III- Year I- Semester	Name of the Course	L	T	P	C
PC3101	Automata and Compiler Design	2	1	0	3

Course Objectives: To learn fundamentals of Regular and Context Free Grammars and Languages

- To understand the relation between Regular Language and Finite Automata and machines
- To understand the relation between Contexts free Languages, Push Down Automata and Turing Machine
- To study various phases in the design of compiler and understanding the machine independent phases of compiler
- To understand machine dependent phases of compiler

Syllabus:

UNIT-I: Finite Automata **12hrs**

Need for Automata Theory, Alphabet, Strings, Language, Operations, Deterministic Finite Automata, Non-Deterministic Finite Automata, Design of DFA, Design of NFA, Equivalence of NFA, DFA

Finite Automata Conversions:

Conversion from NFA to DFA, NFA ε to NFA, Minimization of DFA, Moore and Mealy Machines.

UNIT-II: Regular Expressions and Grammars **14hrs**

Regular Expressions: Regular Sets, Identity Rules,Constructing finite Automata for a given regular expressions, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets.

Grammars: Grammars, Classification of Grammars, Regular grammars- Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion, Context Free Grammar, Leftmost and Rightmost Derivations, derivation trees.

Unit-III: Context Free Grammar, Push Down Automata and Turing Machines 12hrs

Context Free Grammar:

Ambiguous Grammars, Simplification of Context Free Grammars, Normal Forms- Chomsky Normal Form, Griebach Normal Form.

Push Down Automata (PDA):

Definition, Model, Design of PDA, Deterministic PDA, Non-deterministic PDA, Equivalence of PDA and Context Free Grammars.

Turing Machine (TM): Definition, Model, Design of Turing Machine, Deterministic TM, Non-deterministic TM.

UNIT-IV: Machine Independent Phases **14hs**

Lexical Analysis: Logical phases of compiler**,** Lexical Analysis, Lexemes Tokens and patterns, Lexical Errors.

Syntax Analysis: Parsing definition, types of parsers, left recursion, left factoring, Top-down parser, First() and Follow(), LL(1) Grammars, Non- Recursive predictive parsing, Bottom-up Parsers, Shift Reduce Parser, LR parsers.

Semantic Analysis: Syntax Directed Translation, L-attributed and S-attributed definitions.

UNIT-V: Machine Dependent Phases **12hrs**

Intermediate Code Generation: Intermediate code, three address code, quadruples, triples, indirect triples, Directed acyclic graph.

Code Optimization: Common sub expression elimination, copy propagation, dead code

elimination, constant folding, strength reduction, loop optimization.
Code Generation: Basic blocks & flow graphs, Peephole optimization.

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008
2. Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, 3rd Edition, PHI, 2007
3. Compilers, Principles Techniques and Tools- Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D. Ullman,2nd ed, Pearson,2007.

Reference Books

1. Elements of Theory of Computation, Lewis H.P. &Papadimition C.H., Pearson /PHI
2. Theory of Computation, V. Kulkarni, Oxford University Press, 2013
3. Principles of compiler design, V. Raghavan, 2nd ed, TMH, 2011.
4. Compiler construction, Principles and Practice, Kenneth C Louden, CENGAGE

Web Resources

1.https://nptel.ac.in/courses/106/104/106104028/
2.https://nptel.ac.in/courses/106/105/106105190/

University AcademyYoutube Channel for Automata Theory and Compiler Design:

1.https://www.youtube.com/playlist?list=PL-JvKqQx2AtdhlS7j6jFoEnxmUEEsH9KH
2.https://www.youtube.com/playlist?list=PL-JvKqQx2Ate5DWhppx-MUOtGNA4S3spT

GATE Lectures:

1.https://www.youtube.com/playlist?list=PLEbnTDJUr_IdM___FmDFBJBz0zCsOFxfK
2.https://www.youtube.com/playlist?list=PLMzYNEvC0P7FwwnrXwAjPq8zLTC4MDQKQ

Course Outcomes: By the end the of the course, the student will be able to

CO1: Classify machines by their power to recognize languages.
CO2: Summarize language classes and grammars relationship among them with the help of Chomsky hierarchy.
CO3: Employ finite state machines in problem solving and also illustrate deterministic and non-deterministic machines.
CO4: Design and implement scanners and parsers.
CO5: Perform code optimization to improve performance and apply algorithms to generate code.

III- Year I- Semester	Name of the Course	L	T	P	C
PC3102	Machine Learning	3	0	0	3

Course Objectives:

1. Familiarity with a set of well-known supervised, unsupervised and semi-supervised learning algorithms.
2. Understanding the machine learning model prediction through classification, scoring and ranking using R.
3. Predict objects classification through decision tree building and rule building.
4. Know the importance of features and perform feature engineering
5. Summarizing the data from large tables into smaller set of summary indices through principal component analysis.

Syllabus:

Unit 1-Introduction to Statistical Learning and Linear Regression

Introduction to Statistical Learning: What Is Statistical Learning? Assessing Model Accuracy.
Linear Regression: Simple Linear Regression, Estimating the Coefficients, Assessing the Accuracy of the Coefficient Estimates, Assessing the Accuracy of the Model.

Unit 2-Multiple Linear Regression and Classification

Multiple Linear Regression: Estimating the Regression Coefficients, Other Considerations in the Regression Model, Comparison of Linear Regression with K-Nearest Neighbours.

Classification: An Overview of Classification, Why Not Linear Regression?Logistic Regression, Generative Models for Classification, A Comparison of Classification Methods.

