# CRYPTOGRAPHY, NETWORK SECURITY AND CYBER LAW

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

## SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Total Number of Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS61</td>
<td>20</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>80</td>
<td>3</td>
</tr>
</tbody>
</table>

### COURSES

#### CREDITS – 04

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

- Explain the concepts of Cyber security
- Illustrate key management issues and solutions.
- Familiarize with Cryptography and very essential algorithms
- Introduce cyber Law and ethics to be followed.

## Module – 1


#### Teaching Hours

- **10 Hours**

## Module – 2


#### Teaching Hours

- **10 Hours**

## Module – 3


#### Teaching Hours

- **10 Hours**

## Module – 4


#### Teaching Hours

- **10 Hours**

## Module – 5

**IT act aim and objectives, Scope of the act, Major Concepts, Important provisions, Attribution, acknowledgement, and dispatch of electronic records, Secure electronic records and secure digital signatures, Regulation of certifying authorities: Appointment of Controller and Other officers, Digital Signature certificates, Duties of Subscribers, Penalties and adjudication, The cyber**

#### Teaching Hours

- **10 Hours**
regulations appellate tribunal, Offences, Network service providers not to be liable in certain cases, Miscellaneous Provisions.

**Course outcomes:** The students should be able to:

- Discuss cryptography and its need to various applications
- Design and develop simple cryptography algorithms
- Understand cyber security and need cyber Law

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

<table>
<thead>
<tr>
<th>Text Books</th>
</tr>
</thead>
</table>

**Reference Books:**

<table>
<thead>
<tr>
<th>Reference Books</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Cyber security and Cyber Laws, Alfred Basta, Nadine Basta, Mary brown, ravindra kumar, Cengage learning</td>
</tr>
</tbody>
</table>
COMPUTER GRAPHICS AND VISUALIZATION
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – VI

Subject Code | 15CS62 | IA Marks | 20
---|---|---|---
Number of Lecture Hours/Week | 4 | Exam Marks | 80
Total Number of Lecture Hours | 50 | Exam Hours | 03

CREDITS – 04

Course objectives:
This course will enable students to

- Explain hardware, software and OpenGL Graphics Primitives.
- Illustrate interactive computer graphic using the OpenGL.
- Design and implementation of algorithms for 2D graphics Primitives and attributes.
- Demonstrate Geometric transformations, viewing on both 2D and 3D objects.
- Infer the representation of curves, surfaces, Color and Illumination models

Module – 1

Overview: Computer Graphics and OpenGL:

Text-1: Chapter -1: 1-1 to 1-9, 2-1 to 2-9 (Excluding 2-5), 3-1 to 3-5, 3-9, 3-20

Module – 2

Fill area Primitives, 2D Geometric Transformations and 2D viewing:
Fill area Primitives: Polygon fill-areas, OpenGL polygon fill area functions, fill area attributes, general scan line polygon fill algorithm, OpenGL fill-area attribute functions. 2D Geometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2D Composite transformations, other 2D transformations, raster methods for geometric transformations, OpenGL raster transformations, OpenGL geometric transformations function, 2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions.

Text-1: Chapter 3-14 to 3-16, 4-9, 4-10, 4-14, 5-1 to 5-7, 5-17, 6-1, 6-4

Module – 3

Clipping, 3D Geometric Transformations, Color and Illumination Models:
Clipping: clipping window, normalization and viewport transformations, clipping algorithms, 2D point clipping, 2D line clipping algorithms: cohen-sutherland line clipping only - polygon fill area clipping: Sutherland-Hodgeman polygon clipping algorithm only. 3D Geometric Transformations: 3D translation, rotation, scaling, composite 3D transformations, other 3D transformations, affine transformations, OpenGL geometric transformations functions. Color Models: Properties of light, color models, RGB and CMY color models. Illumination Models: Light sources, basic illumination models-Ambient light, diffuse reflection, specular and phong
Module – 4

**3D Viewing and Visible Surface Detection:**
- 3D Viewing: Concepts, pipeline, coordinate parameters, transformations from world to viewing coordinates, projections, perspective, transformations, viewport, and 3D screen coordinates.
- OpenGL 3D viewing functions.
- Visible Surface Detection: Classification, back face detection, depth buffer method, and OpenGL visibility detection.

| Text-1: Chapter | 6-2 to 6-08 (Excluding 6-4), 5-9 to 5-17 (Excluding 5-15), 12-1, 12-2, 12-4, 12-6, 10-1, 10-3 |

<table>
<thead>
<tr>
<th>3D Viewing and Visible Surface Detection:</th>
<th>10 Hours</th>
</tr>
</thead>
</table>

Module – 5

**Input & Interaction, Curves and Computer Animation:**
- Input and Interaction: Devices, clients, servers, display lists, modeling, event-driven input, menu picking, building interactive models, animating, interactive programs, design of interactive programs, and logic operations.
- Curved surfaces, quadric surfaces, OpenGL quadric-surface and cubic-surface functions, Bézier curves, Bézier surfaces, and OpenGL curve functions.

