PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR

Name of the Faculty: Science and Technology

CHOICE BASED CREDIT SYSTEM

Syllabus: Computer Science

Name of the Course: B.Sc. (Entire Computer Science) - III (Semester –V and VI)

(Syllabus to be implemented w.e.f. June 2024)

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology
Choice Based Credit System
(CBCS) (w.e.f.2024-25)Revised
Structure for B.Sc. (ECS)-III

Subject/ Core Course	Name and Type of the Paper		No. of Papers/ Practical	Hrs./ Week			Total Mark sper Paper	UA	CA	Credits
	Type	Name		L	T	P				
Class:	**B.Sc. (Entire Computer Science)- III Semester – V**									
Ability Enhancemen tCourse	(AECC)	English (Business English)	Paper II Part A	4	--	--	50	40	10	2.0
Core	%DSC 1E	Data Communicationand Networking	Paper IX	4	--	--	100	80	20	4.0
	DSC 1F	Advanced Java	Paper X	4	--	--	100	80	20	4.0
	DSC 1G	Dot NET Core	Paper XI	4	--	--	100	80	20	4.0
	DSE 1	A-Advanced Python Programming **or** B-Theory of Computatio **or** C-Mobile Application and Development	Paper X	4	--	--	100	80	20	4.0
Skill Enhancemen tCourse	$ SEC 2	Artificial Intelligence	Paper XI	4	--	--	100	80	20	4.0
Total Theory Semester-V				**24**			**550**	**440**	**110**	**22**
Class:	**B.Sc. (Entire Computer Science)- III Semester – VI**									
Ability Enhancemen tCourse	(AECC)	English (Business English)	Paper II Part B	4	--	--	50	40	10	2.0
Core	%DSC 2E	Network Security	Paper XII	4	--	--	100	80	20	4.0
	DSC 2F	Data Warehousing and Data Mining	Paper XIII	4	--	--	100	80	20	4.0
	DSC 2G	ASP.Net Core MVC	Paper XIV	4	--	--	100	80	20	4.0

	DSE 2	A- React JS **or** B- Compiler Construction **or** C- Internet of things	Paper XV	4	--	--	100	80	20	4.0
Total Theory Semester-VI				**20**			**450**	**360**	**90**	**18**
Practical's on		Project		--	--	5	100	80	20	4.0
		DSC 1F and 2F		--	---	5	100	80	20	4.0
		DSC 1G and 2G		--	--	5	100	80	20	4.0
		DSE 1A/B/C and 2A/B/C		--	--	5	100	80	20	4.0
Total (practical's)						**20**	**400**	**320**	**80**	**16**
Grand Total				**42**	**--**	**20**	**1400**	**1120**	**280**	**56**

% Theory paper: (Core Computer Science Course)
$The students can choose MOOCs/ NPTEL/SWAYAM/Pathshala/Add-on / Skill based courses of university/college initiated courses of same credits.
$ These courses are not compulsory, but after completion of these courses students get additional credits on their
Marklist.
$SEC Courses initiated by colleges

Abbreviations:

L: Lectures

T: Tutorials

P: Practicals

UA : University Assessment

CA : College Assessment

CC: Core Course

AEC : Ability Enhancement Course

DSE : Discipline Specific Elective Paper

SEC : Skill Enhancement Course

Type: DSC 1E

Course Title: Data Communication and Networking

(Paper Code: Paper IX)

Course Objectives:

1. To understand the structure of Data Communications System and its components.
2. Be familiarize with different network terminologies.

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

1. Familiarize with contemporary issues in network technologies.
2. Know the layered model approach explained in OSI and TCP/IP network models
3. Identify different types of network devices and their functions within a network.
4. Know the Basic routing mechanisms, IP addressing scheme and internet working concepts.
5. Familiarize with IP and TCP Internet protocols.
6. Understand major concepts involved in design of WAN, LAN and wireless networks.
7. Know the basics of network configuration and maintenance.
8. Know the fundamentals of network security issues.

Unit 1: Introduction to Computer Networks [20]

Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of ISO-OSI Reference Model, Overview of TCP/IP Protocol Suite.

Data Communication Fundamentals and Techniques:

Signals-Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation, Multiplexing Techniques- FDM, TDM, Transmission Media, Switching: Circuit Switching, Message Switching, Packet Switching

Unit 2: Data Link Layer Functions and Protocols [10]

Design issues, Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

Unit 3: Multiple Access Protocol and Network Layer **[10]**

Design issues, CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back -Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols, Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

Unit 4: Transport, Session, Presentation and Application Layer Protocol **[20]**

Transport Services- Error and Flow Control, Connection Establishment and Connection Release, Flow Control & Buffering, TCP/IP protocol suite, Concept of- TCP, UDP, IP, FTP, DNS, Telnet, SMTP, POP, HTTP, WWW, ARP, RARP.

Reference Books:

- B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
- S. Tanenbaum: Computer Networks, Fourth edition, PHI Pvt. Ltd 2002

Type: DSC 1F

Course Title: Advanced Java

(Paper Code: Paper IX)

Course Objectives:

1. To learn GUI programming using swing Technology
2. To study database programming using Java.
3. To study web development concept using Servlet and JSP
4. To learn socket programming concept using Networking.

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

1. Design, create, build, and debug Java applications.
2. Explore Integrated Development Environment (IDE).

Unit 1: Networking and Database **[15]**

Networking: Basics, networking classes and interfaces, using java.net package

Working with databases: Steps for Connecting to databases, Types of Drivers, Handling Exceptions, Creating and Using Statement Objects, Using Statements to Insert, Update, Delete Data into a Database, Using the ResultSet Class, Data navigation, Prepared Statements, Callable Statements.

