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		
DSC 13 Algorithms and Advanced Data Structures	**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 ith 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		
DSC 14 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.

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/ bject 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	Credits	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		
DSC 15 Software Engineering		**3**	**0**	**1**	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

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		
DSC 13 Algorithms and Advanced Data Structures	**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 ith 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		
DSC 14 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.

7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using the Prim's algorithm.
10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

For the algorithms at S.No 1, 2 and 3 , test run the algorithm on 100 different input sizes varying from 30 to 1000. For each size find the number of comparisons averaged on 10 different input instances; plot a graph for the average number of comparisons against each input size. Compare it with a graph of nlogn.

DSC-A3/DSE: DATA MINING-I

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 (if any)
		Lecture	Tutorial	Practical/ Practice		
Data Mining - I	4	3	0	1	Passed 12th class with Mathematics	Programming using Python

Course Objectives

This course aims to introduce data mining techniques and their application on real-life datasets. The students will learn to pre-process the dataset and make it ready for application of data mining techniques. The course will focus on three main techniques of data mining i.e. Classification, Clustering and Association Rule Mining. Different algorithms for these techniques

will be discussed along with appropriate evaluation metrics to judge the performance of the results delivered.

Learning outcomes

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

● Pre-process the data for subsequent data mining tasks

● Apply a suitable classification algorithm to train the classifier and evaluate its performance.

● Apply appropriate clustering algorithm to cluster the data and evaluate clustering quality

● Use association rule mining algorithms and generate frequent item-sets and association rules

Syllabus

Unit 1 **(8 hours)**

Introduction to Data Mining:

Motivation and Challenges for data mining, Types of data mining tasks, Applications of data mining, Data measurements, Data quality, Supervised vs. unsupervised techniques

Unit 2 **(9 hours)**

Data Pre-Processing:

Data aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, variable transformation.

Unit 3 **(11 hours)**

Cluster Analysis:

Basic concepts of clustering, measure of similarity, types of clusters and clustering methods, K-means algorithm, measures for cluster validation, determine optimal number of clusters

Unit 4 **(8 hours)**

Association Rule Mining:

Transaction data-set, frequent itemset, support measure, rule generation, confidence of association rule, Apriori algorithm, Apriori principle

Unit 5 **(9 hours)**

Classification:

Naive Bayes classifier, Nearest Neighbour classifier, decision tree, overfitting, confusion matrix, evaluation metrics and model evaluation.

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. Introduction to Data Mining,2nd edition, Pearson, 2021.

2. Han J., Kamber M. and Pei J. Data Mining: Concepts and Techniques, 3 rd edition, 2011, Morgan Kaufmann Publishers.

3. Zaki M. J. and Meira J. Jr. Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd edition, Cambridge University Press, 2020.

Additional References

1. Aggarwal C. C. Data Mining: The Textbook, Springer, 2015.
2. Dunham M. Data Mining: Introductory and Advanced Topics, 1st edition, Pearson Education India, 2006.

Recommended Datasets for :

Classification: Abalone, Artificial Characters, Breast Cancer Wisconsin (Diagnostic)

Clustering: Grammatical Facial Expressions, HTRU2, Perfume data

Association Rule Mining: MovieLens, Titanics

Practicals

1. Apply data cleaning techniques on any dataset (e,g, wine dataset). Techniques may include handling missing values, outliers, inconsistent values. A set of validation rules can be prepared based on the dataset and validations can be performed.
2. Apply data pre-processing techniques such as standardization/normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset

3. Run Apriori algorithm to find frequent itemsets and association rules on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns a) Use minimum support as 50% and minimum confidence as 75% b) Use minimum support as 60% and minimum confidence as 60 % I.
4. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations: a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3% II. Training set is chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained. Data is scaled to standard format.
5. Use Simple K-means algorithm for clustering on any dataset. Compare the performance of clusters by changing the parameters involved in the algorithm. Plot MSE computed after each iteration using a line plot for any set of parameters.