<table>
<thead>
<tr>
<th>Text-1: Chapter</th>
<th>8-3 to 8-6 (Excluding 8-5), 8-9, 8-10, 8-11, 8-18, 13-11, 3-8, 8-10, 3-2, 13-3, 13-4, 13-10</th>
</tr>
</thead>
</table>

| Text-2: Chapter | 3-1 to 3.11: Input & Interaction |

**Course outcomes:**
- Design and implement algorithms for 2D graphics primitives and attributes.
- Illustrate geometric transformations on both 2D and 3D objects.
- Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and illumination models.
- Decide suitable hardware and software for developing graphics packages using OpenGL.

**Question paper pattern:**
- The question paper will have TEN questions.
- There will be TWO questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
1. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes: *Computer graphics with OpenGL*: pearson education
## Subject: System Software and Compiler Design

**Course Code:** 15CS63  
**IA Marks:** 20  
**Exam Marks:** 80  
**Total Number of Lecture Hours:** 50  
**Exam Hours:** 3  
**Credits:** 4

### Course Objectives:
- Define System Software such as Assemblers, Loaders, Linkers, and Macroprocessors.
- Familiarize with source file, object file, and executable file structures and libraries.
- Describe the front-end and back-end phases of compiler and their importance to students.

### Module 1: Introduction to System Software
- **Teaching Hours:** 10

**Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options.

**Textbook 1:** Chapter 1: 1.1, 1.2, 1.3.1, 1.3.2, Chapter 2: 2.1-2.4, Chapter 4: 4.1.1, 4.1.2

### Module 2: Loaders and Linkers
- **Teaching Hours:** 10

**Textbook 1:** Chapter 3: 3.1-3.5

### Module 3: Introduction to Compiler Structure
- **Teaching Hours:** 10

**Textbook 2:** Chapter 1: 1.1-1.6, Chapter 3: 3.1-3.6

### Module 4: Syntax Analysis
- **Teaching Hours:** 10

**Textbook 2:** Chapter 4: 4.1, 4.2, 4.3, 4.4, 4.5, 4.6

### Module 5: Syntax Directed Translation
- **Teaching Hours:** 10

**Textbook 2:** Chapter 5.1, 5.2, 5.3, 6.1, 6.2, 8.1, 8.2

### Course Outcomes:
- Explain system software such as assemblers, loaders, linkers and macroprocessors.
- Design and develop lexical analyzers, parsers, and code generators.
- Utilize lex and yacc tools for implementing different concepts of system software.
**Question paper pattern:**
The question paper will have TEN questions.  
There will be TWO questions from each module.  
Each question will have questions covering all the topics under a module.  
The students will have to answer FIVE full questions, selecting ONE full question from each module.

<table>
<thead>
<tr>
<th>Text Books:</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. System programming and Compiler Design, K C Louden, Cengage Learning</td>
<td></td>
</tr>
<tr>
<td>3. System software and operating system by D. M. Dhamdhere TMG</td>
<td></td>
</tr>
</tbody>
</table>
### OPERATING SYSTEMS

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

(Effective from the academic year 2016 -2017)

**SEMESTER – VI**

<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>15CS64</td>
<td></td>
<td>80</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

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

- Introduce concepts and terminology used in OS
- Explain threading and multithreaded systems
- Illustrate process synchronization and concept of Deadlock
- Introduce Memory and Virtual memory management, File system and storage techniques

#### Module – 1

**Introduction to operating systems, System structures:**

- What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot. **Process Management** Process concept; Process scheduling; Operations on processes; Inter process communication

**Module – 2**

**Multi-threaded Programming:** Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson’s solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

**Module – 3**

**Deadlocks:**

- Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management:** Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

**Module – 4**

**Virtual Memory Management:** Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System:** File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

**Module – 5**

**Secondary Storage Structures, Protection:** Mass storage structures; Disk

<table>
<thead>
<tr>
<th>Teaching Hours</th>
<th>10 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td></td>
</tr>
</tbody>
</table>

---
structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System:** Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication.