Unit 3-Resampling Methods, Linear Model Selection and Regularization

ResamplingMethods: Cross-Validation,the Bootstrap.
Linear Model Selection and Regularization, Subset Selection,Shrinkage Methods, Dimension Reduction Methods,Considerations in High Dimensions.

Unit 4-Tree-Based Methods

Tree-Based Methods: The Basics of Decision Trees, Regression Trees, Classification Trees, Trees Versus Linear Models,Advantages and Disadvantages of Trees, Bagging, Random Forests, Boosting and Bayesian AdditiveRegression Trees.

Unit 5-Support Vector Machines and Unsupervised Learning

Support Vector Machines, Maximal Margin Classifier,Support Vector Classifiers, Support Vector Machines.
Unsupervised Learning: The Challenge of Unsupervised Learning, Principal Components Analysis, Missing Values and Matrix Completion, Clustering Methods.

Text Books:

1. Gareth James, et al. An Introduction to Statistical Learning: with Applications in R, Springer. 2nd edition (2021 edition).

Reference Books:

1. Tom Mitchell, "Machine Learning", McGraw Hill, 1997
2. E. Alpaydin, "Introduction to Machine Learning", PHI, 2005.
3. Andrew Ng, Machine learning yearning, https://www.deeplearning.ai/machine-learning-yearning/
4. Hands-on machine learning with R" by Bradley Boehmke & Brandon Greenwell
5. "Machine learning with R, the tidyverse, and mlr" by Hefin I. Rhys

Web Resources:

1. R Programming Crash Course – https://www.youtube.com/watch?v=ZYdXI1GteDE&t=1849s
2. Machine Learning With R | Machine Learning Full Course | Machine Learning Tutorial | Simplilearn - https://www.youtube.com/watch?v=6dEUTmoXz0w
3. Statistics for Data Science | Probability and Statistics | Statistics Tutorial | Ph.D. (Stanford) - https://www.youtube.com/watch?v=Vfo5le26IhY&list=PLlgLmuG_KgbaXMKcISC-fdz7HUn1oKr9i
4. Linear Regression Algorithm | Linear Regression Machine Learning | Linear Regression Full Course - https://www.youtube.com/watch?v=tFi4Y_y-GNM
5. Learning: Support Vector Machines - https://www.youtube.com/watch?v=_PwhiWxHK8o
6. https://nptel.ac.in/courses/106105152

Course Outcomes (CO):

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

1. Explain the differences among the three main styles of learning: reinforcement learning, supervised, and unsupervised learning.
2. Implement the algorithms for supervised learning and unsupervised learning using R.
3. Determine which of the three learning styles is appropriate to a particular problem domain.
4. Be able to work with real-world data and perform machine learning through data analytics
5. Characterize the state of the art in learning theory, including its achievements and its challenges.

III- Year I- Semester	Name of the Course	L	T	P	C
PC3103	**Computer Networks**	3	0	0	3

Course Objectives:

1. To summarize OSI and TCP/IP reference models and Example networks, characteristics of transmission media and classify multiplexing techniques
2. To Compute the Error detecting and Flow control Algorithms
3. To illustrate IEEE LAN,MAN and WAN protocols
4. To Compute optimal path using Routing Algorithms
5. To explain the knowledge on various application layer protocols

Syllabus:

UNIT-I: Introduction to Computer Networks and Physical Layer

Introduction: Network Topologies WAN, LAN, MAN. Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models, Example Networks, Physical Layer – Fourier analysis – Bandwidth Limited Signals – The Maximum Data Rate of a Channel Guided Transmission Media, Multiplexing: Frequency Division Multiplexing, Time Division Multiplexing, and Code Division Multiplexing

UNIT-II: Data Link Layer

Data Link Layer Design Issues, Error Detection and Correction, Elementary Data Link Control Protocols, Sliding Window Protocols, HDLC, PPP, Channel Allocation problem, Multiple Access Protocols, IEEE standards for Local Area Networks, WLAN, Bluetooth

UNIT– III: Network Layer

Network Layer Design Issues, Routing Algorithms, Congestion Control Algorithms, Internet Protocol Header, IP Addresses, subnetting and super netting.

UNIT-IV: Transport Layer

Transport Layer Design Issues, Connection Establishment, Connection Termination, Transport and User Datagram Protocols

UNIT – V: Application Layer

Design Issues, DNS, WWW, HTTP/HTTPS, E-mail, FTP

Text Books:

1. Computer Networks, Andrew S Tanenbaum, Pearson, 5th Edition
2. Data Communications and Networking, Behrouz A Forouzan, Tata McGraw Hill, 4th Edition

Reference Book:

1. TCP/IP Protocol Suite, Behrouz A Forouzan**,** Tata McGraw Hill Edition, 3rd Edition

Web Resources:

1.https://youtube.com/playlist?list=PLbRMhDVUMngfpeFloB7kyiA40EptH1up
2. https://www.geeksforgeeks.org/computer-network-tutorials/
3.https://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html
4.https://nptel.ac.in/courses/106105081

Course Outcomes: By the end the of the course, the student will be able to

CO1: Explain OSI and TCP/IP reference models and Example networks, characteristics of transmission media and classify multiplexing techniques (L2)
CO2: Summarize various Error Control and Flow Control techniques (L2)
CO3: Identify Channel Allocation problems and IEEE protocols w.r.t LAN, WLAN. (L1)
CO4: Compute optimal path using Routing Algorithms and Design the sub networks(L4)
CO5: Illustrate the working of various Transport and application layer protocols (L3)

III- Year I- Semester	Name of the Course	L	T	P	C
PC3101L	**Image Processing Using Python**	3	0	0	3

Course Objectives:

1. Familiarize with basic concepts of digital image processing.
2. Learn various image processing techniques like image enhancement both in spatial and frequency domain
3. Familiarize with basic restoration techniques
4. Understand segmentation and morphological techniques applicable to various tasks
5. Understand the need for compression and familiarize few compression methods

Syllabus:

Unit I: FUNDAMENTALS OF IMAGE PROCESSING AND IMAGE TRANSFORMS

Introduction, Fundamental steps in image processing, Image sampling, Quantization, Resolution, Elements of image processing system, Applications of Digital image processing. Color fundamentals, Color image formats and conversion.

