits: 3

Objectives:

3. To explain with examples, the basic terminology of functions, relations, and sets.
4. To perform the operations associated with sets, functions, and relations.
5. To relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
4. To describe the importance and limitations of predicate logic.
5. To relate the ideas of mathematical induction to recursion and recursively defined structures.
6. To use Graph Theory for solving problems.

Outcomes:

After completing this course, the student will be able to:
5. Illustrate by examples the basic terminology of functions, relations, and sets and demonstrate knowledge of their associated operations.
6. Understand basics of counting, apply permutations and combinations to handle different types of objects.
7. Describe and use recursively-defined relationships to solve problems using generating functions.
8. Analyse semi group, monoid group and abelian group with suitable examples and appreciate group theory applications in computer arithmetic.
9. Demonstrate in practical applications the use of basic counting principles of permutations, combinations, inclusion/exclusion principle and the pigeon hole methodology.

UNIT – I
Mathematical Logic: Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Quantifiers, universal quantifiers. Predicates: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving
UNIT – II
Relations: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Lattices, Hasse diagram. Functions: Inverse Function Composition of functions, recursive Functions, Lattice and its Properties, Algebraic structures: Algebraic systems Examples and general properties, Semi groups and monads, groups sub groups' homomorphism, Isomorphism.
UNIT – III
Elementary Combinatorics: Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion – Exclusion. Pigeon hole principles and its application.
UNIT – IV
Recurrence Relation: Generating Functions, Function of Sequences Calculating Coefficient of generating function, Recurrence relations, Solving recurrence relation by substitution and Generating

funds. Characteristics solution of in homogeneous Recurrence Relation.
UNIT – V
Graph Theory: Representation of Graph, DFS, BFS, Spanning Trees, planar Graphs. Graph Theory and Applications, Basic Concepts Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

Suggested Readings:

1. Elements of Discrete Mathematics- A Computer Oriented Approach- C L Liu, D P Mohapatra. Third Edition, Tata McGrawHill.
7. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott, A. Kandel, T.P. Baker, PHI.
8. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition.TMH.
9. Discrete Mathematical Structures Theory and Application-Malik & Sen,Cengage.
10. Discrete Mathematics with Applications, Thomas Koshy,Elsevier
11. Logic and Discrete Mathematics, Grass Man & Trembley, Pearson Education

OOP using JAVA PC303CS

Instruction: 3 +1 periods per week *Duration of SEE: 3 hours*

CIE: 30 marks *SEE: 70 marks*

Credits: 3

Objectives:

1. To understand fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, difference between applet and application programs, using class libraries
2. To create Java application programs using sound OOP practices such as interfaces, exception handling, multi threading.
3. To understand fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, difference between applet and application programs, using class libraries
4. Use Collection framework, AWT and event handling to solve real world problems.
5. Exploring Swing, and implementing Servlets.

Outcomes:

1. Achieve proficiency in object-oriented concepts and also learns to incorporate the same into the Java programming language.
2. Create Java application programs using sound OOP practices e.g. Inheritance, interfaces and proper program structuring by using packages, access control specifiers.
3. Understand and Implement the concepts of Exception Handling in JAVA.
4. Develop the ability to solve real-world problems through software development in high-level programming language using Large APIs of Java as well as the Java standard class library.
5. Understand File, Streams, Input and Output Handling injava.
6. Create graphical user interface and Applets in java as well as apply the knowledge of Event Handling.

UNIT – I

Object Oriented Programming: Principles, Benefits of Object Oriented Programming.

Introduction to Java: Java buzzwords, bytecode. Java Programming Fundamentals: Applet and Application program using simple java program, data types, variables, arrays, operators, expressions, control statements, type conversion and casting, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, introducing

access control, static, final, nested and inner classes, exploring string class, using command-linear arguments.

Inheritance: Inheritance concept, types of inheritance, Member access rules, use of super and final. Polymorphism - dynamic binding, method overriding, abstract classes and methods.

UNIT – II

Interfaces: Defining an interface, implementing interfaces, extending interface.

Packages: Defining, Creating and Accessing a Package, importing packages

Exception handling: Benefits of exception handling, classification, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, built in exceptions, creating own exception sub classes

Multithreading: Java Thread Model, The Main Thread, creating a Thread, creating multiple threads, using is Alive() and join(), thread priorities, synchronization, inter thread communication,deadlock

UNIT – III

Collections: Overview of Java Collection frame work, commonly used Collection classes – Array List, Linked List, Hash Set, Tree Set, Collection Interfaces – Collection, List, Set. Accessing Collection via iterator, working with Map. Legacy classes and interfaces – Vector, Hashtable, Stack, Dictionary, Enumeration interface.

Other Utility classes: String Tokenizer, Date, Calendar, Gregorian Calendar, ScannerJava Input/Output: exploring java.io, Java I/O classes and interfaces, File, Stream classes, byte stream, character stream, serialization.