**Course outcomes:** The students should be able to:

- Demonstrate need for OS and different types of OS
- Apply suitable techniques for management of different resources
- Use processor, memory, storage and file system commands
- Realize the different concepts of OS in platform of usage through case studies

**Question paper pattern:**
The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books**
# DATA MINING AND DATA WAREHOUSING

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

**SEMESTER – VI**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>15CS651</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>3</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

- Define multi-dimensional data models.
- Explain rules related to association, classification and clustering analysis.
- Compare and contrast between different classification and clustering algorithms

**Module – 1**

**Data Warehousing & modeling:** Basic Concepts: Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Data Cube: A multidimensional data model, Stars, Snowflakes and Fact constellations: Schemas for multidimensional Data models, Dimensions: The role of concept Hierarchies, Measures: Their Categorization and computation, Typical OLAP Operations.

**Module – 2**

**Data warehouse implementation & Data mining:** Efficient Data Cube computation: An overview, Indexing OLAP Data: Bitmap index and join index, Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP. : Introduction: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity.

**Module – 3**

**Association Analysis:** Association Analysis: Problem Definition, Frequent Item set Generation, Rule generation. Alternative Methods for Generating Frequent Item sets, FP-Growth Algorithm, Evaluation of Association Patterns.

**Module – 4**

**Classification:** Decision Trees Induction, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers.

**Module – 5**

**Clustering Analysis:** Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms.

**Course outcomes:** The students should be able to:

- Identify data mining problems and implement the data warehouse
- Write association rules for a given data pattern.
- Choose between classification and clustering solution.

**Question paper pattern:**

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

<table>
<thead>
<tr>
<th>Text Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining - Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
</table>
SOFTWARE ARCHITECTURE AND DESIGN PATTERNS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code 15CS652 IA Marks 20
Number of Lecture Hours/Week 3 Exam Marks 80
Total Number of Lecture Hours 40 Exam Hours 03

CREDITS – 03

Course objectives: This course will enable students to

- To Learn How to add functionality to designs while minimizing complexity.
- What code qualities are required to maintain to keep code flexible?
- To Understand the common design patterns.
- To explore the appropriate patterns for design problems

Module – 1

Introduction: what is a design pattern? describing design patterns, the catalog of
design pattern, organizing the catalog, how design patterns solve design
problems, how to select a design pattern, how to use a design pattern. What is
object-oriented development?, key concepts of object oriented design other
related concepts, benefits and drawbacks of the paradigm

Module – 2

Analysis a System: overview of the analysis phase, stage 1: gathering the
requirements functional requirements specification, defining conceptual classes
and relationships, using the knowledge of the domain. Design and
Implementation, discussions and further reading.

Module – 3

Design Pattern Catalog: Structural patterns, Adapter, bridge, composite,
decorator, facade, flyweight, proxy.

Module – 4

Interactive systems and the MVC architecture: Introduction, The MVC
architectural pattern, analyzing a simple drawing program, designing the system,
designing of the subsystems, getting into implementation, implementing undo
operation, drawing incomplete items, adding a new feature, pattern based
solutions.

Module – 5

Designing with Distributed Objects: Client server system, java remote method
invocation, implementing an object oriented system on the web (discussions and
further reading) a note on input and output, selection statements, loops arrays.

Course outcomes: The students should be able to:

- Design and implement codes with higher performance and lower complexity
- Be aware of code qualities needed to keep code flexible
- Experience core design principles and be able to assess the quality of a design
  with respect to these principles.
- Capable of applying these principles in the design of object oriented systems.
- Demonstrate an understanding of a range of design patterns. Be capable of
  comprehending a design presented using this vocabulary.
- Be able to select and apply suitable patterns in specific contexts

Question paper pattern:
The question paper will have TEN questions. 
There will be TWO questions from each module. 
Each question will have questions covering all the topics under a module. 
The students will have to answer FIVE full questions, selecting ONE full question from each module.

<table>
<thead>
<tr>
<th><strong>Text Books:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Object-oriented analysis, design and implementation, brahma dathan, sarnath rammath, universities press, 2013</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Reference Books:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Frank Bachmann, RegineMeunier, Hans Rohnert “Pattern Oriented Software Architecture” – Volume 1, 1996.</td>
</tr>
</tbody>
</table>
OPERATIONS RESEARCH
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS653</td>
<td>20</td>
<td>3</td>
<td>80</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS – 03

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

- Formulate optimization problem as a linear programming problem.
- Solve optimization problems using simplex method.
- Formulate and solve transportation and assignment problems.
- Apply game theory for decision making problems.