Project: Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSE-A4/DSE: DATA MINING-II

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 (if any)
		Lecture	Tutorial	Practical/ Practice		

2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery
4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

GE6d/DSE: DATA PRIVACY

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		
Data Privacy	4	3	0	1	Pass in Class XII	NIL

Objective:

This course aims to provides students with the ability to identify privacy related aspects of data uses, attacks on data privacy, evaluate proposed technical mechanisms for privacy protection and understand ethical issues related to data privacy

Course Learning Outcomes:

By the end of this course, students will be able to:

- Understand the basic principles of data privacy and the implications of data breaches.
- Identify and evaluate different methods of protecting sensitive data.

- Explain the role of privacy regulations in safeguarding personal information.
- Implement basic cryptographic techniques to secure data.
- Apply data anonymization techniques to protect personal information.
- Analyze the ethical considerations in data privacy.

Syllabus

Unit 1: Introduction to Data Privacy and Privacy Regulations

Definition of data privacy, Historical context of data privacy, Types of sensitive data, Privacy laws and regulations

Unit 2: Data Privacy Attacks, Cryptography and Data Protection

Type of Attacks/ Data Breaches on Data Privacy, Impact of Data Breaches / Attacks, Introduction to cryptography, Symmetric and asymmetric encryption, Hashing and digital signatures

Unit 3: Data Collection, Use and Reuse

Harms Associated with Data collections, use and reuse, Introduction to data anonymization, Data Anonymization Techniques for anonymizing data, Challenges in anonymizing data

Unit 4: Ethical considerations in Data Privacy

Privacy and Surveillance, Ethics of Data Collection and Use, Bias and discrimination in data analysis

References:

1. Ronald Leenes, Rosamunde van Brakel, and Serge Gutwirth: Data Protection and Privacy: The Age of Intelligent Machines, Hart Publishing, 2017.

2. Naavi: Personal Data Protection Act of India (PDPA 2020) : Be Aware, Be Ready and Be Compliant, 2020.

3. Ravinder Kumar Gaurav Goyal, The Right to Privacy in India: Concept and Evolution, Publisher: Lightning Source, 2016.

Additional References:

1. https://onlinecourses.nptel.ac.in/noc22_cs37/preview
2. https://www.coursera.org/learn/northeastern-data-privacy/home/info

Suggested Practicals:

Students may be asked to perform some of the following practical activities related to data privacy:

1. Data Privacy Audit: Students can conduct a data privacy audit of a company or organization to identify potential vulnerabilities and risks in their data privacy practices.

2. Privacy Impact Assessment: Students can conduct a privacy impact assessment (PIA) of a new technology or system to identify potential privacy risks and develop strategies to mitigate them.

3. Regulation Compliance: Students can explore the requirements of the Data Protection Regulations and develop a plan for ensuring compliance with the regulation.

4. Cryptography: Students can learn about different cryptographic techniques and tools, such as encryption, hashing, and digital signatures, and implement them in practice.

5. Anonymization Techniques: Students can learn about data anonymization techniques, such as k-anonymity, differential privacy, and data masking, and apply them to a real-world dataset.

6. Privacy Policy Analysis: Students can analyze the privacy policies of different companies and identify gaps or areas for improvement.

7. Privacy-Enhancing Technologies: Students can explore privacy-enhancing technologies (PETs), such as virtual private networks (VPNs), Tor, and secure messaging apps, and evaluate their effectiveness in protecting privacy.

8. Privacy Breach Response Plan: Students can develop a privacy breach response plan for a company or organization, including steps to take in the event of a data breach and strategies for communicating with affected parties.

9. Ethical Considerations: Students can explore ethical considerations in data privacy, such as the balance between privacy and security, the impact of data collection and analysis on marginalized communities, and the role of data ethics in technology development.