Unit II: IMAGE ENHANCEMENT

Spatial domain methods: Point & Histogram processing, Fundamentals of Spatial filtering, Smoothing spatial filters, Sharpening spatial filters.

Frequency domain methods: Basics of filtering in frequency domain, image smoothing, image sharpening, Selective filtering.

Unit III: IMAGE RESTORATION AND RECONSTRUCTION

A model of the image degradation and Restoration process, Noise models, restoration in the presence of noise only-Spatial Filtering, Periodic Noise Reduction by frequency domain filtering, Linear, Position –Invariant Degradations, Estimating the degradation function, Inverse filtering, Minimum mean square error (Wiener) filtering, constrained least squares filtering.

Unit IV: IMAGE SEGMENTATION

Fundamentals, point, line, edge detection, thresholding, and region –based segmentation.

MORPHOLOGICAL IMAGE PROCESSINGs

Preliminaries, Erosion and dilation, opening and closing, basic morphological algorithms for boundary extraction, thinning.

Unit V:IMAGE COMPRESSION

Introduction, Need for image compression, Redundancy in images, Classification of redundancy in images, image compression scheme, Classification of image compression schemes, Fundamentals of information theory, Run length coding, Shannon – Fano coding, Huffman coding, Arithmetic coding, Predictive coding.

Text books:

1. Digital Image Processing – Gonzalez and Woods, 2nd Ed., Pearson.
2. S. Jayaraman, S. Esakkirajan and T. VeeraKumar, “Digital Image processing, Tata McGraw Hill publishers, 2009

Reference books:

1. Anil K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall of India, 9th Edition, Indian Reprint, 2002.
2. J. T. Tou, R. C. Gonzalez, “Pattern Recognition Principles”, Addison-Wesley, 1974.
3. B. Chanda, D. Dutta Majumder, “Digital Image Processing and Analysis”, PHI, 2009.

Web Resources:

NPTEL Lecture material

1. Lecture Series on Digital Image Processing by Prof. P. K. Biswas, Department of Electrical& Electronic Communication Engineering,IIT Kharagpur. https://www.youtube.com/playlist?list=PLuv3GM6-gsE08DuaC6pFUvFaDZ7EnWGX8

Course Outcomes

Upon successful completion of the course, the student will be able to

CO1 Perform image manipulations and color format conversions.{Understanding level, KL2}

CO2 Apply various spatial and frequency domain techniques for the smoothing and sharpening of images.{Applying level, KL3}

CO3 Describe various image restoration techniques. {Understanding level, KL2}

CO4 Apply various segmentation and morphological operators on images. {Applying level, KL3}

CO5 Analyzethe performance of different image compression techniques. {Analyzing level, KL4}

III- Year I- Semester	Name of the Course	L	T	P	C
PE3101	**Artificial Neural Networks** **(Professional Elective - 1.1)**	2	0	2	3

Course Objectives:

1. To learn data processing and cleaning
2. To understand mathematical model of artificial neuron
3. To learn various activation functions used in neural network
3. To understand training and learning process in neural networks
4. To implement neural network models using python

Syllabus:

Unit-1: Foundations

Linear Algebra- Creating matrices add vectors using NumPy, implementation of operations on matrices-addition, subtraction, multiplication, transpose, inverse, determinant, Vectors- addition, subtraction, dot product, various norms. Linear transformations, pre-processing data using pandas. Scikit Learn-data processing, creating model using scikit-learn.

Unit-2: Introduction to Artificial Neural Network

Biological Model of Neuron, ANN model, McCulloch and Pitts model, Adaline, Perceptron, Activation functions, realizing logic gates using perceptron, implementing perceptron using Python, implementing functionality of logic gates using perceptron in python.

Unit-3: Single Layer Perceptron

Architectural Models for ANN, Single Layer Perceptron Perceptron as a classifier, implementing classification using perceptron, Learning and training ANN, optimization- Gradient descent algorithm, stochastic gradient descent algorithm, implementation of gradient descent using python.

Unit-4-Multilayer Perceptron

Multilayer Perceptron- architecture, functionality of neurons in different layers, implementing multilayer perceptron using scikit-learn, Back propagation algorithm-training and convergence, design issues, example, implementation using python.

Unit-5- Linear, Logistic regression and Classification

Linear and logistic regression using MLP, multivariate regression, implementation of linear and logistic regression using scikit-learn, Function Approximation using MLP, RBF networks, RBF Training. Implementation of classification using ANN with scikit-learn on IRIS dataset.

. **Text Books:**

1. Simon Haykin, "Neural Networks: A comprehensive foundation", Second Edition, Pearson Education Asia.
2. Satish Kumar, "Neural Networks: A classroom approach", Tata McGraw Hill, 2004.
3. Pradhan Manaranjan , U Dinesh Kumar,"Machine Learning Using Python" , wiley.

References:

1. Yegna narayana B,"Artificial Neural Networks", PHI Learning Pvt. Ltd, 2009.
2. Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, Orlando De Jesús, Neural

Network Design, 2nd Edtion (Free Online version available at url 4)
3. Aurélien Géron, Neural networks and deep learning, O'Reilly Media, 2018.

