MARTHANDAM COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTERSCIENCE AND ENGINEERING

COMPUTER LAB II

Equipment's Available in the Lab

Sl.No	Hardware	Specification	Quantity
1	Desktops	**HCL** Intel Dual core 2.7GHz Processor, Intel G31 Chipset Motherboard, 1GB DDR2 RAM 80GB, 160GB SATA HDD HCL 17" LCD Monitor HCL Keyboard and Optical Mouse	55 Nos
Software			
1	Operating Systems: Linux		
2	Front End Tools: Code blocks		
3	Front End Tools: Eclipse IDE / Netbeans IDE		
4	ArgoUML Selenium, JUnit or Apache JMeter		
5	**Net Beans With Java ,Apache and Mysql,** MY SQL,ORACLE		

COURSES OFFERED

Sl.No	Odd Sem (Course code & Name)	Class	Even Sem (Course code & Name)	Class
1	CS3311 Data structures lab	II CSE	CS8461 Operating Systems Laboratory	II CSE
2	CS3362 C Programming and Datastructures Lab	II ECE	CS8661 Internet Programming Laboratory	IIICSE

3	CS3381 Object Oriented Programming Lab	II CSE	CS8481 Database Management System Laboratory	II CSE II IT
4	CS8582 Object Oriented Analysis And Design Lab	III CSE	CS3481 Database Management System Laboratory	II CSE
5	IT8761 Security Lab	IV CSE	CCS356 Object Oriented Software Engineering	III CSE

CS3311   DATASTRUCTURES

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.

OUTCOMES:

- Implement Linear data structure algorithms.

- Implement applications using Stacks and Linked lists

- Implement Binary Search tree and AVL tree operations.

- Implement graph algorithms.

- Analyze the various searching and sorting algorithms.

LIST OF EXPERIMENTS

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)11: Implementation of Matrix Chain Multiplication

CS3362   C PROGRAMMING AND DATASTRUCTURES LAB

OBJECTIVES:

➢ To develop applications in C

➢ To implement linear and non-linear data structures

➢ To understand the different operations of search trees

➢ To get familiarized to sorting and searching algorithms•

OUTCOMES:

➢ Use different constructs of C and develop applications

➢ Write functions to implement linear and non-linear data structure operations

➢ Suggest and use the appropriate linear / non-linear data structure operations for a given problem

➢ Apply appropriate hash functions that result in a collision free scenario for data storage and Retrieval

➢ Implement Sorting and searching algorithms for a given application

LIST OF EXPERIMENTS:

1. Practice of C programming using statements, expressions, decision making and iterative statements

2. Practice of C programming using Functions and Arrays

3. Implement C programs using Pointers and Structures

4. Implement C programs using Files

5. Development of real time C applications

6. Array implementation of List ADT

7. Array implementation of Stack and Queue ADTs

8. Linked list implementation of List, Stack and Queue ADTs

9. Applications of List, Stack and Queue ADTs

10. Implementation of Binary Trees and operations of Binary Trees

11. Implementation of Binary Search Trees

12. Implementation of searching techniques

13. Implementation of Sorting algorithms : Insertion Sort, Quick Sort, Merge Sort

14. Implementation of Hashing – any two collision technique

CS3381   OBJECT ORIENTED PROGRAMMING LABORATORY

OBJECTIVES:

• To build software development skills using java programming for real-world applications.

• To understand and apply the concepts of classes and packages.

• To understand and apply the concepts of interfaces and inheritance.

- To understand and apply the concepts of exception handling and file processing.
- To develop applications using generic programming and event handling.

OUTCOMES:

- Design and develop java programs using object oriented programming concepts.
- Develop simple applications using object oriented concepts such as package, exceptions.
- Implement multithreading and generics concepts.
- Create GUIs and event driven programming applications for real world problems.
- Implement and deploy web applications using Java.

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.

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 Multiple inheritance 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.

