ADVANCES IN OPERATING SYSTEMS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016-2017)

SEMESTER – 1

Subject Code: 16SCS11
IA Marks: 20
Number of Lecture Hours/Week: 04
Exam Marks: 80
Total Number of Lecture Hours: 50
Exam Hours: 03

CREDITS – 04

Course objectives: This course will enable students to
- Define the fundamentals of Operating Systems.
- Explain distributed operating system concepts that includes architecture, Mutual exclusion algorithms, Deadlock detection algorithms and agreement protocols
- Illustrate distributed resource management components viz. the algorithms for implementation of distributed shared memory, recovery and commit protocols
- Identify the components and management aspects of Real time, Mobile operating systems

Module 1
Operating System Overview, Process description & Control:

10 Hours

Module 2
Threads, SMP, and Microkernel, Virtual Memory:
- Processes and Threads, Symmetric Multiprocessing (SMP), Micro Kernels, Windows Vista Thread and SMP Hours Management, Linux Process and Thread Management, Hardware and Control Structures, Operating System Software, UNIX Memory Management, Windows Vista Memory Management, Summary

10 Hours

Module 3
Multiprocessor and Real-Time Scheduling:
- Multiprocessor Scheduling, Real-Time Scheduling, UNIX PrecksSl) Scheduling, Windows Vista Hours Scheduling, Process Migration, Distributed Global States, Distributed Mutual Exclusion, Distributed Deadlock

10 Hours

Module 4
Embedded Operating Systems:

10 Hours

Module 5
Kernel Organization:

10 Hours

Course Outcomes
The students should be able to:
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Learn the various resource management techniques for distributed systems
• Identify the different features of real time and mobile operating system
• Modify existing open source kernels in terms of functionality or features used

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**

---

**CLOUD COMPUTING**

[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016 -2017)

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCS12/16SCE12</td>
<td>20</td>
<td>80</td>
<td>03</td>
</tr>
<tr>
<td>16SIT22/16SSE254</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16SCN22/16LN1151</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CREDITS – 04**

**Course objectives:** This course will enable students to

• Define and Cloud, models and Services.
• Compare and contrast programming for cloud and their applications
• Explain virtualization, Task Scheduling algorithms.
• Apply ZooKeeper, Map-Reduce concept to applications.

**Module 1**

**Introduction, Cloud Infrastructure:** Cloud computing, Cloud computing delivery models and services, Ethical issues, Cloud vulnerabilities, Cloud computing at Amazon, Cloud computing the Google perspective, Microsoft Windows Azure and online services, Open-source software platforms for private clouds, Cloud storage diversity and vendor lock-in, Energy use and ecological impact, Service level agreements, User experience and software licensing. Exercises and problems.

**Module 2**


**Module 3**

**Cloud Resource Virtualization:** Virtualization, Layering and virtualization, Virtual machine monitors, Virtual Machines, Performance and Security Isolation, Full virtualization and paravirtualization, Hardware support for virtualization, Case Study:
Xen a VMM based paravirtualization, Optimization of network virtualization, vBlades, Performance comparison of virtual machines, The dark side of virtualization, Exercises and problems

| Module 4 |
|-------------------|---------------------------------|
| **Cloud Resource Management and Scheduling:** Policies and mechanisms for resource management, Application of control theory to task scheduling on a cloud, Stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds, Coordination of specialized autonomic performance managers, A utility-based model for cloud-based Web services, Resourcing bundling: Combinatorial auctions for cloud resources, Scheduling algorithms for computing clouds, Fair queuing, Start-time fair queuing, Borrowed virtual time, Cloud scheduling subject to deadlines, Scheduling MapReduce applications subject to deadlines, Resource management and dynamic scaling, Exercises and problems. | 10 Hours |

| Module 5 |
|-------------------|---------------------------------|
| **Cloud Security, Cloud Application Development:** Cloud security risks, Security: The top concern for cloud users, Privacy and privacy impact assessment, Trust, Operating system security, Virtual machine Security, Security of virtualization, Security risks posed by shared images, Security risks posed by a management OS, A trusted virtual machine monitor, Amazon web services: EC2 instances, Connecting clients to cloud instances through firewalls, Security rules for application and transport layer protocols in EC2, How to launch an EC2 Linux instance and connect to it, How to use S3 in java, Cloud-based simulation of a distributed trust algorithm, A trust management service, A cloud service for adaptive data streaming, Cloud based optimal FPGA synthesis .Exercises and problems. | 10 Hours |