**Module – 1**

**Introduction, Linear Programming:** Introduction: The origin, nature and impact of OR; Defining the problem and gathering data; Formulating a mathematical model; Deriving solutions from the model; Testing the model; Preparing to apply the model; Implementation.

**Introduction to Linear Programming Problem (LPP):** Prototype example, Assumptions of LPP, Formulation of LPP and Graphical method various examples.

**Module – 2**

**Simplex Method – 1:** The essence of the simplex method; Setting up the simplex method; Types of variables, Algebra of the simplex method; the simplex method in tabular form; Tie breaking in the simplex method, Big M method, Two phase method.

**Module – 3**

**Simplex Method – 2: Duality Theory** - The essence of duality theory, Primal dual relationship, conversion of primal to dual problem and vice versa. The dual simplex method.

**Module – 4**


**Module – 5**

**Game Theory:** Game Theory: The formulation of two persons, zero sum games; saddle point, maximin and minimax principle, Solving simple games- a prototype example; Games with mixed strategies; Graphical solution procedure.

**Metaheuristics:** The nature of Metaheuristics, Tabu Search, Simulated Annealing, Genetic Algorithms.

**Course outcomes:** The students should be able to:

- Select and apply optimization techniques for various problems.
- Model the given problem as transportation and assignment problem and solve.
- Apply game theory for decision support system.

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
<tr>
<td>8 Hours</td>
</tr>
<tr>
<td>8 Hours</td>
</tr>
<tr>
<td>8 Hours</td>
</tr>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>
**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

<table>
<thead>
<tr>
<th><strong>Text Books:</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Reference Books:</strong></th>
</tr>
</thead>
</table>
# DISTRIBUTED COMPUTING SYSTEM

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

## SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS654</td>
<td>20</td>
<td>3</td>
<td>80</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

## Course Objectives:
This course will enable students to

- Explain distributed system, their characteristics, challenges and system models.
- Describe IPC mechanisms to communicate between distributed objects.
- Illustrate the operating system support and File Service architecture in a distributed system.
- Analyze the fundamental concepts, algorithms related to synchronization.

## Module – 1

### Characterization of Distributed Systems:
Introduction, Examples of DS, Resource sharing and the Web, Challenges

### System Models:
Architectural Models, Fundamental Models

8 Hours

## Module – 2

### Inter Process Communication:
Introduction, API for Internet Protocols, External Data Representation and Marshalling, Client – Server Communication, Group Communication

### Distributed Objects and RMI:
Introduction, Communication between Distributed Objects, RPC, Events and Notifications

8 Hours

## Module – 3

### Operating System Support:
Introduction, The OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture

### Distributed File Systems:
Introduction, File Service architecture, Sun Network File System

8 Hours

## Module – 4

### Time and Global States:
Introduction, Clocks, events and process status, Synchronizing physical clocks, Logical time and logical clocks, Global states

### Coordination and Agreement:
Introduction, Distributed mutual exclusion, Elections

8 Hours

## Module – 5

### Distributed Transactions:
Introduction, Flat and nested distributed transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed deadlocks

8 Hours

## Course Outcomes:
The students should be able to:

- Explain the characteristics of a distributed system along with its and design challenges
- Illustrate the mechanism of IPC between distributed objects
- Describe the distributed file service architecture and the important characteristics of SUN NFS.
- Discuss concurrency control algorithms applied in distributed transactions

## Question Paper Pattern:
The question paper will have TEN questions.
There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**


**Reference Books:**

SYSTEM SOFTWARE AND OPERATING SYSTEM LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMMETER – VI

Subject Code 15CSL67  IA Marks 20
Number of Lecture Hours/Week 01L + 02P  Exam Marks 80
Total Number of Lecture Hours 40  Exam Hours 03

CREDITS – 02

Course objectives:
This course will enable students to
• To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler Design and implement programs on these phases using LEX & YACC tools and/or C/C++/Java
• To enable students to learn different types of CPU scheduling algorithms used in operating system.
• To make students able to implement memory management - page replacement and deadlock handling algorithms

Description (If any):
Exercises to be prepared with minimum three files (Where ever necessary):
   i. Header file.
   ii. Implementation file.
   iii. Application file where main function will be present.