UNIT – IV

GUI Programming with java: The AWT class hierarchy, MVC architecture. Applet Revisited: Basics, architecture and skeleton, simple applet program.

Event Handling: Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces. Handling mouse and keyboard events, Adapter classes.

Database Programming using JDBC: Introduction to JDBC, JDBC Drivers & Architecture, CURD operation Using JDBC, Connecting to non-conventional Databases.

UNIT – V

Exploring Swing: JLabel, ImageIcon, JTextField, the Swing buttons, JTabbedpane, JScrollPane, JList, JComboBox.

Servlet: Life cycle, using tomcat, simple servlet, servlet API, javax.servlet package, reading servlet parameters, javax.servlet.http package, handling HTTP requests and responses

Suggested Readings:

1. Herbert Scheldt, "The Complete Reference Java, 7th Edition, Tata McGraw Hill,2006.
2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning,2002.
3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 5th Edition, McGraw Hill Publishing,2010.
4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education /PHI.

Course Code	Course Title						Core/Elective
PC351CS	**DATA STRUCTURES AND ALGORITHMS LAB**						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
-	L	T	D	P	**25**	50	1
	-	-	-	2			

Objectives:

1. To develop skills to design and analyse simple linear and nonlinear data structures, such as stacks, queues and lists and their applications.
2. To gain programming skills to implement sorting and searching algorithms
3. To Strengthen the ability to identify and apply the suitable data structures for the given real world problem
4. To Gain knowledge in practical applications of data structures

Outcomes:

After completing this course, the student will be able to:

1. Implement various data structures using arrays, linked lists
2. Develop ADT necessary for solving problems based on Stacks and Queues
3. Implement binary trees, general tree structures, advanced search trees, heaps, graphs.
4. Implement hash functions and handle collisions
5. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given problem

List of Experiments (Using C programming Language):

1. Implementation of Stacks and Queues using Arrays.
2. Implementation of Circular Queue.
3. Implementation of Infix to Postfix Conversion, Postfix ExpressionEvaluation.
4. Implementation of Singly Linked List
5. Implementation of Doubly Linked List.
6. Implementation of Circular Linked List.
7. Implementation of Stacks, Queues using Linked Lists.
8. Implementation of Binary Search and Hashing
9. Implementation of Operations on Binary Tree (Delete Entire Tree, Copy Entire Tree, Mirror Image, Level Order, Search for a Node etc.)
10. Implementation of Tree Traversals on Binary Trees.
11. Implementation of Binary Search Tree. (Insertion, Deletion and Search operations)
12. Implementation of operations on AVL Trees.
13. Implementation of Traversal on Graphs.
14. Implementation of Prims and Kruskals Algorithm.
15. Implementation of Selection, Merge, Quick, Heap, and InsertionSort.

ADVANCED COMPUTER SKILLS LAB

PC352CS

Instruction: 2 periods per week | Duration of SEE: 3 hours

CIE: 25 marks | SEE: 50 marks

Objectives:

1. To learn programming of python with a focus of basic structure
2. To gain programming skills of python using function and OOP concept
3. To gain practical knowledge of MATLAB toolkit along with operations in matrices and plotting 2D graph

Outcomes:

After completing this course, the student will be able to:
1. Implement basic syntax in Python
2. Analyze and implement different kinds of OOP concept in real world problems.
3. Implement MATLAB operations and graphic functions.

List of Programming Exercises:

1. Python Variables, Executing Python from the Command Line, Editing Python Files, Python Reserved Words.
2. Comments, Strings and Numeric Data Types, Simple Input and Output
3. Control Flow and Syntax, Indenting, if Statement, Relational Operators, Logical Operators, Bit Wise Operators, while Loop, break and continue, for Loop, Lists, Tuples, Sets, Dictionaries
4. Functions: Passing parameters to a Function, Variable Number of Arguments, Scope, Passing Functions to a Function, Mapping Functions in a Dictionary, Lambda, Modules, Standard Modules
5. OOP concepts: Classes, File Organization, Special Methods, Inheritance, Polymorphism, Special Characters, Character Classes, Quantifiers, Dot Character, Greedy Matches, Matching at Beginning or End, Match Objects, Compiling Regular Expressions
6. MATLAB Menus, Toolbars, Computing with MATLAB, Script Files and the Editor/Debugger, MATLAB help System
7. MATLAB controls: Relational Logical Variables. Conditional Statements: if – else – elseif, switch, Loops: for – while –break, continue. User-Defined Functions.
8. Arrays, Matrices and Matrix Operations Debugging MATLAB Programs. Working with Data Files, and Graphing Functions: XY Plots –Sub-plots

Suggested Readings:

1. Mark Summerfield," Programming in Python A Complete introduction to the Python Language", Addison-Wesley Professional,2009.

2. Martin C. Brown," PYTHON: The Complete Reference", McGraw-Hill,2001.

3. W.J. Palm III, Introduction to MATLAB 7 for Engineers, McGraw-Hill International Edition,2005.

4. Wesley J Chun," Core Python Applications Programming", Prentice Hall,2012

5. Hordeski, Michael F, HVAC Control in the New Millennium, Fairmont press, 2001

6. Bela G. Liptak, Process Control-Instrument Engineers Handbook, Chilton book co.

BASIC ELECTRONICS LAB

ES 351 EC

Instruction: 2 periods per week — *Duration of SEE: 3 hours*
CIE: 25 marks — *SEE: 50 marks*
Credits: 1

Objectives:

1.To understand the characteristics of diodes and transistorconfigurations
2.To understand the design concepts of biasing of BJT andFET
3.To understand the design concepts of feedback amplifiers andoscillators
4.To study the design concepts of OP Amp and dataconverters

Outcomes:

1.After completing this course, the student will be able to:
2.Ability to design diode circuits & understand the application of Zenerdiode.
3.Ability to analyse characteristics of BJTs &FETs.
4.Ability to understand the different oscillatorcircuits.
5.Ability to understand operation of HWR & FWR circuits with & withoutfilters.
6.Ability tom design Analog-to-Digital converters & Digital-to-Analogconverters.

List of Experiments:

1	CRO-Applications, Measurements of R, L and C using LCR meter, Colour code method and solderingpractice.
2	Characteristics of Semiconductors diode (Ge, Si andZener)
3	Static Characteristics of BJT-CommonEmitter
4	Static Characteristics of BJT-CommonBase
5	Static Characteristics ofFET
6	RC-Phase ShiftOscillator
7	Hartley and ColpittsOscillators
8	Common EmitterAmplifier
9	AstableMultivibrator
10	Full-wave rectifier with and without filters usingBJT
11	Operational AmplifierApplications
12	Strain GaugeMeasurement
13	Analog-to-Digital and Digital to AnalogConverters

SuggestedReading:

1	MaheshwariandAnand,*LaboratoryExperimentsandPSPICESimulationsinAnalogElectronics,* 1st edition, Prentice Hall of India, 2006.
2	David Bell A., *Laboratory Manual for Electronic Devices and Circuits,* Prentice Hall of India,2001.

OOP using JAVA Lab

PC 353 CS

Instruction: 2 periods per week *Duration of SEE: 3 hours*
CIE: 25 marks *SEE: 50 marks*
Credits: 1

Objectives:

The objectives of the course are to impart knowledge of:
1. To build software development skills using java programming for real world applications.
2. To implement frontend and backend of an application
3. To implement classical problems using java programming.

Outcomes:

After the completion of the course, the student will be able to:
1. Develop Java applications using the concepts of Inheritance, interfaces, packages, access control specifiers.
2. Implement the concepts of Exception Handling in java Applications.
3. Read and write data using different Java I/O streams.
4. Create graphical user interfaces and Applets by applying the knowledge of Event Handling.
5. Create robust applications using Java standard class libraries and retrieve data from a database with JDBC.
6. Ability to solve real-world problems by designing user friendly GUI with befitting backend through the APIs of Java.

List of Experiments:

1. Write a Java program to illustrate the concept of class with method overloading
2. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util)
3. Write a Java program to illustrate the concept of Single level and Multi level Inheritance.
4. Write a Java program to demonstrate the Interfaces & Abstract Classes.
5. Write a Java program to implement the concept of exception handling.

6. Write a Java program to illustrate the concept of threading using Thread Class and runnable Interface.
7. Write a Java program to illustrate the concept of Thread synchronization.
8. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
9. Write a Java program to illustrate collection classes like Array List, LinkedList, Tree map and Hash map.
10. Write a Java program to illustrate Legacy classes like Vector, Hash table, Dictionary & Enumeration interface.
11. Write a Java program to implement iteration over Collection using Iterator interface and List Iterator interface
12. Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
13. Write a Java program to illustrate the concept of I/O Streams
14. Write a Java program to implement serialization concept
15. Write a Java applet program to implement Colour and Graphics class
16. Write a Java applet program for handling mouse & key events
17. Write a Java applet program to implement Adapter classes
18. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
19. Write an example for JDBC prepared statement with ResultSet
20. Write a Java Program to get primary key value (auto-generated keys) from inserted queries using JDBC
21. Write a Java Program to create a simple JList
22. Write a Java Program to create a simple checkbox using JCheckBox
23. Write a Java Program to create a checkbox and Item Listener toit.
24. 1. Write Servlet application to print current date &time 2. Html & Servlet Communication 3. Auto refresh a page 4. Demonstrate session tracking 5. Select record from database 6. Application for login page 7. Insert record into database 8. Count the visits on webpage 9. Insert teacher record in Database