**Course Outcomes**
The students should be able to:
- Compare the strengths and limitations of cloud computing
- Identify the architecture, infrastructure and delivery models of cloud computing
- Apply suitable virtualization concept.
- Choose the appropriate cloud player
- Address the core issues of cloud computing such as security, privacy and interoperability
- Design Cloud Services
- Set a private cloud

**Question paper pattern:**
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
ADVANCES IN DATA BASE MANAGEMENT SYSTEMS  
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016-2017)

SEMESTER – 1

Subject Code | IA Marks | Exam Marks | Exam Hours
--- | --- | --- | ---
16SSE151/16SIT13/16SCS13 | 20 | 80 | 03

Number of Lecture Hours/Week | 04
Total Number of Lecture Hours | 50

CREDITS – 04

Course Objectives: This course will enable students to

- Define parallel and distributed databases and its applications.
- Show applications of Object Oriented database
- Explain basic concepts, principles of intelligent databases.
- Utilize the advanced topics of data warehousing and mining.
- Infer emerging and advanced data models
- Extend knowledge in research topics of databases.

Module 1

**Review of Relational Data Model and Relational Database Constraints:** Relational model concepts; Relational model constraints and relational database schemas; Update operations, anomalies, dealing with constraint violations, Types and violations. Overview of Object-Oriented Concepts – Objects, Basic properties. Advantages, examples, Abstract data types, Encapsulation, class hierarchies, polymorphism, examples.

Module 2

**Object and Object-Relational Databases:** Overview of OOP; Complex objects; Identity, structure etc. Object model of ODMG, Object definition Language ODL; Object Query Language OQL; 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; syntax and demo examples, The nested relational model. Overview of C++ language binding;

Module 3

**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.

Module 4

**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-structured rules; ROC and CMC Curves; Clustering; Similarity search over sequences; Incremental mining and data streams; Additional data mining tasks.

Module 5

**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.

Course Outcomes

The students should be able to:
- Select the appropriate high performance database like parallel and distributed database
- Infer and represent the real world data using object oriented database
- Interpret rule set in the database to implement data warehousing of mining
- Discover and design database for recent applications database for better interoperability

**Question paper pattern:**
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**

---

**PROBABILITY STATISTICS AND QUEUING THEORY**  
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016 -2017)

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>16LNI14 / 16SCN14/16SCS14/16SSE14 / 16SIT14 /16SCE14 / 16SFC14</td>
<td>20</td>
</tr>
</tbody>
</table>

| Number of Lecture Hours/Week | 04 | Exam Marks | 80 |
| Total Number of Lecture Hours | 50 | Exam Hours | 03 |

**CREDITS – 04**

**Course objectives:** This course will enable students to
- Develop analytical capability and to impart knowledge of Probability, Statistics and Queuing.
- Apply above concepts in Engineering and Technology.
- Acquire knowledge of Hypothesis testing and Queuing methods and their applications so as to enable them to apply them for solving real world problems

**Module 1**
Axioms of probability, Conditional probability, Total probability, Baye’s theorem, Discrete Random variable, Probability mass function, Continuous Random variable, Probability density function, Cumulative Distribution Function, and its properties, Two-dimensional Random variables, Joint pdf / cdf and their properties  
10 Hours

**Module 2**
10 Hours

**Module 3**
10 Hours

**Module 4**
Testing Hypothesis: Testing of Hypothesis: Formulation of Null hypothesis, critical
10 Hours
region, level of significance, errors in testing. Tests of significance for Large and Small Samples, t-distribution, its properties and uses, F-distribution, its properties and uses, Chi-square distribution, its properties and uses, $\chi^2$ – test for goodness of fit, $\chi^2$ test for independence.

**Module 5**


**Course Outcomes**

The students should be able to:

- Demonstrate use of probability and characterize probability models using probability mass (density) functions & cumulative distribution functions.
- Explain the techniques of developing discrete & continuous probability distributions and its applications.
- Describe a random process in terms of its mean and correlation functions.
- Outline methods of Hypothesis testing for goodness of fit.
- Define the terminology & nomenclature appropriate queuing theory and also distinguish various queuing models.

**Question paper pattern:**

The question paper will have ten questions.

There will be 2 questions from each module.

Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**


---

**ADVANCES IN DIGITAL IMAGE PROCESSING**

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016-2017)

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCS151</td>
<td>20</td>
<td>80</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

**Course objectives:** This course will enable students to

- Explain image fundamentals and mathematical transforms necessary for image processing and to study the image enhancement techniques.
- Demonstrate the image segmentation and representation techniques.
- How image are analyzed to extract features of interest.
- Introduce the concepts of image registration and image fusion.
- Analyze the constraints in image processing when dealing with 3D data sets.