CS8461 OPERATING SYSTEMS LABORATORY

OBJECTIVES

- To learn Unix commands and shell programming
- To implement various CPU Scheduling Algorithms
- To implement Process Creation and Inter Process Communication.
- To implement Deadlock Avoidance and Deadlock Detection Algorithms
- To implement Page Replacement Algorithms
- To implement File Organization and File Allocation Strategies

OUTCOMES:

- Compare the performance of various CPU Scheduling Algorithms
- Implement Deadlock avoidance and Detection Algorithms
- Implement Semaphores
- Create processes and implement IPC
- Analyze the performance of the various Page Replacement Algorithms
- Implement File Organization and File Allocation Strategies

LIST OF EXPERIMENTS

1. Basics of UNIX commands

2. Write programs using the following system calls of UNIX operating system fork, exec, getpid,exit, wait, close, stat, opendir, readdir

3. Write C programs to simulate UNIX commands like cp, ls, grep, etc.

4. Shell Programming

5. Write C programs to implement the various CPU Scheduling Algorithms

6. Implementation of Semaphores

7. Implementation of Shared memory and IPC

8. Bankers Algorithm for Deadlock Avoidance

9. Implementation of Deadlock Detection Algorithm

10. Write C program to implement Threading & Synchronization Applications

11. Implementation of the following Memory Allocation Methods for fixed partition

a) First Fit   b) Worst Fit   c) Best Fit

12. Implementation of Paging Technique of Memory Management

13. Implementation of the following Page Replacement Algorithms

a) FIFO   b) LRU   c) LFU

 14. Implementation of the various File Organization Techniques

15. Implementation of the following File Allocation Strategies

a) Sequential   b) Indexed   c) Linked

CS8582 OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

OBJECTIVES:

To capture the requirements specification for an intended software system

- To draw the UML diagrams for the given specification

- To map the design properly to code

- To test the software system thoroughly for all scenarios

- To improve the design by applying appropriate design patterns.

- Draw standard UML diagrams using an UML modeling tool for a given case study and map design to code and implement a 3 layered architecture. Test the developed code and validate whether the SRS is satisfied.

OUTCOMES:

- Perform OO analysis and design for a given problem specification.

- Identify and map basic software requirements in UML mapping.

- Improve the software quality using design patterns and to explain the rationale behind

•Applying specific design patterns Test the compliance of the software with the SRS.

LIST OF EXPERIMENTS

1. Identify a software system that needs to be developed.

2. Document the Software Requirements Specification (SRS) for the identified system.

3. Identify use cases and develop the Use Case model.

4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagram from that.

5. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams

6. Draw relevant State Chart and Activity Diagrams for the same system.

7. Implement the system as per the detailed design

8. Test the software system for all the scenarios identified as per the usecase diagram

9. Improve the reusability and maintainability of the software system by applying appropriate design patterns.

10. Implement the modified system and test it for various scenarios

SUGGESTED DOMAINS FOR MINI-PROJECT:

1. Passport automation system.

2. Book bank

3. Exam registration

4. Stock maintenance system.

5. Online course reservation system

6. Airline/Railway reservation system

7. Software personnel management system

8. Credit card processing

9. e-book management system

10. Recruitment system

11. Foreign trading system

12. Conference management system

13. BPO management system

14. Library management system

15. Student information system

CS8661 INTERNET PROGRAMMING LABORATORY

OBJECTIVES:

- To be familiar with Web page design using HTML/XML and style sheets
- To be exposed to creation of user interfaces using Java frames and applets.
- To learn to create dynamic web pages using server side scripting.
- To learn to write Client Server applications.
- To be familiar with the PHP programming.
- To be exposed to creating applications with AJAX

OUTCOMES:

- Construct Web pages using HTML/XML and style sheets.
- Build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.
- Develop dynamic web pages using server side scripting.
- Use PHP programming to develop web applications.
- Construct web applications using AJAX and web services.

