Semester I Year: 2014-2015

Course Title: Software Testing Course Code: 14SSE11
Credits(L:T:P): 3:0:1 Core/Elective: Core
Type of Course: Lecture & Practical Total Contact Hours: 50

COURSE OBJECTIVES

- To explore the basics and goals of software testing.
- To discuss various types of software testing and its techniques
- To list out various tools which can be used for automating the testing process
- To introduce various software quality standards for establishing quality environment
- To discuss various methods and evaluation procedures for improving the quality Models

TOPICS

MODULE-I


MODULE-II

Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the NextDate function, Test cases for the commission problem, Guidelines and observations. Data Flow Testing: Definition-Use testing, Slice-based testing, Guidelines and observations. Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. Integration Testing: A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations, Case study. 10 hours

MODULE-III


MODULE-IV

Object-Oriented Integration Testing: UML support for integration testing, MM-paths for object-oriented software, A framework for object-oriented dataflow integration testing. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program. Object-Oriented System Testing: Currency converter UML description, UML-based system testing, Statechart-based system testing. 10 hours
I Sem M.TECH (SE)

MODULE-V

**Exploratory Testing:** The context-driven school, Exploring exploratory testing, Exploring a familiar example, Exploratory and context-driven testing observations. **Model-Based Testing:** Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing. **Test-Driven Development:** Test-then-code cycles, Automated test execution, Java and JUnit example, Remaining questions, Pros, cons, and open questions of TDD, Retrospective on MDD versus TDD.

10 hours

LABORATORY WORK

1. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive at least 10 different test cases, execute these test cases and discuss the test results.

2. Design, develop, code and run the program in any suitable language to solve the NextDate problem. Analyze it from the perspective of decision table-based testing, derive at least 10 different test cases, execute these test cases and discuss the test results.

3. Design, develop, code and run the program in any suitable object-oriented language to solve the calendar problem. Analyze it from the perspective of OO testing, derive test cases to test the method that increments the date and the method that increments the month, execute these test cases and discuss the test results.

4. Design, develop, code and run the program in any suitable object-oriented language to solve the currency converter problem. Analyze it from the perspective of use case-based system testing, derive appropriate system test cases, execute these test cases and discuss the test results.

COURSE OUTCOMES:
Upon Completion of the course, students shall be able to

- Compare and pick out the right type of software testing process for any given real world problem
- Carry out the software testing process in efficient way
- Automate the testing process by using several testing tools
- Establish a quality environment as specified in standards for developing quality software
- Analyze and improve the quality procedures based on the past experience

TEXT BOOKS:


REFERENCE BOOKS:


Course Title: Advanced Algorithms  Course Code: 14SSE12  
Credits(L:T:P): 3:0:1  Core/Elective: Core  
Type of Course: Lecture & Practical  Total Contact Hours:50  

**COURSE OBJECTIVES**  
- To learn the graph search algorithms.  
- To study network flow and linear programming problems.  
- To learn the hill climbing and dynamic programming design techniques.  
- To develop recursive backtracking algorithms.  
- To get an awareness of NP completeness and randomized algorithms.  

**TOPICS**  

**MODULE-I**  
**Review of Analysis Techniques:** Growth of Functions; Asymptotic notations; Standard notations and common functions; Recurrences and Solution of Recurrence equations- The substitution method, The recurrence – tree method, The master method; Amortized Analysis: Aggregate, Accounting and Potential Methods.  
10 hours  

**MODULE-II**  
**Graph Algorithms:** Bellman - Ford Algorithm; Single source shortest paths in a DAG; Johnson’s Algorithm for sparse graphs; Flow networks and Ford-Fulkerson method; maximum bipartite matching.  
**Polynomials and the FFT:** Representation of polynomials; The DFT and FFT; Efficient implementation of FFT.  
10 hours  

**MODULE-III**  
**Number -Theoretic Algorithms:** Elementary notions; GCD; Modular Arithmetic; Solving modular linear equations; The Chinese remainder theorem; Powers of an element; RSA cryptosystem; Primality testing; Integer factorization.  
10 hours  