Web Resources:

1. https://numpy.org/doc/stable/user/quickstart.html
2. https://pandas.pydata.org/docs/user_guide/index.html
3. https://scikit-learn.org/stable/modules/neural_networks_supervised.html
4. https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-1-missing-data-45e76b781993
5. https://hagan.okstate.edu/NNDesign.pdf
6. https://nptel.ac.in/courses/117105084
7. https://nptel.ac.in/courses/108108148
8. https://nptel.ac.in/courses/106105152

Course Outcomes: By the end of the course the student will be able to

- Perform data pre processing.
- Able to implement mathematical model of neural network using python.
- Implement training process using training data.
- Improve accuracy and performance by tuning parameters.
- Classify data using neural network models

III- Year I- Semester	Name of the Course	L	T	P	C
PE3101	**Software project Management (Professional Elective - 1.2)**	2	0	2	3

Course Objectives:

1. To study how to plan and manage projects at each stage of the software development life cycle (SDLC).
2. To train software project managers and other individuals involved in software project planning and tracking and oversight in the implementation of the software project management process.
3. To understand successful software projects that support organization's strategic goals.

Syllabus:

UNIT -I:

Conventional Software Management: The waterfall model, conventional software Management performance.

Evolution of Software Economics: Software Economics, pragmatic software cost estimation. Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achievingrequired quality, peer inspections

UNIT -II:

The Old Way and The New: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process. Life Cycle Phases: Engineering and production stages, inception, Elaboration, construction, transition phases. Artifacts of The Process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

UNIT -III:

Model Based Software Architectures: A Management perspective and technical perspective. Work Flows of the Process: Software process workflows, Iteration workflows. Checkpoints of the Process: Major mile stones, Minor Milestones, Periodic status assessments.

UNIT -IV:

Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning. Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

UNIT -V:

Process Automation: Automation Building blocks, The Project Environment. Project Control and Process Instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation. Project Estimation and Management: COCOMO model, Critical Path Analysis, PERT technique, Monte Carlo approach (Text book 2)

Text Books:

1) Software Project Management, Walker Royce, Pearson Education, 2005.
2) Software Project Management, Bob Hughes, 4th edition, Mike Cotterell, TMH.

Reference Books:

1) Software Project Management, Joel Henry, Pearson Education.
2) Software Project Management in practice, Pankaj Jalote, Pearson Education, 2005.
3) Effective Software Project Management, Robert K.Wysocki, Wiley,2006

Web Resource

1.https://nptel.ac.in/courses/106105218

Course Outcomes:

1. To match organizational needs to the most effective software development model
2. To understand the basic concepts and issues of software project management
3. To effectively Planning the software projects
4. To implement the project plans through managing people, communications and change
5. To select and employ mechanisms for tracking the software projects
6. To conduct activities necessary to successfully complete and close the Software projects
7. To develop the skills for tracking and controlling software deliverables

III- Year I- Semester	Name of the Course	L	T	P	C
PE3101	**COMPUTER GRAPHICS (Professional Elective - 1.3)**	2	0	2	3

Course Objectives:

1. To develop, design and implement two and three dimensional graphical structures.
2. To enable students to acquire knowledge Multimedia compression and animations.
3. To learn Creation, Management and Transmission of Multimedia objects.

Syllabus:

UNIT - I

Introduction to Computer Graphics : Applications of Computer Graphics, 2D Primitives:- Output Primitives: Points,Lines,Planes,Frame-Buffers,Video-display devices, Line Drawing Algorithms: DDA Line drawing, Bresenham's Line Drawing ,Parallel Line Drawing ,Circle and Ellipse Generation, Polygon Generation, Polygon Filling Algorithms,Attributes of Output Primitives.

UNIT - II

2D Transformations & Viewing: Basic Transformations: Translationa, Rotation, Scaling, Other Transformations:Reflection, Shear,Composite Transformations,Coordinate Transformation

Viewing Pipeline: Viewing reference frame, window, Viewport, Window-to-view port transformation, Multiple window transformation.

Clipping: Line clipping, Cohen-sutherland line clipping algorithm, Poligon clipping: Sutherland Hodgeman polygon clipping algorithm, Text clipping.

UNIT - III

3D Concepts: 3D Object Representation: Polygons, Curved Lines, Splines, Quadric Surfaces, **3D Transformations : Basic :**Translation, Coordinate-axis-Rotation, Arbitrary-axis Rotation, Scaling, Other: Reflection, Shear, Composition of 3D transformations, ,Projections : Parallel, Perspective, 3D Viewing, Visible-Surface Detection Algorithms: Back face removal, Z-Buffer, A-Buffer, Area-sub-division, Depth-Sorting(painter's),BSP-Tree,Octree,3D Clipping

UNIT - IV

Graphics Programming Color Models – RGB, YIQ, CMY, HSV – Animations – General Computer Animation, Raster, Keyframe Graphics programming using OPENGL – Basic graphics primitives –Drawing three dimensional objects - Drawing three dimensional scenes Rendering Introduction to Shading models – Flat and Smooth shading – Adding texture to faces –Adding shadows of objects – Building a camera in a program – Creating shaded objects– Rendering texture – Drawing Shadows

UNIT - V

Fractals Fractals and Self similarity – Peano curves – Creating image by iterated functions – Mandelbrot sets – Julia Sets – Random Fractals.

Overview of Ray Tracing Intersecting rays with other primitives – Adding Surface texture – Reflections andTransparency – Boolean operations on Objects.