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/ bject 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.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Introduction to Web Programming

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 (if any)
		Lecture	Tutorial	Practical/ Practice		
Introduction to web programming	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Learning Objectives

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple websites using HTML, CSS and Javascript.

Learning outcomes

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

- Build websites using the elements of HTML.
- Build dynamic websites using the client side programming techniques with CSS, Javascript and jQuery.
- Learn to validate client-side data.

SYLLABUS OF DSE

Unit 1 (5 hours)

Introduction: Introduction to Internet and web design. Basic concepts of web architecture.

Unit 2 (12 hours)

HTML: Introduction to hypertext mark-up language (html), creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames.

Unit 3 (8 hours)

Cascading style sheet (CSS): Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding

properties, margin properties).

Unit 4 (10 hours)

Javascript: Document object model, data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators, validations.

Unit 5 (10 hours)

jQuery and JSON: Introduction to jQuery, syntax, selectors, events. JSON file format for storing and transporting data.

Essential/recommended readings

1. Nixon, R. *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Rielly, 2018.
2. Powell, T.A. *HTML & CSS: The Complete Reference, 5th edition*, Tata McGrawHill, 2010.
3. Duckett, J. *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley, 2014.

Additional References

1. Minnick, J. *Web Design with HTML5 and CSS3*, 8th edition, Cengage Learning, 2015.
2. Boehm, A., & Ruvalcaba, Z. *Munarch's HTML5 and CCS,* 4th edition, Mike Murach & Associates, 2018.
3. J. A. Ramalho *Learn Advanced HTML 4.0 with DHTML*, BPB Publications, 2007.
4. Ivan Bayross *Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI*, BPB Publications, 2009.

Suggested Practical List (If any): (30 Hours)

Practical exercises such as

HTML

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create an HTML displaying this semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame showing your college's name and logo. Bottom horizontal frame split into two vertical frames. The left frame with hyperlinks to pages related to faculty, courses, student activities, etc. The right frame showing corresponding pages based on the link clicked on the left frame.
5. Create a student registration form using HTML which has the following controls:

I. Text Box
II. Dropdown box
III. Option/radio buttons
IV. Check boxes
V. Reset and Submit button

CSS

Create a webpage for your department with drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors, ID, class.

Javascript

1. Create event driven programs for the following:
    a. Enter a number and on click of a button print its multiplication table.
    b. Print the largest of three numbers entered by the user.
    c. Find the factorial of a number entered by the user.
    d. Enter a list of positive numbers using the prompt terminated by a zero. Find the sum and average of these numbers.
2. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customise the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:
    a. Roll number is a 7-digit numeric value
    b. Name should be an alphabetical value(String)
    c. Non-empty and valid fields like DOB

jQuery and JSON

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery
4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

4. Apply Partitioning Methods, Hierarchical Methods, Density-Based Methods for clustering on a data set and compare the performance of the obtained results using different metrics
5. Create an ensemble using Random Forest and show the impact of bagging and boosting on the performance
6. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations:
   I. a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3%
   II. Training set should be chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained.
   III. Data should be scaled to the standard format.

Project

Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or on a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. This will allow the students to have practical knowledge of how to apply the various skills learned in the subject to a single problem/project.

DISCIPLINE SPECIFIC Elective (DSE 03b): Web Design and Development

Semester 5

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 (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE 03b Web Design and Development	**4**	**3**	**0**	**1**	Pass in Class XII	Knowledge of Structured Query Language (SQL)

Learning Objectives

The course aims at introducing the basic concepts and techniques of complete website-based programming. The student shall be able to develop simple, interactive, and dynamic websites using HTML, Javascript, PHP and database.

Learning outcomes

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

1. Build interactive and dynamic websites.
2. Use the client-side validation techniques using Javascript.
3. Write the server-side programming techniques with PHP for accessing the contents to/from the server.
4. Use GET and POST methods for exchanging data between client and server.
5. Learn to connect PHP with databases, save and retrieve data dynamically.