**MODULE-IV**  
**String-Matching Algorithms:** Naïve string Matching; Rabin - Karp algorithm; String matching with finite automata; Knuth-Morris-Pratt algorithm; Boyer – Moore algorithms.  
10 hours  

**MODULE-V**  
**Probabilistic and Randomized Algorithms:** Probabilistic algorithms; Randomizing deterministic algorithms, Monte Carlo and Las Vegas algorithms; Probabilistic numeric algorithms.  
10 hours
LABORATORY WORK

1. Design, develop, and run a program in any language to implement the Bellman-Ford algorithm and determine its performance.
2. Design, develop, and run a program in any language to implement Johnson’s algorithm and determine its performance.
3. Design, develop, and run a program in any language to implement a Monte Carlo algorithm to test the primality of a given integer and determine its performance.
4. Design, develop, and run a program in any language to solve the string matching problem using naïve approach and the KMP algorithm and compare their performances.
5. Design, develop, and run a program in any language to solve modular linear equations.
6. Design, develop, and run a program in any language to implement the FFT algorithm efficiently.

COURSE OUTCOMES:
Upon completion of the course, students shall be able to
- Design and apply iterative and recursive algorithms.
- Design and implement optimisation algorithms in specific applications.
- Design appropriate shared objects and concurrent objects for applications.
- Implement and apply concurrent linked lists, stacks, and queues.

TEXT BOOKS:

REFERENCE BOOKS:
Semester I         Year: 2014-2015

Course Title: Advances In Database Management Systems | Course Code: 14SSE13
Credits(L:T:P): 4:0:0 | Core/Elective: Core
Type of Course: Lecture | Total Contact Hours:50

COURSE OBJECTIVES:

- To acquire knowledge on parallel and distributed databases and its applications.
- To study the basics, usage and applications of Object Oriented database.
- To understand the advanced topics data warehousing and mining.
- To learn emerging and advanced data models
- To acquire inquisitive attitude towards research topics in databases.

TOPICS

MODULE-I

Review of Relational Data Model and Relational Database Constraints: Relational model concepts; Relational model constraints and relational database schemas; Update operations, transactions and dealing with constraint violations. 10 hours

MODULE-II

Object and Object-Relational Databases: Overview of Object-Oriented Concepts – Objects, Encapsulation, Type and class hierarchies, complex objects; Object model of ODMG, Object definition Language ODL; Object Query Language OQL; Overview of C++ language binding; Conceptual design of Object database. Overview of object relational features of SQL; Object-relational features of Oracle; Implementation and related issues for extended type systems; The nested relational model.

10 hours

MODULE-III

Parallel and Distributed Databases: Architectures for parallel databases; Parallel query evaluation; Parallelizing individual operations; Parallel query optimizations; Introduction to distributed databases; Distributed DBMS architectures; Storing data in a Distributed DBMS; Distributed catalog management; Distributed Query processing; Updating distributed data; Distributed transactions; Distributed Concurrency control and Recovery.

10 hours

MODULE-IV

Data Warehousing, Decision Support and Data Mining: Introduction to decision support; OLAP, multidimensional model; Window queries in SQL; Finding answers quickly; Implementation techniques for OLAP; Data Warehousing; Views and Decision support; View materialization; Maintaining materialized views. Introduction to Data Mining; Counting co-occurrences; Mining for rules; Tree-
I Sem M.TECH (SE)

structured rules; Clustering; Similarity search over sequences; Incremental mining and data streams; Additional data mining tasks.  

10 hours

MODULE-V

**Enhanced Data Models for Some Advanced Applications:** Active database concepts and triggers; Temporal, Spatial, and Deductive Databases – Basic concepts. More Recent Applications: Mobile databases; Multimedia databases; Geographical Information Systems; Genome data management.