Text Books:

1. Donald Hearn, Pauline Baker, Computer Graphics – C Version, second edition Pearson Education,2004.
2. F.S. Hill, Computer Graphics using OPENGL, Second edition, Pearson Education, 2003.

Reference Books:

1. James D. Foley, Andries Van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics- Principles and practice, Second Edition in C, Pearson Education, 2007.

Web Resource

1. https://nptel.ac.in/courses/106106090

Course Outcomes:

1. Know and be able to describe the general software architecture of programs that use 3D computer graphics.
2. Know and be able to discuss hardware system architecture for computer graphics.
3. Know and be able to select among models for lighting/shading: Color, ambient light.

III- Year I- Semester	Name of the Course	L	T	P	C
SAC3101	**.NET Programming (Skill Advanced Course-1)**	1	0	2	2

Course Objectives:

1. To introduce students with component based development
2. To familiarize IDE for .Net environment
3. To familiarize students with .Net framework
4. To introduce students with C#

Syllabus:

UNIT-I: Introduction to .NET Technology

Introduction: Introduction to .NET Framework, Visual Studio, Features of .NET, .NET Framework Architecture.

UNIT-II: Introduction to C#.NET

Introduction to C#.NET, OOPS in C#.NET, IDE OF Forms, Assemblies, and Namespaces, Streams, Multithreading.

UNIT– III: Introduction to ASP.NET and Programming

Introduction to ASP.NET and Programming, Web Form Fundamentals, Web Controls, State Management, Tracing, Session tracking, Fundamentals of ASP.net core.

UNIT-IV: Introduction to ADO.NET Fundamentals

ADO.NET Fundamentals, Data Binding-Single valued, Multi valued, The Data Controls-Form View, Grid View.

UNIT – V: Introduction to LINQ and Entity Framework.

LINQ and the Entity Framework, working with Services, Putting ASP.NET MVC in Context, Your First MVC Application.

Reference:

1. Andrew Stellman, Jennifer Greene, "Head First C#", 4th Edition, 2020, O'Reilly Media.

Course Outcomes:

CO1: S1tudent able to work with visual studio IDE

CO2: Student able to configure .Net environment

CO3: Student able to create forms and can handle different events

CO4: Student able to develop application programs

****

III- Year I- Semester	Name of the Course	L	T	P	C
Course Code: MC3101	Indian Constitution	2	0	0	0

Objective:

Main objective of this course is to familiarize students with Indian constitution,articles and amendments.

Syllabus:

UNIT-I

Introduction to Indian Constitution: Constitution' meaning of the term, Indian Constitution - Sources and constitutional history, Features - Citizenship, Preamble, Fundamental Rights and Duties, Directive Principles of State Policy.

LEARNING OUTCOMES:

After completion of this unit student will

- Understand the concept of Indian constitution
- Apply the knowledge on directive principle of state policy
- Analyse the History, features of Indian constitution
- Evaluate Preamble Fundamental Rights and Duties

UNIT-II

Union Government and its Administration Structure of the Indian Union: Federalism, Centre-State relationship, President: Role, power and position, PM and Council of ministers, Cabinet and Central Secretariat, Lok Sabha, Rajya Sabha, The Supreme Court and High Court: Powers and Functions;

LEARNING OUTCOMES: - After completion of this unit student will

- Understand the structure of Indian government
- Differentiate between the state and central government
- Explain the role of President and Prime Minister
- Know the Structure of supreme court and High court

UNIT-III

State Government and its Administration Governor - Role and Position - CM and Council of ministers, State Secretariat: Organisation, Structure and Functions

LEARNING OUTCOMES: - After completion of this unit student will

- Understand the structure of state government
- Analyse the role Governor and Chief Minister
- Explain the role of state Secretariat
- Differentiate between structure and functions of state secretariat

UNIT-IV

Local Administration - District's Administration Head - Role and Importance, Municipalities - Mayor and role of Elected Representative - CEO of Municipal Corporation Panchayats: Functions PRI: Zila Panchayat, Elected officials and their roles, CEO Zila Panchayat: Block level Organizational Hierarchy - (Different departments), Village level - Role of Elected and Appointed officials - Importance of grass root democracy

LEARNING OUTCOMES: - After completion of this unit student will

- Understand the local Administration
- Compare and contrast district administration role and importance
- Analyse the role of Myer and elected representatives of Municipalities
- Evaluate Zilla panchayat block level organisation

UNIT-V

Election Commission: Election Commission- Role of Chief Election Commissioner and Election Commissioner ate State Election Commission: Functions of Commissions for the welfare of SC/ST/OBC and women

LEARNING OUTCOMES: - After completion of this unit student will

- Know the role of Election Commission apply knowledge
- Contrast and compare the role of Chief Election commissioner and Commissioner ate
- Analyse role of state election commission
- Evaluate various commissions of viz SC/ST/OBC and women

REFERENCES:

1. Durga Das Basu, Introduction to the Constitution of India, Prentice – Hall of India Pvt. Ltd.. New Delhi
2. Subash Kashyap, Indian Constitution, National Book Trust
3. J.A. Siwach, Dynamics of Indian Government & Politics
4. D.C. Gupta, Indian Government and Politics
5. H.M.Sreevai, Constitutional Law of India, 4th edition in 3 volumes (Universal Law Publication)
6. J.C. Johari, Indian Government and Politics Hans
7. J. Raj Indian Government and Politics
8. M.V. Pylee, Indian Constitution Durga Das Basu, Human Rights in Constitutional Law, Prentice – Hall of India Pvt. Ltd.. New Delhi
9. Noorani, A.G., (South Asia Human Rights Documentation Centre), Challenges to Civil Right), Challenges to Civil Rights Guarantees in India, Oxford University Press 2012