SYLLABUS OF DSE 03b

Unit 1. Introduction (2 hours)

Introduction to internet and web design. Basic concepts of web architecture.

Unit 2. HTML (11 hours)

Introduction to hypertext mark-up language (HTML), creating web pages, lists, elements of HTML, hyperlinks, tables, forms, inserting images.

Unit 3. Basics of Javascript (12 hours)

Document object model, data types and variables, functions, methods, and events, controlling program flow, client-side form validation.

Unit 4. Introduction to PHP (8 hours)

Basic syntax, defining variables and constants, data types, operators and expression, decision making statements, loop constructs, functions.

Unit 5. Handling HTML Form with PHP (6 hours)

Connecting an HTML form with PHP, submitting data to the server using GET and POST methods, GET vs POST methods.

Unit 6. Database Connectivity (6 hours)

Connectivity with database, database creation, creating tables, insertion and retrieval of the data from the database, data manipulation.

Essential/recommended readings

Suggested References

1. Nixon, R. **Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5,** 2018, O'Rielly.
2. Powell, T.A. (2017). **HTML & CSS: The Complete Reference**. 5th edition, 2017, Tata McGrawHill.
3. Duckett, J. **JavaScript and JQuery: Interactive Front-End Web Development**, 2014, Wiley.
4. Murach J., **Murach's PHP and MySQL,** 2nd Edition, 2014, Mike Murach & Associates.
5. Holzner S. **PHP: The Complete Reference,** 2017, McGraw Hill**.**
6. Ivan Bayross, **Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI**, 2010, BPB Publications.

Suggested Practical List (If any): (30 Hours)

HTML

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Marquee text.
2. Create an HTML document with Ordered and Unordered lists, Images, Internal and External linking.
3. Create an HTML document for displaying the current semester's timetable.
4. Create a student registration form using HTML which has the following controls:
    a. Text Box

b. Dropdown box
c. Option/radio button
d. Check boxes
e. Reset and Submit button.

Javascript

1. Write event-driven programs in JavaScript for the following:
    a. Enter a number and on click of a button print its factors.
    b. Print the smallest of five numbers entered by the user.
    c. Find the factorial of a number entered by the user.
    d. Take a number in an input text box, on click of a button, display its multiplication table.
2. Create a student registration form. Create functions to perform the following checks:
    a. Student id is a 10-digit alphanumeric value
    b. Name should be an alphabetical value (String)
    c. Non-empty fields like Age
3. Create a form containing various HTML elements and perform appropriate validations on each of them while submitting the form.

PHP

1. Write a PHP script to print the sum of even digits of a number.
2. Write a script in PHP to display a Multiplication Table.
3. Design a Student Registration form, using appropriate input fields consisting of following:
    a. Roll Number
    b. First Name
    c. Last Name
    d. Gender
    e. Department
    f. DOB

    Submit, retrieve the form data using the $_POST variable and display it.
4. Write PHP Code to create a database, connect to it, create tables, insert and access their contents.
5. Write PHP code to insert, delete, and retrieve data from the database. Create proper forms for performing the above operations. Display the messages such as "The record is added in the database!" when data is inserted into the database, "A record is deleted from the database" when data is deleted from the database. Use appropriate button names such as Add Data, Delete Data, and Display Data.