**Module 1**


<table>
<thead>
<tr>
<th>Module 2</th>
<th>Image Enhancement in the Spatial Domain: Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial Enhancement Methods, Image Enhancement in the Frequency Domain: Introduction to the Fourier Transform and the Frequency Domain, Smoothing frequency-Domain Filters, Sharpening Frequency-Domain Filters, Homomorphic Filtering. 8 Hours</th>
</tr>
</thead>
</table>

| --- | --- |

| --- | --- |

<table>
<thead>
<tr>
<th>Module 5</th>
<th>Morphological Image Processing: Preliminaries, Dilation and Erosion, Opening and Closing, The Hit-or-Miss Transformation, Some Basic Morphological Algorithms. Image Segmentation: Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation. 8 Hours</th>
</tr>
</thead>
</table>

**Course Outcomes**

The students should be able to:
- Explain image formation and the role human visual system plays in perception of gray and color image data.
- Apply image processing techniques in both the spatial and frequency (Fourier) domains.
- Design image analysis techniques in the form of image segmentation and to evaluate the Methodologies for segmentation.
- Conduct independent study and analysis of feature extraction techniques.
- Explain the concepts of image registration and image fusion.
- Analyze the constraints in image processing when dealing with 3D data sets and to apply image
- Apply algorithms in practical applications.

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**

EMBEDDED COMPUTING SYSTEMS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016-2017)

SEMESTER – 1

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCE13</td>
<td></td>
<td>80</td>
<td>03</td>
</tr>
<tr>
<td>16SCS152</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Course objectives: This course will enable students to
- Explain a general overview of Embedded Systems
- Show current statistics of Embedded Systems
- Examine a complete microprocessor-based hardware system
- Design, code, compile, and test real-time software
- Integrate a fully functional system including hardware and software
- Make intelligent choices between hardware/software trade-offs

Module 1
Introduction to embedded systems: Embedded systems, Processor embedded into a system, Embedded hardware units and device in a system, Embedded software in a system, Examples of embedded systems, Design process in embedded system, Formalization of system design, Design process and design examples, Classification of embedded systems, skills required for an embedded system designer.

Module 2
Devices and communication buses for devices network: IO types and example, Serial communication devices, Parallel device ports, Sophisticated interfacing features in device ports, Wireless devices, Timer and counting devices, Watchdog timer, Real time clock, Networked embedded systems, Serial bus communication protocols, Parallel bus device protocols-parallel communication internet using ISA, PCI, PCI-X and advanced buses, Internet enabled systems-network protocols, Wireless and mobile system protocols.

Module 3
Device drivers and interrupts and service mechanism: Programming-I/O busy-wait approach without interrupt service mechanism, ISR concept, Interrupt sources, Interrupt servicing (Handling) Mechanism, Multiple interrupts, Context and the periods for context switching, interrupt latency and deadline, Classification of processors interrupt service mechanism from Context-saving angle, Direct memory access, Device driver programming.

Module 4
Inter process communication and synchronization of processes, Threads and tasks: Multiple process in an application, Multiple threads in an application, Tasks, Task states, Task and Data, Clear-cut distinction between functions. ISRS and tasks by their characteristics, concept and semaphores, Shared data, Inter-process communication,
Signal function, Semaphore functions, Message Queue functions, Mailbox functions, Pipe functions, Socket functions, RPC functions.

Module 5
Real-time operating systems: OS Services, Process management, Timer functions, Event functions, Memory management, Device, file and IO subsystems management, Interrupt routines in RTOS environment and handling of interrupt source calls, Real-time operating systems, Basic design using an RTOS, RTOS task scheduling models, interrupt latency and response of the tasks as performance metrics, OS security issues. Introduction to embedded software development process and tools, Host and target machines, Linking and location software.

Course Outcomes
The students should be able to:
- Distinguish the characteristics of embedded computer systems.
- Examine the various vulnerabilities of embedded computer systems.
- Design an embedded system.
- Design and develop modules using RTOS.
- Implement RPC, threads and tasks.

Question paper pattern:
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

Reference Books:
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 2**

**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.

| 8 Hours |

**Module 3**

**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.

| 8 Hours |

**Module 4**

**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.

| 8 Hours |

**Module 5**


| 8 Hours |

**Course Outcomes**

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

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

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

**Reference Books:**

# ADVANCES IN COMPUTER GRAPHICS
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016 -2017)  

## SEMESTER – I

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>16SCS154 / 16SIT422</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>03</td>
<td>Exam Marks</td>
<td>80</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

## CREDITS – 03

### Course objectives: This course will enable students to
- Explain basic and fundamental computer graphics techniques.
- Compare and contrast image synthesis techniques.
- Examine applications of modeling, design and visualization.
- Discuss different color modeling and computer animation.
- Explain hierarchical modeling and graphing file formats.