E-RESOURCES:

1. nptel.ac.in/courses/109104074/8
2. nptel.ac.in/courses/109104045/
3. nptel.ac.in/courses/101104065/
4. www.hss.iitb.ac.in/en/lecture-details
5. www.iitb.ac.in/en/event/2nd-lecture-institute-lecture-series-indian-constitution

III- Year I- Semester	Name of the Course	L	T	P	C
PC3101L	Unix & Network Programming Lab	0	0	3	1.5

Course Objectives:

1. To familiarize students with the Linux/Unix environment.
2. To learn the fundamentals of shell scripting/programming.
3. To conceptualize Data Mining and the need for pre-processing.
4. To learn the algorithms used for various types of Data Mining Problem.

List of Shell Scripts:

1. Create a script that, given a user name, finds the home directory of the user using the /etc/passwd file.

Preparation:

- None

Script:

- **Script Name: findHomeDirectory.scr**
- **Arguments:** One, The user name.
- **Validation:** The minimum validation requirements are :
  i.Ensure that there is only one argument.
- **Body Section:** Create a script that, given the name of a user (as the only argument), prints the absolute pathname of the user's home directory

Testing the Script:

- Test the script with two or more arguments.
- Test the script with no arguments.
- Test the script with one argument.

Testing the Effect of the Script:

- Verify the script by using your user name.

2. Write a script that creates a file out of the /etc/passwd file.

Preparation:

- None

Script:

- **Script Name: newEtcPasswd.scr**
- **Arguments:** One, The name of the file.
- **Validation:** The minimum validation requirements are:
  i.Ensure that there is only one argument.
- **Body Section:** Create a script that makes a file out of the information in the /etc/passwd file using the following format.

```
User NameUser IdGroup IDHome Directory
- - - - - - - -   - - - - - - - - - - - - - - - - - - - -
ram23423/etc/usr/student/ram
- - -
```

Testing the Script:

- Test the script with two or more arguments.
- Test the script with no arguments.
- Test the script with one argument that is not the name of a file.
- Test the script with one argument that is the name of a file.

Testing the Effect of the Script:

- Verify the file was created and contains the correct information and format.

3. In a C Program, there is only one comment format. All comments must start with an open comment token, /*, and end with a close comment token, */. C++ programs use the C tokens for comments that span several lines. Single-line comments start with two slashes (//). In either case, the start token can be anywhere on the line.

Write a script to change every single-line comment in a C++ source file that uses C program start and end comment tokens to a single-line comment starting with a C++ single-line token. The comment itself is to be unchanged.

Preparation:

- Create at least five C++ source files in your home directory. The files do not have to be real C++ source files; they can contain only a few lines of comments, some with C program tokens and some with C++ single-line tokens. Each program should have at least one multiple comment and at least one single-line comment that uses the C program tokens. Use one or more blank lines between comments. The name of the files should have C++ extension (.c++), such as file1.c++.

Script:

- **Script Name: commentType.scr**
- **Arguments:** None
- **Validation:** The minimum validation requirements are:
  i. Ensure that there is no argument.
- **Body Section:** Create a script that finds all files with extension (.c++) under your directory and change only the lines with comments. The name of the files should be preserved. If a file has the name file1.c++, the name still should be file1.c++ after the change.

Testing the Script:

- Test the script with one or two arguments.
- Test the script with no arguments.

Testing the Effect of the Script:

- Check to see if the comments are changed in the files.

4. Write a script to backup and archive a list of files.

Preparation:

- Create a file and type in it the list of files (in your home directory) that you want to back and archive
- Create a directory in which you will store the backed-up files and archive file.

Script

- **Script Name:** backup.scr
- **Arguments:** A filename and a directory. The filename holds the list of the files that should be backed-up. The directory is where the backed-up files should be stored.
- **Validation:** The minimum validation requirements are:
  i. Ensure that exactly two arguments are entered.
  ii. Check that the first argument is the name of a file exists
  iii. Check that the second argument is the name of the directory that exists
- **Body Section:** Create backup files for files listed in the first argument. The backup files should have the same name as the original file with the extension bak. They should be copied to the directory given as the second argument.

Testing the Script:

- Test the script with no arguments
- Test the script with one argument
- Test the script with three arguments
- Test the script with two arguments in which the first one is not the name of the file
- Test the script with two arguments in which the second one is the name of a file rather than a directory.

- Test the script with name of the file and the name of the directory you created in the preparation section.

Testing the Effect of the Script:

- Check the contents of the directory to be sure that the files are copied

5. Write a script that finds all soft links to a specific file.

Preparation:

- Create a file and type some junk in it.
- Make at least five soft links to this file using completely arbitrary names.

Script:

- **Script Name: softLinkFinder.scr**
- **Arguments:** A filename. The file for which we want to find the soft links.
- **Validation:** The minimum validation requirements are:
  i. Ensure that exactly one argument is entered.
  ii. Check that only argument is the name of a file and that the specified file exists.
- **Body Section:** Use ls -l and grep command to find all the soft links attached to $1 positional parameter. Note that a file of type soft link is distinguished by lower case l. Be sure to find the soft links to the file defined in $1 and not other files.

Testing the Script:

- Test the script with no arguments
- Test the script with one argument
- Test the script with one argument that is not a file
- Test the script with one valid argument.

Testing the Effect of the Script:

- Check to make sure all the soft links you created are included in the list of soft links.

