ACE
EngineeringCollege
Ankushapur(V), Ghatkesar(M), R.R.Dist - 501301
(Autonomous)

B.TECH COURSE STRUCTURE & SYLLABUS

(Common to CSE, EEE, IT, CSE(DS))

(R20)

II YEAR II SEMESTER

S. No.	Course Code	Course Title	% Deviation	L	T	P	Credits
1	CS409PC	Formal Language and Automata Theory	0	3	0	0	3
2	CS410PC	Software Engineering	0	3	0	0	3
3	CS403PC	Operating Systems	0	3	0	0	3
4	CS404PC	Database Management Systems	0	3	1	0	4
5	CS405PC	Java Programming	0	3	1	0	4
6	CS406PC	Operating Systems Lab	0	0	0	3	1.5
7	CS407PC	Database Management Systems Lab	0	0	0	3	1.5
8	CS408PC	Java Programming Lab	0	0	0	2	1
9	MC409HS	Constitution of India	0	3	0	0	0
Total Credits				**18**	**2**	**8**	**21**

CS409PC/CS501PC: FORMAL LANGUAGE AND AUTOMATA THEORY

B.Tech. II Year II Semester

Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
CS409PC/CS501PC	PCC	3	0	0	3	30	70	100
Contact Classes: 45	Tutorial Classes:	Practical Classes: Nil				Total Classes: 45		

Prerequisite: Nil

Course Objectives:

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automatically.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

Course Outcomes:

Upon completion of the Course, the students will be able to:

- Able to understand the concept of abstract machines and their power to recognize the languages.
- Able to employ finite state machines for modeling and solving computing problems.
- Able to design context free grammars for formal languages.
- Able to distinguish between decidability and undecidability.

Able to gain proficiency with mathematical tools and formal methods

UNIT: I	Introduction to Finite Automata, NFA & DFA	No.of Classes: 09

Regular Expressions – Concept of regular expression, re module, Using Special Characters, Regular **Introduction to Finite Automata:** Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT: II	Regular Expressions, Pumping lemma for regular expressions	No.of Classes: 09

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT: III	Context free grammers, Push Down Automata	No.of Classes: 09

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages. **Push Down Automata**: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA toCFG.

UNIT: IV	Normal forms for CFG, Turing Machines	No.of Classes: 09

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €- Productions. Chomsky Normal form Griebech Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications

Closure Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties ofCFL's

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turingmachine.

UNIT: V	Types of Turing machine	No.of Classes: 09

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

Text Books:

1. 1Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd edition, PHI.

Reference Books:

1. Introduction to Languages and The Theory of Computation, John C Martin,TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, JohnWiley.
3. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge UniversityPress.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, CengageLearning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R,Pearson.

Web References:

E-Text Books:

CS410PC/CS502PC: SOFTWARE ENGINEERING

B.Tech. II Year II Semester								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
CS410PC/ CS502PC	**PCC**	3	0	0	3	30	70	100
Contact Classes: 45	**Tutorial Classes:**	**Practical Classes: Nil**				**Total Classes: 45**		

Prerequisite: Nil

Course Objectives:

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes:

- Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document(SRD).
- Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

UNIT: I	**Introduction to Software Engineering, Process models**	No.of Classes: 09

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.

Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

UNIT: II	**Software Requirements, Requirements for engineering process, System models**	No.of Classes: 09

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

System models: Context models, behavioral models, data models, object models, structured methods.

UNIT: III	**Design Engineering**	No.of Classes: 09

Design Engineering: Design process and design quality, design concepts, the design model. **Creating an architectural design:** software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic

structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT: IV	Testing Strategies, Product matics	No.of Classes: 09

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT: V	Risk Management, Quality Management	No.of Classes: 09

Metrics for Process and Products: Software measurement, metrics for software quality.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

Text Books:

0. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
1. Software Engineering- Sommerville, 7th edition, Pearson Education.
2. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

Reference Books:

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.