### Module 1
Three-Dimensional Object Representations: Polyhedra, OpenGL Polyhedron Functions, Curved Surfaces, Quadric Surfaces, Super quadrics, OpenGL Quadric-Surface and Cubic-Surface Functions, Blobby Objects, Spline Representations, Cubic-Spline Interpolation Methods, Bezier Spline Curves, Bzier Surfaces B-Spline Curves, B-Spline Surfaces, Beta- Splines, Reltional Splines, Conversion Between Spline Representations, Displaying Spline Curves and rfaces, OpenGL Approximation-Spline Functions, Sweep Representations, Constructive Solid –Geometry Method, Octrees, BSP Trees, Fractal-Geometry Methods, Shape Grammars and Others Procedural Methods, Particle Systems, Physically Based Modeling, Visualization Of Data Sets.  

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module 2

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module 3

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module 4

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module 5
Hierarchical modeling and Graphics file formats: Basic modeling concepts, Modeling packages, General hierarchical modeling methods, Hierarchical modeling using openGL display list, Image-File configurations, Color-reduction methods, File-compression techniques, Composition of the major file formats.  

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>8 Hours</th>
</tr>
</thead>
</table>
**Course Outcomes**

The students should be able to:

- Discuss and implement images and objects using 3D representation and openGL methodologies.
- Design and develop surface detection using various detection methods.
- Choose various illumination models for provides effective standards of objects.
- Design of develop effective computer animations.

**Question paper pattern:**

The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**


**OPERATING SYSTEMS AND ADBMS LABORATORY**

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCS16</td>
<td>20</td>
<td>80</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 02**

**Course objectives:** This course will enable students to

- To provide students with contemporary knowledge in Data Compression and Coding.
- To equip students with skills to analyze and evaluate different Data Compression and Coding methods
- To be instrumental to handle multi dimension data compression
- To acquire practical knowledge on advanced databases and its applications.
- To analyze 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

**PART – A  OS LABORATORY WORK:**

1. Design and Develop a UNIX/LINUX shell program that should support at least 10 commands(assume suitable application). OR Design a front-end application upon click of a button corresponding shell command should be executed.

2.

3. Design and develop a program to implement lazy buddy system algorithm.

4.

5. Write a multi-class multithreaded program that simulates multiple sleeping barbers, all in one barbershop that has a finite number of chairs in the waiting room. Each customer is instantiated from a single customer class; each barber is instantiated from a single Barber.
6.
7. 4. Create two processes and demonstrate the usage of Shared segment by the above processes (use shmget, signal, fork etc. to simulate the working environment of the program).
8.
9. 5. Design and develop a program to realize the virus classification, such as boot sector infector, file infector, and macro.

PART – B  ADBMS LABORATORY WORK

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.

1. 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."

**Course Outcomes**

The students should be able to:

- Work on the concepts of Software Testing and ADBMS at the practical level
- 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
- Establish a quality environment as specified in standards for developing quality software
- 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

**Conduction of Practical Examination:**

1. All laboratory experiments (nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from each part and execute both
3. Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
4. **PART – A:** Procedure + Conduction + Viva: 10 + 20 + 10 (40)
5. **PART – B:** Procedure + Conduction + Viva: 10 + 20 + 10 (40)
6. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

**SEMINAR**

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 - 2017)

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>16SCE17 / 16SCN17 / 16LNI17 / 16SIT17 / 16SSE17 / 16SCS17 / 16SFC17</td>
<td>100</td>
<td>----</td>
<td>-</td>
</tr>
</tbody>
</table>

**CREDITS – 01**

**Course objectives:** This course will enable students to

- Motivate the students to read technical article
- Discover recent technology developments

**Descriptions**

The students should read a recent technical article (try to narrow down the topic as much as possible) from any of the leading reputed and refereed journals like:

1. IEEE Transactions, journals, magazines, etc.
2. ACM Transactions, journals, magazines, SIG series, etc.
3. Springer
4. Elsevier publications etc

In the area of (to name few and not limited to)
- Web Technology
- Cloud Computing
- Artificial Intelligent
- Networking
- Security
- Data mining

**Course Outcomes**

The students should be able to:
- Conduct survey on recent technologies
- Infer and interpret the information from the survey conducted
- Motivated towards research

**Conduction:**
The students have to present at least ONE technical seminar on the selected topic and submit a report for internal evaluation.

**Marks Distribution:** Literature Survey + Presentation (PPT) + Report + Question & Answer + Paper: 20 + 30 + 30 + 20 (100).