10 hours

**COURSE OUTCOMES:**
Upon completion of the course, students shall be able to
- Select the appropriate high performance database like parallel and distributed database
- Model and represent the real world data using object oriented database
- Embed the rule set in the database to implement data warehousing of mining
- Choose and design database for recent applications database for better interoperability

**TEXT BOOKS:**

**REFERENCE BOOKS:**
Semester I         Year: 2014-2015

Course Title: Computer Systems Performance Analysis
Credits(L:T:P): 4:0:0
Type of Course: Lecture

COURSE OBJECTIVES:
• To understand the mathematical foundations needed for performance evaluation of computer systems
• To understand the metrics used for performance evaluation
• To understand the analytical modeling of computer systems
• To enable the students to develop new queuing analysis for both simple and complex systems
• To introduce the students to analytical techniques for evaluating scheduling policies

TOPICS:
MODULE I

MODULE II
Workloads, Workload Selection and Characterization: Types of Work loads, addition instructions, Instruction mixes, Kernels; Synthetic programs, Application benchmarks, Popular benchmarks. Work load Selection: Services exercised, level of detail; Representativeness; Timeliness, Other considerations in workload selection. Work load characterization Techniques: Terminology; Averaging, Specifying dispersion, Single Parameter Histograms, Multi Parameter Histograms, Principle Component Analysis, Markov Models, Clustering. 10 Hours

MODULE III
Monitors, Program Execution Monitors and Accounting Logs: Monitors: Terminology and classification; Software and hardware monitors, Software versus hardware monitors, Firmware and hybrid monitors, Distributed System Monitors, Program Execution Monitors and Accounting Logs, Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and
I Sem M.TECH (SE)

Interpretation of Accounting log data, Using accounting logs to answer commonly asked questions.  

10 Hours

MODULE IV  
Capacity Planning and Benchmarking: Steps in capacity planning and management; Problems in Capacity Planning; Common Mistakes in Benchmarking; Benchmarking Games; Load Drivers; Remote-Terminal Emulation; Components of an RTE; Limitations of RTEs. Experimental Design and Analysis: Introduction: Terminology, Common mistakes in experiments, Types of experimental designs, 2k Factorial Designs, Concepts, Computation of effects, Sign table method for computing effects; Allocation of variance; General 2k Factorial Designs, General full factorial designs with k factors: Model, Analysis of a General Design, Informal Methods.  

10 Hours

MODULE V  
Queuing Models: Introduction: Queuing Notation; Rules for all Queues: Little’s Law, Types of Stochastic Process. Analysis of Single Queue: Birth-Death Processes; M/M/1 Queue; M/M/m Queue; M/M/m/B Queue with finite buffers; Results for other M/M/1 Queuing Systems. Queuing Networks: Open and Closed Queuing Networks; Product form networks, queuing Network models of Computer Systems. Operational Laws: Utilization Law; Forced Flow Law; Little’s Law; General Response Time Law; Interactive Response Time Law; Bottleneck Analysis; Mean Value Analysis and Related Techniques; Analysis of Open Queuing Networks; Mean Value Analysis; Approximate MVA; Balanced Job Bounds; Convolution Algorithm, Distribution of Jobs in a System, Convolution Algorithm for Computing G(N), Computing Performance using G(N). Timesharing Systems, Hierarchical Decomposition of Large Queuing Networks; Load Dependent Service Centers, Hierarchical Decomposition, Limitations of Queuing Theory.  

10 Hours

COURSE OUTCOMES:  
Upon completion of the course, students shall be able to  
• Identify the need for performance evaluation and the metrics used for it  
• Implement Little law and other operational laws  
• Apply the operational laws to open and closed systems  
• Use discrete-time and continuous-time Markov chains to model real world systems  
• Develop analytical techniques for evaluating scheduling policies  

Text Book:  

Reference Books:  
Semester I         Year: 2014-2015

<table>
<thead>
<tr>
<th>Course Title:</th>
<th>Advances in Computer Networks</th>
<th>Course Code: 14SSE151</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credits(L:T:P):</td>
<td>4:0:0</td>
<td>Core/Elective: Elective</td>
</tr>
<tr>
<td>Type of Course:</td>
<td>Lecture</td>
<td>Total Contact Hours:50</td>
</tr>
</tbody>
</table>