6. Create a script that simulates the ls -l command but prints only three columns of our choice.

Preparation:

- None

Script:

- **Script Name: ls.scr**
- **Arguments:** Three numeric arguments defining the column number of the ls -l output to be printed in the order we specify.
- **Validation:** The minimum validation requirements are :
  i. Ensure that exactly three arguments are entered.
  ii. Ensure that all three arguments are numeric
  iii. Ensure that each argument is less than or equal to the actual number of columns in the ls -l command output.

- **Body Section:** Creates a new command that shows the output of the ls -l command to be printed in three columns in the order we like.

Testing the Script:

- Test the script with no arguments.
- Test the script with one argument.
- Test the script with two arguments.

- Test the script with three arguments, one of them nonnumeric.
- Test the script with three arguments, two of them nonnumeric.
- Test the script with three arguments, one of them too large.
- Test the script with three arguments,1 4 5
- Test the script with three arguments,3 7 1

Testing the Effect of the Script:

- None

7. Create a script that sends contents of a message file to everybody who logged in.

Preparation:

- Create a file of a short friendly message and mention that this is a test message that should be discarded by the receiver

Script:

- **Script Name: message.scr**
- **Arguments:** One argument, a message file.
- **Validation:** The minimum validation requirements are:
  i. Ensure that exactly one argument is entered.
  ii. Ensure that the argument is a readable filename.
- **Body Section:** Create a script that uses awk to create a temporary file containing the usernames of those users who are logged into the system at this moment. Then send the message contained in the first argument to every logged-in user. Note that a user who has logged in more than once should receive only one message.

Testing the Script:

- Test the script with no arguments.
- Test the script with two arguments.
- Test the script with one argument that is not a readable file.
- Test the script with one valid argument.

Testing the Effect of the Script:

- You should include yourself in the recipient list. Check to see if you have received the message.

8. Create a script that can be executed only from a specific terminal. This is done for security purposes. For example, a superuser may write scripts that can only be executed from his or her office and nowhere else.

Preparation:

- None

Script:

- **Script Name: security.scr**
- **Arguments:** None.
- **Validation:** The minimum validation requirements are:
  i. Ensure that no argument is entered.
- **Body Section:** Create a script that prints a friendly message. However, the script can be executed only for one terminal. You can use the name of the terminal you are using when you write the script. If somebody uses the script from a terminal that is not authorized, the script is to exit immediately. Hint: Use the tty command to show your current terminal.

Testing the Script:

- Test the script with one argument.
- Test the script from right terminal.
- Log into the system using another terminal and test the script.

Testing the Effect of the Script:

- None

9. Create a script that finds each line in a file that contains a specified string.

Preparation:

- Create a file of at least 20 lines and insert a double quoted string, such as "hello," in several lines.

Script:

- **Script Name: search.scr**
- **Arguments:** Two arguments, the first is the string to be found; the second is the name of the file.
- **Validation:** The minimum validation requirements are:
  i. Ensure that exactly two arguments are entered.
  ii. Ensure that the second argument is the name of the file that exists and is not empty.
- **Body Section:** Create a script that uses grep and loops to find the line numbers in which the string is found. Note that grep should be applied to each line, not the whole file. The script should print the result in the following format:

Line Number: [Line contents]

Testing the Script:

- Test the script with no arguments.
- Test the script with one argument.
- Test the script with two argument but the second one is not a file.
- Test the script with two correct arguments.

Testing the Effect of the Script:

- Compare the results of your script with a printout of the file.

10. Create a script that compiles all C source files in your home directory and create executable files.

Preparation:

- Create at least five C source files in your home directory. The files do not have to be real C source files; at a minimum they should contain a comment line that contain a unique program name such as the following example:

/* .....file1.c.....*/

The name of the files should have a C source file extension (.c), such as file1.c.

Script:

- **Script Name: compile.scr**
- **Arguments:** Two arguments, the first is the string to be found; the second is the name of the file.
- **Validation:** The minimum validation requirements are :
  i. Ensure that there is no argument
- **Body Section:** Create a script that finds all files with extension (**.**c) under your home directory and compiles them one by one. Each executable file should have the same name as the source file except that the extension should be (**.**exe). For example, if the source filename is file1.c, the executable filename should be file1.exe. Use the following command to compile:

cc -o executable_filename source_filename

Testing the Script:

- Test the script with one or two arguments.

- Test the script with no arguments.

Testing the Effect of the Script:

- Verify that executable files were created under your home directory.

11. Create a script that finds all files in subdirectories that have the same filename.

Preparation:

- Make several directories, at different levels, under your home directory. For example, make ~/A, ~/B, ~/C, ~/A/AA, ~/A/BB, ~/A/AA/AAA, and so on until you have at least 15 directories. Copy a small junk file named file1 under some of these directories; do not change its name. Copy another small junk file named file2 under some other directories. Copy a third junk file under several directories. Be sure that some directories get a combination of file1 and file2 or file1 and file3. In at least three of the directories, create a junk file with a unique name.

Script:

- **Script Name: duplicateName.scr**
- **Arguments:** None
- **Validation:** The minimum validation requirements are :
  i. Ensure that there is no argument.
- **Body Section:** Create a script that uses find and awk commands to create a list of files that are duplicated; use the full pathname for the duplicated filenames. Hint: Use a basename command and an array in awk. The output should look like the following example:

  file1: ~/A/file1~/A/AA/file1~/A/B/BB/BBB/file1
  file2: ~/B/file2~/C/file2
  . . .

Testing the Script:

- Test the script with one argument.
- Test the script with no arguments.

Testing the Effect of the Script:

- Use a recursive long list command to list the complete contents of your home directory. Verify the output of your script against the list command output.

12. Create a script that search for multiple occurrences of the specified string in each line.

Preparation:

- Create a file of at least 20 lines and insert a double quoted string, such as "hello," in several lines.
- Include two or three occurrences of the string in some lines.

Script:

- **Script Name: search.scr**
- **Arguments:** Two arguments, the first is the string to be found; the second is the name of the file.
- **Validation:** The minimum validation requirements are :
  i. Ensure that exactly two arguments are entered.
  ii. Ensure that the second argument is the name of the file that exists and is not empty.
- **Body Section:** Create a script that uses grep and loops to find the line numbers in which the string is found. Note that grep should be applied to each line, not the whole file. The script should print the result in the following format:

Line Number: [Line contents]

Testing the Script:

- Test the script with no arguments.
- Test the script with one argument.
- Test the script with two argument but the second one is not a file.
- Test the script with two correct arguments.

Testing the Effect of the Script:

- Compare the results of your script with a printout of the file.

Text Books:

1. UNIX and Shell Programming, Behrouz A, Forouzan and Richard F.Gilberg, Cengage Learning, 2003.
2. Advanced Programming in UNIX Environment, W.Richard Stevens, Stephen A Rago, 3rd Edition, Addison-Wesley Professional, 2013.

Reference Books:

1. UNIX and shell programming by B.M. Harwani, OXFORD university press.
2. Unix essentials by Sumitabha Das
3. Unix Shell Programming, Stephen G.Kochan, Patrick Wood, 3/e, Pearson

Course Outcomes:

1. To use Unix utilities and perform basic shell control of the utilities
2. To use the Unix file system and file access control.
3. To use of an operating system to develop software
4. Students will be able to use Linux environment efficiently
5. Solve problems using bash for shell scripting

III- Year I- Semester	Name of the Course	L	T	P	C
PC3102L	Machine Learning Lab	0	0	3	1.5

Course objectives: This course will enable students to

1. Make use of Data sets in implementing the machine learning algorithms
2. Implement the machine learning concepts and algorithms in any suitable language of choice.

Exercise 1: Introduction to R, Basic Commands, Graphics, Indexing Data, Loading Data, Additional Graphical and Numerical Summaries.

Exercise 2:

Using simple linear regression perform the following tasks on the Autodata set.

(a) Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor.

(b) Use the summary() function to print the results. Comment on the output.

That is

i. Is there a relationship between the predictor and the response?
ii. How strong is the relationship between the predictor and the response?
iii. Is the relationship between the predictor and the response positive or negative?
iv. What is the predicted mpg associated with a horsepower of98? What are the associated 95% confidence and prediction intervals?

(c) Plot the response and the predictor. Use the abline() function to display the least squares regression line.

(d) Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

Exercise 3:

Using multiple linear regression perform the following tasks on the Autodata set.

(a) Produce a scatter plot matrix which includes all of the variable sin the data set.

(b) Compute the matrix of correlations between the variables using the function cor() . You will need to exclude the name variable, cor() which is qualitative.

(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results.

Comment on the output, That is

i. Is there a relationship between the predictors and the response?
ii. Which predictors appear to have a statistically significant relationship to the response?
iii. What does the coefficient for the year variable suggest?

(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit.

Do the residual plots suggest any unusually large outliers?

Does the leverage plot identify any observations with unusually high leverage?

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

(f) Try a few different transformations of the variables, such aslog(X),$\sqrt{}$ X, X 2 . Comment on your findings.

Exercise 4:

Implementation of KNN on the Breast Cancer Data set.

Exercise 5:

Implement LDA, QDA, and NAÏVE BAYES on the Stock market data and produce the empirical comparison.

Exercise 6:

Analyse the CAR SEATS dataset using Decision Trees.

Exercise 7:

Application of SVM for Gene Expression Data.

Exercise 8:

Perform PCA on the USArrests data set.

Exercise 9:

Perform K-Means Clustering on NC160 Dataset.

Exercise 10:

Perform Hierarchical clustering on NC160 Dataset.

Text Books:

1. Gareth James, et al. An Introduction to Statistical Learning: with Applications in R,Springer. 2nd edition (2021 edition).

Reference Books:

1. Tom Mitchell, "Machine Learning", McGraw Hill, 1997
2. E. Alpaydin, "Introduction to Machine Learning", PHI, 2005.
3. Andrew Ng, Machine learning yearning, https://www.deeplearning.ai/machine-learning-yearning/
4. Hands-on machine learning with R" by Bradley Boehmke & Brandon Greenwell
5. "Machine learning with R, the tidyverse, and mlr" by Hefin I. Rhys

Web Resources:

1. R Programming Crash Course – https://www.youtube.com/watch?v=ZYdXI1GteDE&t=1849s
2. Machine Learning With R | Machine Learning Full Course | Machine Learning Tutorial | Simplilearn - https://www.youtube.com/watch?v=6dEUTmoXz0w
3. Statistics for Data Science | Probability and Statistics | Statistics Tutorial | Ph.D. (Stanford) - https://www.youtube.com/watch?v=Vfo5le26IhY&list=PLlgLmuG_KgbaXMKcISC-fdz7HUn1oKr9i
4. Linear Regression Algorithm | Linear Regression Machine Learning | Linear Regression Full Course - https://www.youtube.com/watch?v=tFi4Y_y-GNM
5. Learning: Support Vector Machines - https://www.youtube.com/watch?v=_PwhiWxHK8o

Course Outcomes:

1. Familiarize with machine learning algorithms.
2. Able to implement machine learning algorithms using R programming.
3. Analyze and visualize data using R programming language.