5. Write a program to implement a B-Tree.
6. Write a program to implement the Tree Data structure, which supports the following operations:
   I. Insert
   II. Search
7. Write a program to search a pattern in a given text using the KMP algorithm.
8. Write a program to implement a Suffix tree.

DISCIPLINE SPECIFIC CORE COURSE – 16 (DSC-16): Theory of Computation

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
DSC 16 Theory of Computation	**4**	**3**	**0**	**1**	Pass in Class XII	DSC04 Object Oriented Programming with C++ / GE1a Programming using C++ /A course in C/C++ at plus 2 level

Learning Objectives

This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine; and their relationships with formal languages. make students aware of the notion of computation using abstract computing devices. Students will also learn about the limitations of computing machines as this course addresses the issue of which problems can be solved by computational means (decidability vs undecidability

Learning outcomes

On successful completion of the course, students will be able to:

- design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.

- apply pumping lemma to prove that a language is non-regular/non-context-free.
- describe limitations of a computing machines and
- recognize what can be solved and what cannot be solved using these machines.

SYLLABUS OF DSC 14

Unit 1 (7 hours)

Introduction: Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

Unit 2 (15 hours)

Finite Automata and Regular: Regular expressions, Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NFA), relationship between NFA and DFA, Transition Graphs (TG), properties of regular languages, the relationship between regular languages and finite automata, pumping lemma, Kleene's theorem.

Unit 3 (15 hours)

Context-Free Languages (CFL): Context-Free Grammars (CFG), deterministic and non-deterministic Pushdown Automata (PDA), relationship between CFG and PDA, parse trees, leftmost derivation, Ambiguities in grammars, pumping lemma for CFL, properties of CFL, Chomsky Normal Form.

Unit 4 (8 hours)

Turing Machines and Models of Computations: Turing machine as a model of computation, configuration of Turing machine, Recursive and recursively enumerable languages, Church Turing Thesis, Universal Turing Machine, decidability, Halting problem.

Essential/recommended readings

1. Harry R. Lewis and Christos H. Papadimitriou, *Elements of the Theory of Computation*, 2nd Edition, Prentice Hall of India (PHI), 2002
2. Daniel I.A. Cohen, *Introduction to Computer Theory*, 2nd Edition, Wiley India Pvt. Ltd., 2011.

Additional References

1. J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 3rd edition, Addison Wesley, 2006.
2. Peter Linz, *An Introduction to Formal Languages and Automata*, 6th edition, Jones & Bartlett Learning, 2017.
3. Michael Sipser, *Introduction to the Theory of Computation*, Cengage, 2014

DSC17 / **DSC-A6/DSE: DEEP LEARNING**

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Deep Learning	**4**	**3**	**0**	**1**	Pass in Class XII	Programming using Python/Object Oriented Programming using Python/Mathematics for Computing

Course Objectives

The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Learning outcomes

On successful completion of this course, the student will be able to:

• Describe the feed-forward and deep networks.

• Design single and multi-layer feed-forward deep networks and tune various hyper parameters.

• Implement deep neural networks to solve a problem

• Analyze performance of deep networks.

• Use pre-trained models to solve a problem.

SYLLABUS

Unit 1 **(8 Hours)**

Introduction to neural networks:

Artificial neurons, perceptron, computational models of neurons, Structure of neural networks, Multilayer feedforward neural networks (MLFFNN), Backpropagation learning, Empirical risk minimization, bias-variance tradeoff, Regularization, output units: linear, softmax , hidden units:tanh, RELU

Unit 2 **(8 Hours)**

Deep neural networks:

Difficulty of training DNNs, Greedy layerwise training, Optimization for training DNN's, Newer optimization methods for neural networks(AdaGrad, RMSProp, Adam), Regularization methods(dropout, drop connect, batch normalization).

Unit 3 **(8 Hours)**

Convolution neural networks(CNNs):

Introduction to CNN - convolution, pooling, Deep CNNs - LeNet, AlexNet. Training CNNs, weights initialization, batch normalization, hyperparameter optimization, Using a pre trained convnet

Unit 4 **(8 Hours)**

Recurrent neural networks (RNNs):

Sequence modeling using RNNs, Backpropagation through time, LongShort Term Memory (LSTM), Bidirectional RNN

Unit 5 **(8 Hours)**

Unsupervised deep learning:

Autoencoders, Generative Adversarial Networks.

Unit 6 **(5 Hours)**

Applications:

Computer vision, Speech recognition and NLP.

Essential/recommended readings

1. Ian Goodfellow, Yodhua Bengio and Aaron Courville, Deep Learning, MIT Press Book, 2016.

2. Francois Chollet, Deep Learning with python, 2nd edition, Meaning Publications Co, 2021.

Additional References

1. Bunduma, N., Fundamentals of Deep Learning, 1st edition, O'reilly Books, 2017.

2. Heaton, J., Deep Learning and Neural Networks, 1st edition, Heaton Research Inc., 2015.

Suggested Practical List :

Practical exercises such as

The following practicals are to be conducted using Python.

1. Implement a feed-forward neural networks for classifying movie reviews as positive or negative(using IMDB dataset)
2. Implement a deep-neural feed-forward network for estimating the price of house, given real-estate data(Boston Housing Price)
3. Implement a deep-neural network for classifying news wires by topic (Reuters dataset).
4. Implement CNN for classifying MNIST dataset
5. Create a model for time-series forecasting using RNN/LSTM 6. Implement an auto-encoder

DSE: NUMERICAL OPTIMIZATION

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course	Eligibility criteria	

DISCIPLINE SPECIFIC ELECTIVE COURSE: Computer Graphics

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Computer Graphics	**4**	**3**	**1**	**0**	Pass in Class XII	DSC 03 (Mathematics for Computing - I), DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b Java Programming

Learning Objectives

This course introduces fundamental concepts of Computer Graphics with focus on modeling, rendering and interaction aspects of computer graphics. The course emphasizes the basic principles needed to design, use and understand computer graphics system.

Learning outcomes

On successful completion of the course, students will be able to:

- Describe Standard raster and vector scan devices as well as Graphical Input and output devices
- Implement algorithms for drawing basic primitives such as line, circle and ellipse.
- Implement algorithms for line clipping, polygon clipping and polygon filling.
- Implement a 3D object representation scheme, carryout 2D and 3D transformation, 3D projections
- Implement visible surface determination algorithms, Illumination models and surface rendering methods
- Implement a simple computer animation algorithm

SYLLABUS OF DSE

Unit 1 (8 Hours)

Introduction: Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics. Architecture of Raster and Random scan display devices, input/output devices.

Unit 2 (8 Hours)
Drawing and clipping primitives: Raster scan line, circle and ellipse drawing algorithms, Polygon filling, line clipping and polygon clipping algorithms

Unit 3 (12 Hours)
Transformation and Viewing: 2D and 3D Geometric Transformations, 2D and 3D Viewing transformations (Projections- Parallel and Perspective), Vanishing points.

Unit 4 (9 Hours)
Geometric Modeling: Polygon Mesh Representation, Cubic Polynomial curves (Hermite and Bezier).

Unit 5 (8 Hours)
Visible Surface determination and Surface Rendering: Z-buffer algorithm, List-priority algorithm and area subdivision algorithm for visible surface determination. Illumination and shading models, RGB Color model and Basics of Computer Animation.

Essential/recommended readings

1. Hearn, D & Baker, M.P. *Computer Graphics*, 2nd edition, Prentice Hall of India, 2009.
2. Foley, J. D., Dam, A.V, Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice in C*, 2nd edition, Pearson education, 2002.
3. Rogers, D. F. *Mathematical Elements for Computer Graphics*, 2nd edition, McGraw Hill Education, 2017.

Additional References

1. Bhattacharya, S. *Computer Graphics*, Oxford University Press, 2018.
2. Marschner, S., & Shirley, P. *Fundamentals of Computer Graphics,* 4th edition CRC Press, 2017.

Suggested Practical List :

Practical exercises such as

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement a midpoint circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to fill a polygon using the Scan line fill algorithm.
6. Write a program to apply various 2D transformations on a 2D object (use homogeneous Coordinates).
7. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
8. Write a program to draw Hermite /Bezier curve.