2. Privacy Impact Assessment: Students can conduct a privacy impact assessment (PIA) of a new technology or system to identify potential privacy risks and develop strategies to mitigate them.
3. Regulation Compliance: Students can explore the requirements of the Data Protection Regulations and develop a plan for ensuring compliance with the regulation.
4. Cryptography: Students can learn about different cryptographic techniques and tools, such as encryption, hashing, and digital signatures, and implement them in practice.
5. Anonymization Techniques: Students can learn about data anonymization techniques, such as k-anonymity, differential privacy, and data masking, and apply them to a real-world dataset.
6. Privacy Policy Analysis: Students can analyze the privacy policies of different companies and identify gaps or areas for improvement.
7. Privacy-Enhancing Technologies: Students can explore privacy-enhancing technologies (PETs), such as virtual private networks (VPNs), Tor, and secure messaging apps, and evaluate their effectiveness in protecting privacy.
8. Privacy Breach Response Plan: Students can develop a privacy breach response plan for a company or organization, including steps to take in the event of a data breach and strategies for communicating with affected parties.
9. Ethical Considerations: Students can explore ethical considerations in data privacy, such as the balance between privacy and security, the impact of data collection and analysis on marginalized communities, and the role of data ethics in technology development.
10. Case Studies: Students can analyze case studies of privacy breaches or successful privacy protection strategies, and identify key lessons and takeaways.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Unix Network Programming

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	
		Lecture	Tutorial	Practical/ Practice		
Unix Network Programming	**4**	**3**	**0**	**1**	Pass in Class XII	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 the concepts of Internet protocols, ports used during communication, Client/Server concepts and various transport protocols used in computer network applications and services. The objective is to equip the students with technical knowledge of it comprises of the study of the sockets used with TCP and UDP.

Learning outcomes

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

- Describe and analyze the various Internet Transport layer protocols used in TCP AND UDP.
- Comprehend the concepts and structures of both TCP based connection-oriented and UDP based connectionless client server applications.
- Write various real-life client-server applications using socket programming.
- Modify, maintain and extend the present internet client-server applications and write any new type of internet applications to suit the current needs of Internet users.

SYLLABUS OF DSE

Unit 1 (6 hours)

Introduction Basics of Client Server applications, Example of day time client server, concurrent servers, protocols, sockets, port numbers.

Unit 2 (17 hours)

Connection-oriented Socket Applications: Elementary TCP sockets – Socket, connect, bind, listen, accept, fork and exec function, close function, Socket Address Structures, Byte Ordering and Manipulation Functions, TCP Client and Server for Echo, Signal Handling in case of crashing and rebooting of server, Shutdown process function. Socket Options: Getsockopt and stockpot functions, Socket states, Generic socket option.

Unit 3 (15 hours)

Connectionless Socket Applications: TCP-oriented basic concurrent client server applications, UDP oriented Echo client and server application, Handling of errors like lost datagram, Lack of flow control with UDP, determining outgoing interface with UDP.

Unit 4 (7 hours)

Elementary name and Address conversions: Domain Name System, socket functions like gethostbyname, gethostbyname2, gethostbyaddr function, uname function, gethostname function, getservbyname and getservbyport functions.

Essential/recommended readings

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, *Unix Network Programming*, The sockets Networking API, Vol. 1, 3rd Edition, PHI.
2. B. A. Forouzan: *Data Communications and Networking*, THM Publishing Company Ltd
3. R. Stevens, *Unix Network Programming*, PHI 2nd Edition

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Implement TCP Echo client and TCP Echo server (Iterative).
2. Implement TCP Echo client and TCP Echo server (Concurrent).
3. Implement TCP daytime client and TCP daytime server (Iterative).
4. Implement TCP daytime client and TCP daytime server (concurrent).
5. Implement UDP Echo Client and UDP Echo Server.
6. Implement UDP daytime Client and UDP daytime server.
7. Implement TCP client and server (concurrent) where client gets input from the user and sends it to server. Server displays it on the screen. Server then gets another input from the user and sends it to client. Client displays it on the screen. The process continues till server or client sends “bye” to the other party.
8. Implement TCP client and server (concurrent) where client requests server to transfer a file. Assume file is smaller than 1K size. If the file is present on the server, it is sent to the client otherwise an error message is sent to client. Client copies the file on the hard disk and disconnects.
9. Implement UDP client and UDP server where server displays the IP address and port number of the client sending the datagram. Client sends a datagram (size 64 bytes) three times to the same server. Server sends the message back to client. Client reports the time elapsed in sending and receiving of the message. Use connected UDP sockets.
10. Write a program to
    a) display name of the host
    b) all IP addresses of the host.
    c) Check whether FTP and HTTP services are running on the system.
    d) Display the name of the service running on port number specified by user.

