CRYPTOGRAPHY, NETWORK SECURITY AND CYBER LAW
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS61	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 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**
Introduction - Cyber Attacks, Defence Strategies and Techniques, Guiding Principles, Mathematical Background for Cryptography - Modulo Arithmetic's, The Greatest Comma Divisor, Useful Algebraic Structures, Chinese Remainder Theorem, Basics of Cryptography - Preliminaries, Elementary Substitution Ciphers, Elementary Transport Ciphers, Other Cipher Properties, Secret Key Cryptography – Product Ciphers, DES Construction, Modes of Operation, MAC and Other Applications, Attacks, Linear Cryptanalysis.	**10 Hours**
Module – 2	
Public Key Cryptography and RSA – RSA Operations, Why Does RSA Work?, Performance, Applications, Practical Issues, Public Key Cryptography Standard (PKCS), Cryptographic Hash - Introduction, Properties, Construction, Applications and Performance, The Birthday Attack, Discrete Logarithm and its Applications - Introduction, Diffie-Hellman Key Exchange, Other Applications, Elliptic Curve Cryptography and Advanced Encryption Standard - Elliptic Curve Cryptography, Applications, Practical Considerations, Advanced Encryption Standard (AES).	**10 Hours**
Module – 3	
Key Management - Introduction, Digital Certificates, Public Key Infrastructure, Identity–based Encryption, Authentication–I - One way Authentication, Mutual Authentication, Dictionary Attacks, Authentication – II – Centalised Authentication, The Needham-Schroeder Protocol, Kerberos, Biometrics, IPSec-Security at the Network Layer – Security at Different layers: Pros and Cons, IPSec in Action, Internet Key Exchange (IKE) Protocol, Security Policy and IPSEC, Virtual Private Networks, Security at the Transport Layer - Introduction, SSL Handshake Protocol, SSL Record Layer Protocol, OpenSSL.	**10 Hours**
Module – 4	
IEEE 802.11 Wireless LAN Security - Background, Authentication, Confidentiality and Integrity, Viruses, Worms, and Other Malware - Preliminaries Viruses, Worm Features, Internet Scanning Worms, Topological Worms, Web Worms and Case Study, Firewalls – Basics, Practical Issues, Intrusion Prevention and Detection - Introduction, Prevention Versus Detection, Types of Instruction Detection Systems, DDoS Attacks Prevention/Detection, Web Service Security – Motivation, Technologies for Web Services, WS-Security, SAML, Other Standards.	**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 regulations appellate tribunal, Offences, Network service providers not to be liable in certain cases, Miscellaneous Provisions.	**10 Hours**

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:

1. Cryptography, Network Security and Cyber Laws – Bernard Menezes, Cengage Learning, 2010 edition (Chapters-1,3,4,5,6,7,8,9,10,11,12,13,14,15,19(19.1-19.5),21(21.1-21.2),22(22.1-22.4),25

Reference Books:

1. Cryptography and Network Security- Behrouz A Forouzan, Debdeep Mukhopadhyay, Mc-GrawHill, 3rd Edition, 2015
2. Cryptography and Network Security- William Stallings, Pearson Education, 7th Edition
3. Cyber Law simplified- Vivek Sood, Mc-GrawHill, 11th reprint , 2013
4. Cyber security and Cyber Laws, Alfred Basta, Nadine Basta, Mary brown, ravindra kumar, Cengage learning

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	**Teaching Hours**
Overview: Computer Graphics and OpenGL: Computer Graphics:Basics of computer graphics, Application of Computer Graphics, Video Display Devices: Random Scan and Raster Scan displays, color CRT monitors, Flat panel displays. Raster-scan systems: video controller, raster scan Display processor, graphics workstations and viewing systems, Input devices, graphics networks, graphics on the internet, graphics software. OpenGL: Introduction to OpenGL ,coordinate reference frames, specifying two-dimensional world coordinate reference frames in OpenGL, OpenGL point functions, OpenGL line functions, point attributes, line attributes, curve attributes, OpenGL point attribute functions, OpenGL line attribute functions, Line drawing algorithms(DDA, Bresenham's), circle generation algorithms (Bresenham's). **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**	**10 Hours**
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. 2DGeometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2DComposite 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**	**10 Hours**
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.3DGeometric 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	**10 Hours**

model, Corresponding openGL functions. **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**	
Module – 4	
3D Viewing and Visible Surface Detection: 3DViewing:3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters , Transformation from world to viewing coordinates, Projection transformation, orthogonal projections, perspective projections, The viewport transformation and 3D screen coordinates. OpenGL 3D viewing functions. Visible Surface Detection Methods: Classification of visible surface Detection algorithms, back face detection, depth buffer method and OpenGL visibility detection functions. **Text-1:Chapter: 7-1 to 7-10(Excluding 7-7), 9-1 to 9-3, 9-14**	**10 Hours**
Module – 5	
Input& interaction, Curves and Computer Animation: Input and Interaction: Input devices, clients and servers, Display Lists, Display Lists and Modelling, Programming Event Driven Input, Menus Picking, Building Interactive Models, Animating Interactive programs, Design of Interactive programs, Logic operations .Curved surfaces, quadric surfaces, OpenGL Quadric-Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions. Corresponding openGL functions. **Text-1:Chapter :8-3 to 8-6 (Excluding 8-5),8-9,8-10,8-11,3-8,8-18,13-11,3-2,13-3,13-4,13-10** **Text-2:Chapter 3: 3-1 to 3.11: Input& interaction**	**10 Hours**

Course outcomes: The students should be able to:

- 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:

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011
2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008

Reference Books:

1. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education
2. Xiang, Plastock : Computer Graphics , sham's outline series, 2nd edition, TMG.
3. Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning
4. M M Raiker, Computer Graphics using OpenGL, Filip learning/Elsevier

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

Subject Code	15CS63	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

- 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	**Teaching Hours**
Introduction to System Software, Machine Architecture of SIC and SIC/XE. **Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. **Macroprocessors:** Basic macro processor functions, machine independent macro processor features, Macro processor design options, implementation examples **Text book 1: Chapter 1: (1.1-1.3.2), Chapter2: 2.1- 2.4 ,Chapter4**	**10 Hours**
Module – 2	
Loaders and Linkers: Basic Loader Functions**,** Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options, Implementation Examples. **System File and Library Structure**: Introduction, Library And File Organization, Design Of A Record Source Program File Structure, Object Code, Object File, Object File Structure, Executable File, Executable File Structure, Libraries, Image File Structure . **Text book 1 : Chapter 3 , Reference 1: Chapter 5**	**10 Hours**
Module – 3	
Lexical Analysis: Introduction, Alphabets And Tokens In Computer Languages, Representation, Token Recognition And Finite Automata, Implementation, Error Recovery. **Text book 2: Chapter 1(1.1-1.5), Chapter 3(3.1-3.5)**	**10 Hours**
Module – 4	
Syntax Analysis: Introduction, Role Of Parsers, Context Free Grammars, Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing **Text book 2: Chapter 4 (4.1 – 4.6)**	**10 Hours**
Module – 5	
Syntax Directed Translation, Intermediate code generation, Code generation **Text book 2: Chapter 5 (5.1-5.2),6.1-6.4,8.1-8.7**	**10 Hours**

Course outcomes: The students should be able to:

- 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.

Text Books:

1. System Software by Leland. L. Beck, D Manjula, 3rd edition, 2012
2. Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2nd edition, 2007

Reference Books:

1. Systems programming – Srimanta Pal , Oxford university press, 2016
2. System programming and Compiler Design, K C Louden, Cengage Learning
3. System software and operating system by D. M. Dhamdhere TMG
4. Compiler Design, K Muneeswaran, Oxford University Press 2013.

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

Subject Code	15CS64	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

- 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	**Teaching Hours**
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	**10 Hours**
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.	**10 Hours**
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.	**10 Hours**
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.	**10 Hours**
Module – 5	
Secondary Storage Structures, Protection: Mass storage structures; Disk	**10 Hours**

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:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.

Reference Books

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw-Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

DATA MINING AND DATA WAREHOUSING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS651	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

- 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	**Teaching Hours**
Data Warehousing & modeling: Basic Concepts: Difference between Operational Database systems and Data warehouse, Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse ,Data mart and virtual warehouse, Extraction, Transformation and loading, Metadata Repository, Data warehouse design and usage: Business Analysis framework, Data warehouse design process and usage for information processing, Online analytical processing to multidimensional data mining. 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.	**8 Hours**
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,	**8 Hours**
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.	**8 Hours**
Module – 4	
Classification : Basics: General approach to solve classification problem, Decision Trees Induction, Model Over fitting, Evaluating the performance of a classifier, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers.	**8 Hours**
Module – 5	
Clustering Analysis: Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms.	**8 Hours**

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.

Text Books:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson, First impression,2014.
2. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.

Reference Books:

1. Sam Anahory, Dennis Murray: Data Warehousing in the Real World, Pearson,Tenth Impression,2012.
2. Michael.J.Berry,Gordon.S.Linoff: Mastering Data Mining , Wiley Edition, second edtion,2012.

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	**Teaching Hours**
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	**8 Hours**
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.	**8 Hours**
Module – 3	
Design Pattern Catalog: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.	**8 Hours**
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.	**8 Hours**
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.	**8 Hours**

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.

Text Books:

1. Object-oriented analysis, design and implementation, brahma dathan, sarnath rammath, universities press,2013
2. Design patterns, erich gamma, Richard helan, Ralph johman , john vlissides ,PEARSON Publication,2013.

Reference Books:

1. Frank Bachmann, ReginеMeunier, Hans Rohnert “Pattern Oriented Software Architecture” –Volume 1, 1996.
2. William J Brown et al., "Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998.

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

Subject Code	15CS653	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

- 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	**Teaching Hours**
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.	**8 Hours**
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.	**8 Hours**
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.	**8 Hours**
Module – 4	
Transportation and Assignment Problems: The transportation problem, Initial Basic Feasible Solution (IBFS) by North West Corner Rule method, Matrix Minima Method, Vogel's Approximation Method. Optimal solution by Modified Distribution Method (MODI). The Assignment problem; A Hungarian algorithm for the assignment problem. Minimization and Maximization varieties in transportation and assignment problems.	**8 Hours**
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.	**8 Hours**

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.

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:

1. D.S. Hira and P.K. Gupta, Operations Research, (Revised Edition), Published by S. Chand & Company Ltd, 2014

Reference Books:

1. S Kalavathy, Operation Research, Vikas Publishing House Pvt Limited, 01-Aug-2002
2. S D Sharma, Operation Research, Kedar Nath Ram Nath Publishers.

DISTRIBUTED COMPUTING SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CS654	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

- 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	**Teaching Hours**
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:

1. George Coulouris, Jean Dollimore and Tim Kindberg: Distributed Systems – Concepts and Design, 5th Edition, Pearson Publications, 2009

Reference Books:

1. Andrew S Tanenbaum: Distributed Operating Systems, 3rd edition, Pearson publication, 2007
2. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2008
3. Sunita Mahajan, Seema Shan, " Distributed Computing", Oxford University Press,2015

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

Subject Code	15CSL67	IA Marks	20
Number of Lecture Hours/Week	01I + 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: $A \rightarrow aBa$, $B \rightarrow bB \mid \varepsilon$. 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: $E \rightarrow E+T \mid T$, $T \rightarrow T*F \mid F$, $F \rightarrow (E) \mid id$ 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 * (C +D)*** whose intermediate code in three-address form:

T1 = -B
T2 = C + D
T3 = T1 + T2
A = T3

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.
10. a) Design, develop and implement a C/C++/Java program to simulate a ***numerical calculator***
    b) Design, develop and implement a C/C++/Java program to simulate ***page replacement technique***

    Note: In Examination, for question No 10: Students may be asked to execute any one of the above (10(a) or 10(b)- Examiner choice)

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**

COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – VI

Subject Code	15CSL68	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

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 Brenham'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+05 = 30 Marks
7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

Reference books:

1. Donald Hearn & Pauline Baker: Computer Graphics-OpenGL Version,3rd Edition, Pearson Education,2011
2. Edward Angel: Interactive computer graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2011
3. M M Raikar, Computer Graphics using OpenGL, Fillip Learning / Elsevier, Bangalore / New Delhi (2013)