The idea behind using three files is to differentiate between the developer and user sides. In the developer side, all the three files could be made visible. For the user side only header file and application files could be made visible, which means that the object code of the implementation file could be given to the user along with the interface given in the header file, hiding the source file, if required. Avoid I/O operations (printf/scanf) and use data input file where ever it is possible

Lab Experiments:

1.  
a) Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately.

b) Write YACC program to evaluate arithmetic expression involving operators: +, -, *, and /

2.  
Develop, Implement and Execute a program using YACC tool to recognize all strings ending with $b$ preceded by $n$ $a$'s using the grammar $a^n b$ (note: input $n$ value)

3.  
Design, develop and implement YACC/C program to construct Predictive / LL(1) Parsing Table for the grammar rules: $A \rightarrow aB a$ , $B \rightarrow bB | e$. Use this table to parse the sentence: $abba$

4.  
Design, develop and implement YACC/C program to demonstrate Shift Reduce Parsing technique for the grammar rules: $E \rightarrow E + T | T$, $T \rightarrow T * F | F$, $F \rightarrow (E)$ | id and parse the sentence: $id + id * id$.

5.  
Design, develop and implement a C/Java program to generate the machine code using
Triples for the statement \( A = -B \times (C + D) \) whose intermediate code in three-address form:

\[
\begin{align*}
T1 &= -B \\
T2 &= C + D \\
T3 &= T1 + T2 \\
A &= T3
\end{align*}
\]

6. a) Write a LEX program to eliminate comment lines in a C program and copy the resulting program into a separate file.
   b) Write YACC program to recognize valid identifier, operators and keywords in the given text (C program) file.

7. Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms. Experiment with different quantum sizes for RR algorithm.

8. Design, develop and implement a C/C++/Java program to implement Banker’s algorithm. Assume suitable input required to demonstrate the results.

9. Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.

Study Experiment / Project:

NIL

Course outcomes: The students should be able to:

- Implement and demonstrate Lexer’s and Parser’s
- Evaluate different algorithms required for management, scheduling, allocation and communication used in operating system.

Conduction of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Students are allowed to pick one experiment from the lot.
- Strictly follow the instructions as printed on the cover page of answer script
- Marks distribution: Procedure + Conduction + Viva: 20 + 50 + 10 (80)
- Change of experiment is allowed only once and marks allotted to the procedure part to be made zero
# COMMERCE LABORATORY WITH MINI PROJECT

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

**SEMESTER – VI**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>15CSL68</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>01I + 02P</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 – 02**

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

- Demonstrate simple algorithms using OpenGL Graphics Primitives and attributes.
- Implementation of line drawing and clipping algorithms using OpenGL functions.
- Design and implementation of algorithms Geometric transformations on both 2D and 3D objects.

**Description (If any):**

--

**Lab Experiments:**

**PART A**

Design, develop, and implement the following programs using OpenGL API

1. Implement Bevan’s line drawing algorithm for all types of slope.
   - Refer: Text-1: Chapter 3.5
   - Refer: Text-2: Chapter 8
2. Create and rotate a triangle about the origin and a fixed point.
   - Refer: Text-1: Chapter 5-4
3. Draw a colour cube and spin it using OpenGL transformation matrices.
   - Refer: Text-2: Modelling a Coloured Cube
4. Draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing.
   - Refer: Text-2: Topic: Positioning of Camera
5. Clip a lines using Cohen-Sutherland algorithm
   - Refer: Text-1: Chapter 6.7
   - Refer: Text-2: Chapter 8
6. To draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the surfaces of the solid object used in the scene.
   - Refer: Text-2: Topic: Lighting and Shading
7. Design, develop and implement recursively subdivide a tetrahedron to form 3D sierpinski gasket. The number of recursive steps is to be specified by the user.
   - Refer: Text-2: Topic: sierpinski gasket
8. Develop a menu driven program to animate a flag using Bezier Curve algorithm
   - Refer: Text-1: Chapter 8-10
9. Develop a menu driven program to fill the polygon using scan line algorithm

**Project:**

**PART –B (MINI-PROJECT):**

Student should develop mini project on the topics mentioned below or similar applications using Open GL API. Consider all types of attributes like color, thickness, styles, font, background, speed etc., while doing mini project.

(During the practical exam: the students should demonstrate and answer Viva-Voce)

**Sample Topics:**

Simulation of concepts of OS, Data structures, algorithms etc.
Course outcomes: The students should be able to:

- Apply the concepts of computer graphics
- Implement computer graphics applications using OpenGL
- Animate real world problems using OpenGL

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks as per 6(b).
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
   a) Part A: Procedure + Conduction + Viva: 10 + 35 + 5 = 50 Marks
   b) Part B: Demonstration + Report + Viva voce = 15 + 10 + 5 = 30 Marks
7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

Reference books: