

## SEMESTER-5

### DEPARTMENT OF COMPUTER SCIENCE [UG Programme for Bachelor in Computer Science (Honours)]

#### DISCIPLINE SPECIFIC CORE COURSE - 13 (DSC-13) : Algorithms and Advanced Data Structures

##### 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		
<b>DSC 13</b> <b>Algorithms and Advanced Data Structures</b>	4	3	0	1	Pass in Class XII	DSC 07 Data Structures with C++, DSC 10 Design and Analysis of Algorithms

#### Learning Objectives

This course is designed to build upon the fundamentals in data structures and algorithm design and gain exposure to more data structures and algorithms for new problems.

#### Learning outcomes

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

- Comprehend and use data structures for lists.
- Use hash tables for dictionaries.
- Comprehend and use data structures and algorithms for string matching.
- Apply disk based data structures.
- Implement and analyze advanced data structures and algorithms for graphs.
- Describe the purpose of randomization in data structures and algorithms.

#### Unit 1 (4 hours)

**List and Iterator ADTs:** Vectors, Lists, Sequences

## **Unit 2 (6 hours)**

**Hash Tables, Dictionaries:** Hash Functions, Collision resolution schemes.

## **Unit 3 (8 hours)**

**Strings:** String Matching: KMP algorithm; Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engines

## **Unit 4 (8 hours)**

**More on Trees:** 2-4 Trees, B Trees

## **Unit 5 (8 hours)**

**More on Graphs:** Bellman Ford Algorithm, Union Find Data Structures - application Kruskal's algorithm

## **Unit 6 (6 hours)**

**Randomization:** Randomized Quicksort, Randomized Select, Skip lists

## **Unit 7 (5 hours)**

**Network Flows:** Ford Fulkerson algorithm for max flow problem.

## **Essential/recommended readings**

1. Goodrich, M.T, Tamassia, R., & Mount, D. *Data Structures and Algorithms Analysis in C++*, 2nd edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4th edition, Prentice Hall of India, 2022.
3. Kleinberg, J., Tardos, E. *Algorithm Design*, 1st edition, Pearson, 2013.
4. Drozdek, A. *Data Structures and Algorithms in C++*, 4th edition, Cengage Learning. 2012.

## **Practical List : (30 Hours)**

Practical exercises such as

1. Write a program to sort the elements of an array using Randomized Quick sort (the program should report the number of comparisons).
2. Write a program to find the  $i$ th smallest element of an array using Randomized Select.
3. Write a program to determine the minimum spanning tree of a graph using Kruskal's algorithm.
4. Write a program to implement the Bellman Ford algorithm to find the shortest paths from a given source node to all other nodes in a graph.

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 – 14 (DSC-14): 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		
<b>DSC 14</b> <b>Theory of Computation</b>	<b>4</b>	<b>3</b>	<b>0</b>	<b>1</b>	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.

10. Case Studies: Students can analyze case studies of privacy breaches or successful privacy protection strategies, and identify key lessons and takeaways.

## **DSC17/DSC-A5/GE7c: MACHINE LEARNING**

**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		
Machine Learning	4	3	0	1	Pass in Class XII	Programming using Python/Object Oriented Programming using Python

### **Course Objectives**

The course aims at introducing the basic concepts and techniques of machine learning so that a student can apply machine learning techniques to a problem at hand.

### **Learning outcomes**

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

- Differentiate between supervised and unsupervised learning tasks.
- State the need of preprocessing, feature scaling and feature selection.
- Formulate classification, regression and clustering problems as optimization problems
- Implement various machine learning algorithms learnt in the course.

**SYLLABUS****Unit 1 (5 Hours)****Introduction:**

Basic definitions and concepts, key elements, supervised and unsupervised learning, applications of ML.

**Unit 2 (8 Hours)****Preprocessing:**

Feature scaling, feature selection methods. dimensionality reduction (Principal Component Analysis), class balancing, outlier detection and removal.

**Unit 3 (12 Hours)****Regression:**

Linear regression with one variable, linear regression with multiple variables, gradient descent, over-fitting, regularization. Regression evaluation metrics.

**Unit 4 (12 Hours)**

**Classification:** Decision trees, Naive Bayes classifier, logistic regression, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM). Classification evaluation metrics

**Unit 5 (8 Hours)**

**Clustering:** Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering.

**Essential/recommended readings**

1. Mitchell, T.M. Machine Learning, McGraw Hill Education, 2017.
2. James, G., Witten. D., Hastie. T., Tibshirani., R. An Introduction to Statistical Learning with Applications in R, Springer, 2014.
3. Alpaydin, E. Introduction to Machine Learning, MIT press, 2009.

## **Additional References**

1. Flach, P., Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2015.
2. Christopher & Bishop, M., Pattern Recognition and Machine Learning, New York: Springer-Verlag, 2016.
3. Sebastian Raschka, Python Machine Learning, Packt Publishing Ltd, 2019

## **Suggested Practical List:**

### **Practical exercises such as**

Use Python for practical labs for Machine Learning. Utilize publicly available datasets from repositories like <https://data.gov.in/> and <https://archive.ics.uci.edu/ml/datasets.php>

For evaluation of the regression/classification models, perform experiments as follows:

- Scale/Normalize the data
- Reduce dimension of the data with different feature selection techniques
- Split datasets into training and test sets and evaluate the decision models
- Perform k-cross-validation on datasets for evaluation

Report the efficacy of the machine learning models as follows:

- MSE and R2 score for regression models
- Accuracy, TP, TN, FP, TN, error, Recall, Specificity, F1-score, AUC for classification models

For relevant datasets make prediction models for the following

1. Naïve Bayes Classifier
2. Simple Linear Regression multiple linear regression
3. Polynomial Regression
4. Lasso and Ridge Regression
5. Logistic regression
6. Artificial Neural Network
7. k-NN classifier
8. Decision tree classification

9. SVM classification

10. K-Means Clustering

11. Hierarchical Clustering

## **DSC16/GE6e/DSE: ARTIFICIAL INTELLIGENCE**

### **CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE**

Course title & Code	Credit s	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Artificial Intelligence	4	3	0	1	Pass in Class XII	Programming using C++/Programming using Python/Object Oriented Programming using Python

### **Course Objectives**

The objectives of this course are to:

- To introduce basic concepts and techniques of Artificial Intelligence (AI).
- To apply informed search techniques for different applications.
- To learn various knowledge representation techniques and writing Prolog programs.
- To learn about the latest techniques for developing AI systems.

### **Learning outcomes**

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

## DISCIPLINE SPECIFIC CORE COURSE- 15 (DSC-15): Software Engineering

### 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		
<b>DSC 15</b> <b>Software</b> <b>Engineering</b>		<b>3</b>	<b>0</b>	<b>1</b>	Pass in Class XII	DSC01 Programming using Python/ DSC04 Object Oriented Programming with C++/A course in C/C++ or Python at plus 2 level

### Learning Objectives

This course will acquaint the student with different approaches and techniques used to develop good quality software. The course includes learning of various software development process frameworks, requirement analysis, design modeling, qualitative and quantitative software metrics, risk management, and testing techniques.

### Learning outcomes

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

- describe the software development models.
- analyse and model customer requirements and build design models.
- estimate and prepare schedule for software projects.
- analyse the impact of risks involved in software development.
- design and build test cases, and perform software testing.

### SYLLABUS OF DSC 15

#### Unit 1 (9 hours)

**Introduction:** Software Engineering - A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile, Agile Model – Scrum.

### **Unit 2 (6 hours)**

**Software Requirements Analysis and Specification:** Use Case Approach, Software Requirement Specification Document, Flow-oriented Model, Data Flow Model

### **Unit 3 (8 hours)**

**Design Modeling:** Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Structure Charts.

### **Unit 4 (7 hours)**

**Software Metrics and Project Estimation:** Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations); Project Scheduling (Timeline charts, tracking the schedule).

### **Unit 5 (5 hours)**

**Quality Control and Risk Management:** Quality Control and Quality Assurance, Software Process Assessment and Improvement; Software Risks, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management.

### **Unit 6 (10 hours)**

**Software Testing:** Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing.

### **Essential/recommended readings**

1. Pressman, R.S. *Software Engineering: A Practitioner's Approach*, 9th edition, McGraw-Hill, 2020.
2. Aggarwal, K.K., Singh, Y. *Software Engineering*, 3rd edition, New Age International Publishers, 2007.
3. Jalote, P. *An Integrated Approach to Software Engineering*, 3rd Edition, Narosa Publishing House, 2005.

### **Additional References**

1. Sommerville, I. *Software Engineering*, 9th edition, Addison Wesley, 2011.
2. *The Definitive Guide to Scrum: The Rules of the Game*, Ken Schwaber, Jeff Sutherland, July 2016.

### **Suggested Practical List :(30 Hours)**

#### **Practical exercises such as**

The students document, design and code a module of a Software Project using an appropriate Software Process model. The Software Project should include the use of software engineering tools and include.

1. Problem Statement, Process Model
2. Requirement Analysis: Create Data Flow, Data Dictionary, Use Cases, Sequence Diagram, Software Requirement Specification Document
3. Project Management: Timeline Chart, Compute FP, Effort, Cost, Risk Table.
4. Design Engineering: Architectural Design, Pseudocode of a small module.
5. Coding: Develop at least a single module using any programming Language
6. Testing: Compute Basic path set for at least one module from a project, Generate test cases.

Some of the sample projects are given below:

1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers
2. DTC Route Information: Online information about the bus routes and their frequency and fares.
3. Car Pooling: To maintain a web-based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation System
8. Automatic Internal Assessment System
9. Parking Allocation System