<p style="text-align:center">MARTHANDAM COLLEGE OF ENGINEERING AND TECHNOLOGY</p>
<p style="text-align:center">DEPARTMENT OF COMPUTERSCIENCE AND ENGINEERING</p>

<p style="text-align:center">LIST OF EXPERIMENTS</p>

1. Web Page Creation Using HTML
2. Web Page Creation Using Cascading Style Sheets
3. Client Side Script Validation Using DHTML
4. Color Palette Using Java Applet
5. Invoking Servlets Using HTML Form
6. Invoking Servlets Using Applet
7. Three Tier Application For Accessing JSP And Database Using Servlets
8. AJAX
9. Program Using PHP
10. Creating XML Document Using CSS
11. Web Service and Database

<p style="text-align:center">IT8761   SECURITY LABORATORY</p>

OBJECTIVES:

- To learn different cipher techniques

- To implement the algorithms DES, RSA,MD5,SHA-1

- To Construct code for authentication algorithms

o   To Develop a signature scheme using Digital signature standard.

- To use network security tools and vulnerability assessment tools

OUTCOMES:

- Develop code for classical Encryption Techniques to solve the problems.

- Build cryptosystems by applying symmetric and public key encryption algorithms.

- Construct code for authentication algorithms.

- Develop a signature scheme using Digital signature standard.

- Demonstrate the network security system using open source tools

LIST OF EXPERIMENTS

1. Perform encryption, decryption using the following substitution techniques
   (i) Ceaser cipher, (ii) playfair cipher iii) Hill Cipher iv) Vigenere cipher

2. Perform encryption and decryption using following transposition techniques
   i) Rail fence ii) row & Column Transformation

3. Apply DES algorithm for practical applications.

4. Apply AES algorithm for practical applications.

5. Implement RSA Algorithm using HTML and JavaScript

6. Implement the Diffie-Hellman Key Exchange algorithm for a given problem.

7. Calculate the message digest of a text using the SHA-1 algorithm.

8. Implement the SIGNATURE SCHEME – Digital Signature Standard.

9. Demonstrate intrusion detection system (ids) using any tool eg. Snort or any other s/w.

10. Automated Attack and Penetration Tools Exploring N-Stalker, a Vulnerability Assessment Tool

11. Defeating Malware
    i) Building Trojans ii) Rootkit Hunter

IT8511   WEB TECHNOLOGY LABORATORY

OBJECTIVES:
- To design interactive web pages using Scripting languages.
- To learn server side programming using servlets and JSP.
- To develop web pages using XML/XSLT.

OUTCOMES:
- Design simple web pages using markup languages like HTML and XHTML.

- Create dynamic web pages using DHTML and java script that is easy to navigate and use.

- Program server side web pages that have to process request from client side web pages.

- Represent web data using XML and develop web pages using JSP.

- Understand various web services and how these web services interact.

LIST OF EXPERIMENTS

1. Web Page Creation Using HTML
2. Web Page Creation Using Cascading Style Sheets
3. Client Side Script Validation Using DHTML
4. Installing and configuring Apache Tom cat Server
5. Invoking Servlets Using HTML Form
6. Invoking Servlets Using Applet
7. Three Tier Application For Accessing JSP Database Using Servlets
8. AJAX
9. Program Using XML And XSL
10. Creating XML Document Using CSS
11. Web Service and Database

CS8481   DATABASE MANAGEMENT SYSTEM LABORATORY

OBJECTIVES:

- To understand data definitions and data manipulation commands
- To learn the use of nested and join queries
- To understand functions, procedures and procedural extensions of data bases
- To be familiar with the use of a front end tool
- To understand design and implementation of typical database applications

OUTCOMES:

- Use typical data definitions and manipulation commands.
- Design applications to test Nested and Join Queries
- Implement simple applications that use Views
- Implement applications that require a Front-end Tool
- Critically analyze the use of Tables, Views, Functions and Procedures

LIST OF EXPERIMENTS

1. Data Definition Commands, Data Manipulation Commands for inserting, deleting, updating and retrieving Tables and Transaction Control statements

2. Database Querying – Simple queries, Nested queries, Sub queries and Joins

3. Views, Sequences, Synonyms

4. Database Programming: Implicit and Explicit Cursors

5. Procedures and Functions

6. Triggers

7. Exception Handling

8. Database Design using ER modeling, normalization and Implementation for any application

9. Database Connectivity with Front End Tools

10. Case Study using real life database applications

CS3481   DATABASE MANAGEMENT SYSTEMS LABORATORY

OBJECTIVES:

- To learn and implement important commands in SQL.
- To learn the usage of nested and joint queries.
- To understand functions, procedures and procedural extensions of databases.
- To understand design and implementation of typical database applications.
- To be familiar with the use of a front end tool for GUI based application development

OUTCOMES:

CO1: Create databases with different types of key constraints.

CO2: Construct simple and complex SQL queries using DML and DCL commands.

CO3: Use advanced features such as stored procedures and triggers and incorporate in GUI based application development. C

CO4: Create an XML database and validate with meta-data (XML schema).

CO5: Create and manipulate data using NOSQL database.

LIST OF EXPERIMENTS

1. Create a database table, add constraints (primary key, unique, check, Not null), insert rows, update and delete rows using SQL DDL and DML commands.
2. Create a set of tables, add foreign key constraints and incorporate referential integrity.
3. Query the database tables using different 'where' clause conditions and also implement aggregate functions.
4. Query the database tables and explore sub queries and simple join operations.
5. Query the database tables and explore natural, equi and outer joins.
6. Write user defined functions and stored procedures in SQL.

7. Execute complex transactions and realize DCL and TCL commands.

8. Write SQL Triggers for insert, delete, and update operations in a database table.

9. Create View and index for database tables with a large number of records.

10. Create an XML database and validate it using XML schema.

11. Create Document, column and graph based data using NOSQL database tools.

12. Develop a simple GUI based database application and incorporate all the above-mentioned

features CS3481 Database Management Systems Lab Manual

13. Case Study using any of the real life database applications from the following list

a) Inventory Management for a EMart Grocery Shop

b) Society Financial Management

c) Cop Friendly App – Eseva

d) Property Management – eMall

e) Star Small and Medium Banking and Finance

 Build Entity Model diagram. The diagram should align with the business and functional goals stated in the application.

 Apply Normalization rules in designing the tables in scope

 Prepared applicable views, triggers (for auditing purposes), functions for enablingenterprise grade features.

 Build PL SQL / Stored Procedures for Complex Functionalities, ex EOD Batch Processing for calculating the EMI for Gold Loan for each eligible Customer. Ability to showcase ACID Properties with sample queries with appropriate settings

CCS356 OBJECT ORIENTED SOFTWARE ENGINEERING

COURSE OBJECTIVES:

• To understand Software Engineering Lifecycle Models

• To Perform software requirements analysis

• To gain knowledge of the System Analysis and Design concepts using UML.

- To understand software testing and maintenance approaches
- To work on project management scheduling using DevOps

COURSE OUTCOMES:

CO1: Compare various Software Development Lifecycle Models

CO2:Evaluate project management approaches as well as cost and schedule estimation strategies.

CO3:Perform formal analysis on specifications.

CO4:Use UML diagrams for analysis and design.

CO5:Architect and design using architectural styles and design patterns, and test the system

LIST OF EXPERIMENTS:

1. Identify a software system that needs to be developed.
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.
4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagram from that.
5. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams
6. Draw relevant State Chart and Activity Diagrams for the same system.
7. Implement the system as per the detailed design
8. Test the software system for all the scenarios identified as per the usecase diagram
9. Improve the reusability and maintainability of the software system by applying appropriate design patterns.
10. Implement the modified system and test it for various scenarios.

SUGGESTED DOMAINS FOR MINI-PROJECT:

1. Passport automation system.

2. Book bank

3. Exam registration

4. Stock maintenance system.

5. Online course reservation system

6. Airline/Railway reservation system

7. Software personnel management system

8. Credit card processing

9. e-book management system

10. Recruitment system

11. Foreign trading system

12. Conference management system