Web References:

E-Text Books:

CS403PC: OPERATING SYSTEMS

B.Tech. II Year II Semester								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
CS403PC	**PCC**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	30	70	100
Contact Classes: 50	**Tutorial Classes: 0**	**Practical Classes: 0**				**Total Classes: 50**		

Prerequisite:

- A course on "Computer Programming and Data Structures".
- A course on "Computer Organization and Architecture".

Course Objectives:

- Provide an introduction to operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
- Introduce the issues to be considered in the design and development of operating system
- Introduce basic Unix commands, system call interface for process management, inter process communication and I/O in Unix

Course Outcomes:

- Will be able to control access to a computer and the files that may be shared
- Demonstrate the knowledge of the components of computer and their respective roles in computing.
- Ability to recognize and resolve user problems with standard operating environments.
- Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

Unit - 1	**Introduction**	No. of Classes:8

Operating System - Introduction, Structures - Simple Batch, Multi programmed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Unit - 2	**Scheduling**	No. of Classes:8

Process and CPU Scheduling - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads, and Interposes Communication, Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling.
System call interface for process management-fork, exit, wait, waitpid, exec

Unit - 3	**Process Management**	No. of Classes:12

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock
Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors
Interposes Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

Unit - 4	Memory Management	No. of Classes:12
Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.		
Unit - 5	**File System**	No. of Classes:10
File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usageof open, create, read, write, close, lseek, stat, ioctl system calls.		
Text Books: 1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7thEdition, John Wiley 2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.		
Reference Books: 1. Operating Systems – Internals and Design Principles Stallings, Fifth Edition– 2005, Pearson Education/PHI 2. Operating System A Design Approach- Crowley,TMH. 3. Modern Operating Systems, Andrew S. Tanenbaum 2ndedition,Pearson/PHI 4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education 5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.		
Web References:		
E-Text Books:		

CS404PC: DATABASE MANAGEMENT SYSTEMS

B.Tech. II Year II Semester

Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
CS404PC	PCC	3	1	0	4	30	70	100
Contact Classes: 45	Tutorial Classes: 15	Practical Classes: Nil				Total Classes: 60		

Prerequisite: A course on "Data Structures".

Course Objectives:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Course Outcomes:

- Gain knowledge of fundamentals of DBMS, database design and normal forms
- Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.
- Familiarity with database storage structures and access techniques

Unit - 1	Database System Applications	No. of Classes:12

Database System Applications: A Historical Perspective, File Systems versus a DBMS**,** the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

Unit - 2	Introduction to the Relational Model	No. of Classes:10

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical data base design, introduction to views, destroying/altering tables and views.

Relational Algebra**,** Tuple relational Calculus, Domain relational calculus.

Unit - 3	SQL	No. of Classes:14

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active data bases.

Schema Refinement: Problems caused by redundancy, decomposition, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multi-valued dependencies, FOURTH normal form, FIFTH normal form.

Unit - 4	Transaction Management	No. of Classes:12

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation-Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

Unit - 5	File Organization	No. of Classes:12

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree base Indexing, Comparison of File Organizations, Indexes and Performance Tuning, Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

Text Books:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, *Tata Mc GrawHill*3rd Edition
2. Database System Concepts, Silberschatz, Korth, *Mc Graw hill*, Vedition.

Reference Books:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7thEdition.
2. Fundamentals of Database Systems, ElmasriNavrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah,*SPD*.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL,Shah,*PHI*.
6. Fundamentals of Database Management Systems, M. L. Gillenson, *Wiley Student Edition*.

Web References:

E-Text Books:

CS405PC: JAVA PROGRAMMING

B.Tech. II Year II Semester								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
CS405PC	**PCC**	3	1	0	4	30	70	100
Contact Classes: 45	**Tutorial Classes: 15**	**Practical Classes: Nil**				**Total Classes: 60**		

Prerequisite:

Course Objectives:

- To introduce the object oriented programming concepts.
- To understand object oriented programming concepts, and apply them in solving problems.
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- To introduce the implementation of packages and interfaces
- To introduce the concepts of exception handling and multi-threading.
- To introduce the design of Graphical User Interface using applets and swing controls.

Course Outcomes:

- Able to solve real world problems using OOPs techniques.
- Able to understand the use of abstract classes.
- Able to solve problems using java collection framework and I/O classes.
- Able to develop multi threaded applications with synchronization.
- Able to develop applets for web applications.
- Able to design GUI based applications

Unit - 1	**Object-Oriented Thinking**	No. of Classes:12

Object-Oriented Thinking- A way of viewing world – Agents and Communities, messages and methods, Responsibilities, Classes and Instances, Class Hierarchies- Inheritance, Method binding, Overriding and Exceptions, Summary of Object-Oriented concepts. Java buzzwords, An Overview of Java, Data types, Variables and Arrays, operators, expressions, control statements, Introducing classes, Methods and Classes, String handling.

Inheritance– Inheritance concept, Inheritance basics, Member access, Constructors, Creating Multilevel hierarchy, super uses, using final with inheritance, Polymorphism-ad hoc polymorphism, pure polymorphism, method overriding, abstract classes, Object class, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance.

Unit - 2	**Packages**	No. of Classes:12

Packages- Defining a Package, CLASSPATH, Access protection, importing packages. Interfaces- defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, The Console class, Serialization, Enumerations, auto boxing, generics.

Unit - 3	Exception handling	No. of Classes:12

Exception handling - Fundamentals of exception handling, Exception types, Termination or resumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes.

Multithreading- Differences between thread-based multitasking and process-based multitasking, Java thread model, creating threads, thread priorities, synchronizing threads, inter thread communication.

Unit - 4	Collections Framework	No. of Classes:12

The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hashtable ,Properties, Stack, Vector More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner

Unit - 5	GUI Programming	No. of Classes:12

GUI Programming with Swing – Introduction, limitations of AWT, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

A Simple Swing Application, Applets – Applets and HTML, Security Issues, Applets and Applications, passing parameters to applets. Creating a Swing Applet, Painting in Swing, A Paint example, Exploring Swing Controls- JLabel and Image Icon, JText Field, **The Swing Buttons**- JButton, JToggle Button, JCheck Box, JRadio Button, JTabbed Pane, JScroll Pane, JList, JCombo Box, Swing Menus, Dialogs.

Text Books:

1. Java The complete reference, 9thedition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books:

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley &sons
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Object Oriented Programming through Java, P. Radha Krishna, University Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ.Press.
5. Java Programming and Object-oriented Application Development, R. A. Johnson, Cengage Learning.

Web References:

1. https://nptel.ac.in/courses/106/105/106105191/

E-Text Books:

1.https://www.oracle.com/technetwork/java/newtojava/java8book-2172125.pdf

CS406PC : OPERATING SYSTEMS LAB

B.Tech. II Year II Semester								
Course Code	**Category**	**Hours/ Week**			**Credits**	**Maximum Marks**		
CS406PC	**PCC**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	3	1.5	30	70	100
Contact Classes: 0	**Tutorial Classes: 0**	**Practical Classes:45**				**Total Classes:45**		

Prerequisite:

- A course on "Programming for Problem Solving".
- A course on "Computer Organization and Architecture".

Course Objectives:

- To provide an understanding of the design aspects of operating system concepts through simulation
- Introduce basic Unix commands, system call interface for process management, inter-process communication and I/O in Unix

Course Outcomes:

- Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
- Able to implement C programs using Unix system calls

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms
   a) FCFS b)SJF c)Round Robin d)priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance andPrevention.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system-calls.
5. Write C programs to illustrate the following IPC mechanisms
   a) Pipe b) FIFOs c) Message Queues d) Shared Memory
6. Write C programs to simulate the following memory management techniques
   a) Paging b) Segmentation

List of Equipment/Software (with Specifications or Range) Required:

A Computer System with Ubuntu operating system and GCC Compiler

References

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System - A Design Approach-Crowley,TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition,Pearson/PHI
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education

CS407PC: DATABASE MANAGEMENT SYSTEMS LAB

<table>
<tr><td colspan="8">B.Tech. II Year II Semester</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2">Category</td><td colspan="3">Hours/ Week</td><td rowspan="2">Credits</td><td colspan="3">Maximum Marks</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>C</td><td>CIA</td><td>SEE</td><td>Total</td></tr>
<tr><td rowspan="2">CS407PC</td><td rowspan="2">PCC</td><td>0</td><td>0</td><td>3</td><td>1.5</td><td>30</td><td>70</td><td>100</td></tr>
<tr><td colspan="2">Contact Classes: 0</td><td colspan="2">Tutorial Classes: 0</td><td colspan="2">Practical Classes: 45</td><td colspan="3">Total Classes:45</td></tr>
<tr><td colspan="8">Prerequisite: Nil</td></tr>
<tr><td colspan="8">Course Objectives:
• Introduce ER data model, database design and normalization
• Learn SQL basics for data definition and data manipulation</td></tr>
<tr><td colspan="8">Course Outcomes:
• Design database schema for a given application and apply normalization
• Acquire skills in using SQL commands for data definition and data manipulation.
• Develop solutions for database applications using procedures, cursors and triggers</td></tr>
<tr><td colspan="8">List of Experiments:
1. Concept design with E-RModel
2. RelationalModel
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage ofCursors</td></tr>
<tr><td colspan="8">List of Equipment/Software (with Specifications or Range) Required:
• System with MySQL / Oracle</td></tr>
<tr><td colspan="8">References
1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7thEdition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah,SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah,PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.</td></tr>
</table>

CS408PC: JAVA PROGRAMMING LAB

B.Tech. II Year II Semester

Course Code	Category	Hours/ Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
CS408PC	PCC	0	0	2	1	30	70	100
Contact Classes: 0	Tutorial Classes: 0	Practical Classes:30				Total Classes:30		

Prerequisite: Nil

Course Objectives:

- To write programs using abstract classes.
- To write programs for solving real world problems using java collection framework.
- To write multi-threaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands on experience with java programming.

Course Outcomes:

- Able to write programs for solving real world problems using java collection framework.
- Able to write programs using abstract classes.
- Able to write multi-threaded programs.
- Able to write GUI programs using swing controls in Java.

List of Experiments:

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2. 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. Handle any possible exceptions like divided by zero.

3. a) Develop an applet in Java that displays a simple message.
   b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

6. Write a Java program for the following: Create a doubly linked list of elements. Delete a given element from the above list. Display the contents of the list after

deletion.

7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.

8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). 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 print Area () that prints the area of the given shape.

9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in GridLayout.

10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).

11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).

12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.

13. Write a Java program to list all the files in a directory including the files present in all its sub-directories.

14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order

15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

List of Equipment/Software (with Specifications or Range) Required:

- Ubuntu System
- Eclipse or Net bean

References

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition *Pearson*education.
2. Thinking in Java, Bruce Eckel, *Pearson*Education.
3. Java Programming, D. S. Malik and P. S. Nair, *Cengage*Learning.
4. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell,*Pearson*.

MC409HS: Constitution of India

B.Tech. II Year II Semester								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
MC409HS	**MC**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	0	30	70	100
Contact Classes: 30	**Tutorial Classes:**	**Practical Classes: Nil**				**Total Classes: 30**		

Prerequisite: Nil

Course Content:

The Constitution of India is the supreme law of India. Parliament of India cannot make any law which violates the Fundamental Rights enumerated under the Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368, however, it cannot use this power to change the "basic structure" of the constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments. The Constitution of India reflects the idea of "Constitutionalism" – a modern and progressive concept historically developed by the thinkers of "liberalism" – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state. The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of "constitutionalism" in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India's legacy of "diversity". It has been said that Indian constitution reflects ideals of its freedom movement; however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values. No law can be "static" and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950. The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it "as one of the strongest court in the world".

Course content

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution ofIndia
3. Salient features and characteristics of the Constitution ofIndia
4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India

9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
11. Emergency Provisions: National Emergency, President Rule, Financial Emergency
12. Local Self Government – Constitutional Scheme in India
13. Scheme of the Fundamental Right to Equality
14. Scheme of the Fundamental Right to certain Freedom under Article19
15. Scope of the Right to Life and Personal Liberty under Article21