DSC08/DSC04/GE5a: OPERATING SYSTEMS

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 (if any)
		Lecture	Tutorial	Practical / Practice		
Operating Systems	4	3	0	1	Passed 12th class with Mathematics	Programming using C++ / Python/Java

Course Objectives

The course provides concepts that underlie all operating systems and are not tied to any particular operating system. The emphasis is on explaining the need and structure of an operating system using its common services such as process management (creation, termination etc.), CPU Scheduling, Process Synchronization, Handling Deadlocks, main memory management, virtual memory, secondary memory management. The course also introduces various scheduling algorithms, structures, and techniques used by operating systems to provide these services.

Learning Outcomes

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

● Describe the need of an operating system and define multiprogramming and Multithreading concepts.
● Implement the process synchronization service (Critical Section, Semaphores), CPU scheduling service with various algorithms.
● Implement Main memory Management (Paging, Segmentation) algorithms, Handling of Deadlocks
● Identify and appreciate the File systems Services, Disk Scheduling service

Syllabus

Unit-1 **(6 hours)**

Introduction: Operating Systems (OS) definition and its purpose, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

Unit-2 **(9 hours)**

Operating System Structures: OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

Unit-3 **(10 hours)**

Process Management: Process Concept, States, Process Control Block, Process Scheduling, Schedulers, Context Switch, Operation on processes, Threads, Multicore Programming, Multithreading Models, Process Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin, Process Synchronization: The critical section problem, Deadlock characterization, Deadlock handling.

Unit-4 **(11 hours)**

Memory Management: Physical and Logical address space, Swapping, Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging. Virtual Memory Management: Demand Paging and Page Replacement algorithms: FIFO Page Replacement, Optimal Page replacement, LRU page replacement.

Unit-5 **(9 hours)**

File System: File Concepts, File Attributes, File Access Methods, Directory Structure: Single Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories. Mass Storage Structure: Magnetic Disks, Solid-State Disks, Magnetic Tapes, Disk Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOk Scheduling.

Essential/recommended readings

1. Silberschatz, A., Galvin, P. B., Gagne G. Operating System Concepts, 9 th edition, John Wiley Publications, 2016.

2. Tanenbaum, A. S. Modern Operating Systems, 3 rd edition, Pearson Education, 2007.
3. Stallings, W. Operating Systems: Internals and Design Principles, 9 th edition, Pearson Education, 2018.

Additional References

1. Dhamdhere, D. M., Operating Systems: A Concept-based Approach, 2nd edition, Tata McGraw-Hill Education, 2017.
2. Kernighan, B. W., Rob Pike, R. The Unix Programming Environment, Englewood Cliffs, NJ: Prentice-Hall, 1984.

Practicals

1. Execute various Linux commands for:
   a. Information Maintenance: wc, clear, cal, who, date, pwd
   b. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep, awk
   c. Directory Management : cd, mkdir, rmdir, ls
2. Execute various Linux commands for:
   a. Process Control: fork, getpid, ps, kill, sleep
   b. Communication: Input-output redirection, Pipe
   c. Protection Management: chmod, chown, chgrp
3. Write a programme (using fork() and/or exec() commands) where parent and child execute:
   a. same program, same code.
   b. same program, different code.
   c. Before terminating, the parent waits for the child to finish its task.
4. Write a program to to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information).
5. Write a program to report behaviour of Linux kernel including information on 19 configured memory, amount of free and used memory. (Memory information)
6. Write a program to copy files using system calls.
7. Use an operating system simulator to simulate operating system tasks.
8. Write a program to implement scheduling algorithms FCFS/ SJF/ SRTF/ non preemptive scheduling algorithms.
9. Write a program to calculate the sum of n numbers using Pthreads. A list of n numbers is divided into two smaller lists of equal size, and two separate threads are used to sum the sublists.
10. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, bc.
4. Write a shell script to display date in the mm/dd/yy format.
5. Write a shell script to display the multiplication table any number.
6. Write a shell script to find the factorial of a given number.
7. Program to show the pyramid of special character "*".
8. Write a shell script to find the sum of digits of a given number.
9. Write a shell script to perform the tasks of basic calculator.
10. Write a shell script to find the power of a given number.
11. Write a shell script to check whether the number is Armstrong or not.
12. Write a shell script to find the GCD (greatest common divisor) of two numbers.
13. Write a shell script to check if the number entered at the command line is prime or not.
14. Write a shell script to display on the screen sorted output of "who" command along with the total number of users.
15. Write a shell script to accept a login name. If not a valid login name display message – "Entered login name is invalid".
16. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
17. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
18. Write a shell script to check whether the file have all the permissions or not.
19. Write a shell script to modify "cal" command to display calendars of the specified months.
20. Write a shell script to modify "cal" command to display calendars of the specified range of months.

GENERIC ELECTIVES (GE-5b): Advanced Web Programming

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		
GE5b: Advanced Web Programming	**4**	**3**	**0**	**1**	Pass in Class XII	Knowledge of HTML, CSS

Learning Objectives

The course aims to familiarize the students with the concepts and techniques of server side web programming. This will enable the students to create dynamically generated web pages using HTML, PHP, MySql and JQuery.

Learning outcomes

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

- develop interactive and dynamic websites.
- write programs to communicate with the server using GET and POST methods
- learn to connect and manipulate databases using MySql
- validate server-side/backend data

SYLLABUS OF GE-5b

Unit 1 (7 hours)

Introduction to PHP: Basic syntax, defining variables and constants, data types including arrays, operators and expressions, decision making statements, constructs for iterations.

Unit 2 (5 hours)

String Handling: Creating a string and accessing its content, searching and replacing content of a string, and other built-in functions.

Unit 3 (12 hours)

Handling HTML Form with PHP: Creating a form, submitting data to the server at the backend using GET and POST methods, GET vs POST methods, PHP global functions.

Unit 4 (15 hours)

Database: Connectivity with MySQL: Connectivity with database, database creation, creating tables, create, retrieve, update, and delete (CRUD) operations

Unit 5 (6 hours)

jQuery and JSON: Introduction to jQuery syntax (selectors, events, AJAX, JSON).

Essential/recommended readings

1. Nixon, R. Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5, O'Reilly, 2018.
2. Holzner S. PHP: The Complete Reference, McGraw Hill, 2017
3. Murach J, Murach's PHP and MySQL, 2nd edition, Mike Murach & Associates, 2014.
4. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development, Wiley, 2014.

Additional References

1. https://www.w3schools.com/php/default.asp

2. https://www.tutorialspoint.com/php/index.htm

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Write a PHP script to reverse the digits of a number.
2. Create a web page containing two text boxes and a button named "Evaluate". When the user enters numbers in the text boxes and clicks on the "Evaluate" button, a function should evaluate the sum of the numbers and display the result.
3. Write a PHP script to perform following string operations using in-built functions and built an interactive web page having buttons for each of the following operation:
    a. Find the length of a string
    b. Find a substring from a string
    c. Replace text within a string
    d. Remove whitespace and other predefined characters from both sides of a string.
    e. Check if a value is a string
    f. Convert the first character of each word in a string into uppercase.
4. Design a Login form and validate that form using PHP code. Display error message box when data is not valid otherwise redirect to the next page and display "Welcome username!".
5. Design a student registration form, using appropriate input fields consisting of following:
    a. First Name
    b. Last Name
    c. Gender
    d. Roll Number
    e. Phone Number
    f. Course

Submit and retrieve the form data using $_POST, $_GET variable.