**COURSE OBJECTIVES:**
- Become familiar with the basics of Computer Networks
- Network architectures
- Concepts of fundamental protocols

**TOPICS**

**MODULE-I**

**Foundation:** Building a Network, Requirements, Perspectives, Scalable Connectivity, Cost-Effective Resource sharing, Support for Common Services, Manageability, Protocol layering, Performance, Bandwidth and Latency, Delay X Bandwidth Product, Perspectives on Connecting, Classes of Links, Reliable Transmission, Stop-and-Wait, Sliding Window, Concurrent Logical Channels.

T1: Chapter 1.1, 1.2, 1.5.1, 1.5.2., 2.1, 2.5  
T2: Chapter 4  
10 hours

**MODULE-II**

**Internetworking- I:** Switching and Bridging, Datagrams, Virtual Circuit Switching, Source Routing, Bridges and LAN Switches, Basic Internetworking (IP), What is an Internetwork ?, Service Model, Global Addresses, Datagram Forwarding in IP, subnetting and classless addressing, Address Translation(ARP), Host Configuration(DHCP), Error Reporting(ICMP), Virtual Networks and Tunnels.

T1: Chapter 3.1, 3.2,  
10 hours

**MODULE-III**
I Sem M.TECH (SE)

**Internetworking- II:** Network as a Graph, Distance Vector(RIP), Link State(OSPF), Metrics, The Global Internet, Routing Areas, Routing among Autonomous systems(BGP), IP Version 6(IPv6), Mobility and Mobile IP. T1: Chapter 3.3, 4.1.1, 4.1.3 T2: Chapter 13.1 to 13.18, Chapter 18. **10 hours**

**MODULE-IV**

*End-to-End Protocols:* Simple Demultiplexer (UDP), Reliable Byte Stream(TCP), End-to-End Issues, Segment Format, Connecting Establishment and Termination, Sliding Window Revisited, Triggering Transmission, Adaptive Retransmission, Record Boundaries, TCP Extensions, Queuing Disciplines, FIFO, Fair Queuing, TCP Congestion Control, Additive Increase/ Multiplicative Decrease, Slow Start, Fast Retransmit and Fast Recovery. T1: Chapter 5.1, 5.2.1 to 5.2.8, 6.2, 6.3 **10 hours**

**MODULE-V**

*Congestion Control and Resource Allocation:* Congestion-Avoidance Mechanisms, DEC bit, Random Early Detection (RED), Source-Based Congestion Avoidance. The Domain Name System(DNS), Electronic Mail(SMTP, POP, IMAP, MIME), World Wide Web(HTTP), Network Management(SNMP). T1: Chapter 6.4 T2: Chapter 23.1 to 23.16, Chapter 24, Chapter 25, Chapter 27.1 to 27.8 **10 hours**

**COURSE OUTCOMES:**
Upon completion of the course, students shall be able to
- List and classify network services, protocols and architectures, explain why they are layered.
- Key Internet applications and their protocols, and will apply to develop their own applications (e.g. Client Server applications, Web Services) using the sockets API.
- Explain various congestion control techniques.

**Text books:**

**References:**
I Sem M.TECH (SE)

Semester I         Year: 2014-2015

Course Title: Distributed Operating Systems    Course Code: 14SSE152
Credits(L:T:P): 4:0:0                        Core/Elective: Elective
Type of Course: Lecture                    Total Contact Hours:50

COURSE OBJECTIVE:
- To explore distributed systems principles associated with communication, naming, synchronization, distributed file systems, system design, distributed scheduling, and several case studies
- To understand both foundational concepts and well as practical deployments.
- To gain knowledge on Distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- To gain insight on to the distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols

TOPICS

MODULE-I


10 hours

MODULE-II
Remote Procedure Calls: Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC

10 hours

MODULE-III

Distributed Shared Memory: Introduction, General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM.

Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms.

10 hours

MODULE-IV


10 hours

MODULE-V


10 hours

COURSE OUTCOMES:

Upon completion of the course, students shall be able to

- Understand distributed systems concepts
- Demonstrate an ability to apply theory and techniques to unseen problems.
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Explore the various resource management techniques for distributed systems

TEXT BOOK:


REFERENCE BOOK:

Course Title: Web Services

Credits(L:T:P): 4:0:0

Type of Course: Lecture

Course Code: 14SSE153

Core/Elective: Elective

Total Contact Hours: 50

COURSE OBJECTIVES

• To provide an in-depth knowledge of Web Services.
• To understand the fundamental concepts of Web services.
• To understand the fundamental concepts of WSDL Web Services.
• To design Web service Architecture.
• To Study Building Blocks of Web services.

TOPICS

MODULE-I

Middleware: Understanding the middle ware, RPC and Related Middle ware, TP Monitors, Object Brokers, Message-Oriented Middleware.

10 hours

MODULE-II


10 hours

MODULE-III

Basic Web Services Technology: WSDL Web Services Description Language, UDDI Universal Description Discovery and Integration, Web Services at work interactions between the Specifications, Related Standards.

10 hours

MODULE-IV
Service Coordination Protocols: Infrastructure for Coordination Protocols, WS-Coordination, WS-Transaction, RosettaNet, Other Standards Related to Coordination Protocols.  

**MODULE-V**

Service Composition: Basic of Service Composition, A New Chance of Success for Composition, Services Composition Models, Dependencies between Coordination and Composition, BPEL: Business Process Execution Language for Web Services, OutLook, Applicability of the Web Services, Web services as a Problem and a Solution: AN Example.

COURSE OUTCOMES

Upon completion of the course, students shall be able to
- Bind and unbind services in UDDI.
- Develop WSDL document
- Implement web service client to call public service.
- Implement a service and exposing it as public service.

Text Book


Semester I         Year: 2014-2015

<table>
<thead>
<tr>
<th>Course Title: Advances in Storage Area Networks</th>
<th>Course Code: 14SSE154</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credits(L:T:P): 4:0:0</td>
<td>Core/Elective: Elective</td>
</tr>
<tr>
<td>Type of Course: Lecture</td>
<td>Total Contact Hours:50</td>
</tr>
</tbody>
</table>

COURSE OBJECTIVES

- To understand the fundamentals of storage centric and server centric systems
- To understand the metrics used for Designing storage area networks
- To understand the RAID concepts
- To enable the students to understand how data centre’s maintain the data with the concepts of backup mainly remote mirroring concepts for both simple and complex systems

**TOPIC**

**MODULE-I**

**Introduction:** Server Centric IT Architecture and its Limitations; Storage – Centric IT Architecture and its advantages. Case study: Replacing a server with Storage Networks. The Data Storage and Data Access problem; The Battle for size and access. Intelligent Disk Subsystems: Architecture of Intelligent Disk Subsystems; Hard disks and Internal I/O Channels; JBOD, Storage virtualization using RAID and different RAID levels; Caching: Acceleration of Hard Disk Access; Intelligent disk subsystems, Availability of disk subsystems.

**MODULE-II**
I Sem M.TECH (SE)

I/O Techniques: The Physical I/O path from the CPU to the Storage System; SCSI; Fibre Channel Protocol Stack; Fibre Channel SAN; IP Storage. Network Attached Storage: The NAS Architecture, The NAS hardware Architecture, The NAS Software Architecture, Network connectivity, NAS as a storage system. File System and NAS: Local File Systems; Network file Systems and file servers; Shared Disk file systems; Comparison of Fibre Channel and NAS.  

10 hours

MODULE-III

Storage Virtualization: Definition of Storage virtualization; Implementation Considerations; Storage virtualization on Block or file level; Storage virtualization on various levels of the storage Network; Symmetric and Asymmetric storage virtualization in the Network.  

10 hours

MODULE-IV

SAN Architecture and Hardware devices: Overview, Creating a Network for storage; SAN Hardware devices; The Fibre channel switch; Host Bus Adaptors; Putting the storage in SAN; Fabric operation from a Hardware perspective. Software Components of SAN: The switch’s Operating system; Device Drivers; Supporting the switch’s components; Configuration options for SANs.  

10 hours

MODULE-V


10 hours

COURSE OUTCOMES:  
The students are be able to
- Identify the need for performance evaluation and the metrics used for it
- Apply the techniques used for data maintenance.
- Realize storage virtualization concept,
- Develop techniques for evaluating policies for LUN masking, file systems

Text Book:
1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, Wiley India, 2013.

Reference Books:
I Sem M.TECH (SE)
LABORATORY OBJECTIVES:

- To acquire practical knowledge on advanced databases and its applications.
- To understand and work on areas like Storage, Retrieval, Multi valued attributes, Triggers and other complex objects, Algorithms etc related to ADBMS.
- To design and implement recent applications database for better interoperability.

Note: The following experiments may be implemented on MySQL/ORACLE or any other suitable RDBMS with support for Object features.

1. Develop a database application to demonstrate storing and retrieving of BLOB and CLOB objects.
   
a) Write a binary large object (BLOB) to a database as either binary or character (CLOB) data, depending on the type of the field in your data source. To write a BLOB value to the database, issue the appropriate INSERT or UPDATE statement and pass the BLOB value as an input parameter. If your BLOB is stored as text, such as a SQL Server text field, pass the BLOB as a string parameter. If the BLOB is stored in binary format, such as a SQL Server image field, pass an array of type byte as a binary parameter.

b) Once storing of BLOB and CLOB objects is done, retrieve them and display the results accordingly.

2. Develop a database application to demonstrate the representation of multi valued attributes, and the use of nested tables to represent complex objects. Write suitable queries to demonstrate their use.

Consider Purchase Order Example: This example is based on a typical business activity: managing customer orders. Need to demonstrate how the application might evolve from relational to object-relational, and how you could write it from scratch using a pure object-oriented approach.

a) Show how to implement the schema -- Implementing the Application under the Relational Model -- using only Oracle's built-in data types. Build an object-oriented application on top of this relational schema using object views.

3. Design and develop a suitable Student Database application by considering appropriate attributes. Couple of attributes to be maintained is the Attendance of a student in each subject for which he/she has enrolled and Internal Assessment Using TRIGGERS, write active rules to do the following:
a) Whenever the attendance is updated, check if the attendance is less than 85%; if so, notify the Head of the Department concerned.

b) Whenever, the marks in an Internal Assessment Test are entered, check if the marks are less than 40%; if so, notify the Head of the Department concerned.

**Use the following guidelines when designing triggers:**

- Use triggers to guarantee that when a specific operation is performed, related actions are performed.

- Use database triggers only for centralized, global operations that should be fired for the triggering statement, regardless of which user or database application issues the statement.

- Do not define triggers that duplicate the functionality already built into Oracle. For example, do not define triggers to enforce data integrity rules that can be easily enforced using declarative integrity constraints.

- Limit the size of triggers (60 lines or fewer is a good guideline). If the logic for your trigger requires much more than 60 lines of PL/SQL code, it is better to include most of the code in a stored procedure, and call the procedure from the trigger.

- Be careful not to create recursive triggers. For example, creating an AFTER UPDATE statement trigger on the EMP table that itself issues an UPDATE statement on EMP causes the trigger to fire recursively until it has run out of memory.

4. **Design, develop, and execute a program to implement specific Apriori algorithm for mining association rules. Run the program against any large database available in the public domain and discuss the results.**

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

**LABORATORY OUTCOMES:**

Upon completion of the laboratory, students shall be able to

- Work on the concepts of ADBMS at the practical level
- Model and represent the real world data using object oriented database
- Embed the rules set in the database to implement various features of ADBMS
- Choose, design and implement recent applications database for better interoperability.