6. Write PHP Code to make connection to MySql database, create database and tables and perform insertion, deletion, and retrieval of the data (Using SQL operations like .JOIN, ORDER BY, GROUP BY) Display the messages such as "The record is added in the database!" when data is inserted into the database, "A record is deleted from the database" when data is deleted from the database. Use appropriate button names such as Add Data, Delete Data, and Display Data.

jQuery and JSON

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery

4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

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		
GE5c: Java Based Web App Development	**4**	**3**	**0**	**1**	Pass in Class XII	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 / A programming course at class XII level

Learning Objectives

The course aims to familiarize the students with the concepts and techniques of web app development based on Java. The students will learn about database connectivity, use of HTTP protocol, client side programming, and use of servlets and JSP for server side programming.

Learning outcomes

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

- develop an understand of client-server architecture, HTTP protocol, and web application components.
- connect an application to database and perform basic database operations.
- create servlets and JSP for web applications
- deploy web applications

SYLLABUS OF GE-5c

Unit 1 (8 hours)

Review of Programming Language: Programming Constructs, Data types, Operators, Concepts of Class, Interface, Inheritance, Exception Handling, Util package, Multithreading, event handling.

4. Handle HTML form, store the data in JSON object, pass them to another page and display it there using jQuery/Javascript

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		
GE5c: Java Based Web App Development	**4**	**3**	**0**	**1**	Pass in Class XII	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 / A programming course at class XII level

Learning Objectives

The course aims to familiarize the students with the concepts and techniques of web app development based on Java. The students will learn about database connectivity, use of HTTP protocol, client side programming, and use of servlets and JSP for server side programming.

Learning outcomes

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

- develop an understand of client-server architecture, HTTP protocol, and web application components.
- connect an application to database and perform basic database operations.
- create servlets and JSP for web applications
- deploy web applications

SYLLABUS OF GE-5c

Unit 1 (8 hours)

Review of Programming Language: Programming Constructs, Data types, Operators, Concepts of Class, Interface, Inheritance, Exception Handling, Util package, Multithreading, event handling.

Unit 2 (10 hours)

Java Database Connections: Database connectivity, Connection, statement, result set object, Metadata, Connection pooling, CRUD operations, Prepared and callable statements

Unit 3 (15 hours)

Introduction to servlets: Concepts of Streams, events and listener, recap of HTML, CSS, XML, Servlet package and interface, life cycle of servlet, deployment descriptor, Filters, HHTP and Generic servlet, request dispatcher, Request Response classes, Dynamic page designing using servlet.

Unit 4 (12 hours)

Introduction to JSP: JSP Life cycle, tags in JSP, custom tags, Expression Language, Introduction to Struts Framework, Implicit objects, database access using JSP

Essential/recommended readings

1. Herbert Schildt, *Java : The Complete Reference*, 12th edition, McGraw-Hill Education, 2021.
2. Hans Bergsten, *Java Server Pages*, 3rd edition, O′Reilly, 2003.
3. Jim Keogh, *The Complete Reference J2EE*, 1st edition, McGraw-Hill Education, 2017.

Suggested Practical List : (30 Hours)

Practical exercises such as

1. Setting up the development environment: Install Java Development Kit (JDK), Eclipse IDE, and Apache Tomcat web server. Create a new web project in Eclipse.
2. Writing and deploying a "Hello World" servlet: Create a simple servlet that prints "Hello World" on the web page. Deploy the servlet on Tomcat and test it in a web browser.
3. Handling HTTP requests and responses: Write a servlet that reads input from HTTP requests and sends output as HTTP responses.
4. Creating a JSP page: Create a JSP page that displays dynamic.
5. Write a servlet that handles form submissions and saves the data to a database using JDBC.
6. Write a servlet that implements user authentication and authorization using a database.
7. Creating a web application using MVC architecture: Create a web application using Model-View-Controller (MVC) architecture. Use servlets as controllers, JSP pages as views.
8. Deploying a web application to a server: Configure and deploy a web application to a server using Apache Tomcat Manager or other deployment tools.