Unit 2: Introducing Servlet **[18]**

Advantages of Servlet over CGI, Features of Servlet, Introducing Servlet API, Javax.servlet package, Javax.servlet.http package, Introducing Servlet, Advantages of Servlet over CGI, Features of Servlet, Servlet life Cycle,Init(),Service(),Destroy(),Working with GenericServlet and HttpServlet, RequestDispatcher interface, Include() and forward() ,Use of RequestDispatcher, Session in Servlet, Introducing session, Session tracking mechanism, Cookies, Advantages and disadvantages,use of cookies, Hidden form filed, Advantages and disadvantages, use of Hidden form filed, URL rewritten, disadvantages, use of URL rewritten, HttpSession, Advantages and disadvantages, use of URL HttpSession

Unit 3: Java Server Pages **[12]**

Introduction, Advantages of JSP over Servlet, JSP architecture , JSP life cycle, Implicit objects in JSP- request, response, out, page, page Context, application, session, config, exception, JSP tag elements- Declarative, Declaration, scriplet, expression, action. Java Bean- Advantages and Disadvantages, Use Bean tag- setProperty and getProperty, Bean In Jsp, JSTL core tag: General purpose tag, conditional tag, networking tag, JSTL SQL tags, JSTL formatting tags, JSTL XML tags, Custom tag: empty tag, body content tag, iteration tag, simple tag , Introducing internationalization and Java: local class, ResourceBundle class, Handling different Error and Exceptions.

Unit 4: Hibernate and Spring [15]

Introduction hibernate,architecture of HB ,HB application,HB application using annotation,introduction to inheritance mapping,introduction to collection mapping.

Spring: Introduction to spring,spring module,application of spring,dependency injection-constructor injection and setter injection,simple spring jdbc template,HB with spring.

Reference books:

- Java the complete Reference by Herbert Schildt
- Java Servlet Programming by Jasan Hunter
- Java Server Programming by Bayross and Shah
- JDBC, Servlet and JSP Black Book- Santosh Kumar K.
- Hibernate- Sharanam Shah and Vaishali Shah
- Spring Persistence with Hibernate- Paul Tepper Fisher, Brian D Murphy

Type: DSC 1G

Course Title: Dot NET Core

(Paper Code: Paper XI)

Course Objectives:

1. To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling and functions in C#.
2. To understand the importance of Classes and objects along with constructors, Arrays and Vectors in C#.
3. Discuss the principles of inheritance, interface and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages in C#.
4. To understand importance of Multi-threading and different exception handling mechanisms in C#.
5. To understand basic idea about how to design GUI base windows application using C#.

Course Outcomes: Students will be able to:

1. Gain proficiency in designing, implementing, testing, debugging, and documenting programs utilizing fundamental data types, computation, basic I/O, conditional and control structures, string manipulation, and functions within the C# programming language.
2. Appreciate the significance of classes, objects, constructors, arrays, and vectors within the C# framework.
3. Explore the principles of inheritance and interfaces, demonstrating their application through problem analysis assignments and their relevance to method design, abstract classes, interfaces, and packages in C#.
4. Recognize the importance of multi-threading and various exception handling mechanisms in C# programming.
5. Acquire a foundational understanding of designing GUI-based Windows applications using C#.

Unit 1: Introduction to C# **[20]**

Understanding .NET- The .NET Framework, .NET Core, Download and install C# Development Environments - Visual Studio, Visual Studio Code, building console apps using Visual Studio 2022 and Building console apps using Visual Studio Code, C# Basics- Variables and Data Types, Reference and Value Types-Nullable types, Elvis operator, Null coalescing operator, Boxing and unboxing, Keywords, Initialization, Type Inference, Console Input and Output., Operators, Operator precedence, Type conversion, C# statements-Branching, Jumping, Looping, Complex data types- Enums, Arrays, Tuples.

Unit 2: Object Oriented Programming **[20]**

Classes and object-Declaration, Access modifiers, Data, Methods, Method parameters, Constructors, Deconstruct, Method overloading, Properties, Local and global variable and methods, Static classes, methods and members, nested classes, Indexers, Partial types and methods, Structs and Records, Inheritance- Base and derived classes, advantages, Types. Constructors in inheritance. Abstract classes, sealed class, Interfaces - Defining and implementing, Default interface methods, Interface inheritance, .NET interfaces, Polymorphism- Virtual methods. Method overriding, operator overloading, Abstract methods, Sealed types.

Unit 3: Threading, Exception and Resource Management **[12]**

Exception- about exception, Exceptions Hierarchy, Throwing and Catching Exception, The try-finally Construct, IDisposable and the "using" Statement, Advantages of Using Exceptions, inbuild exception, custom exception, Threading-about threading, Thread Name, Thread Priority, and Thread State, Foreground and background threads in C#, Multithreading - An Overview, The Thread Class, ThreadPool Threads, Collections- Generic collections, Concurrent collections, Specialized collections, Performance considerations, Resource Management- Finalizers, Garbage Collection, IDisposable, The using statement, Serialization-Attributes, JSON serialization, Binary serialization, XML serialization

Unit 4:Delegate, Event and LINQ **[8]**

Delegates-Multicast delegates, generic delegates, Action<T>, Predicate<T>, Func<T> , Lambdas-Expression and statement lambdas, Parameters, Return type, Captures, Events-Defining, Raising, Standard and custom events, LINQ- Enabling features, LINQ expression, LINQ pattern, Joins, Aggregations, Basic of Windows application.

Reference Books:

- Nagel, Karli Watson, Jay Glynn, Morgan Skinner, Bill Evjen.
- Inside C# - Microsoft Press by Tom Archer, Andrew Whitechapel.
- Programming Microsoft Visual C# 2005 - The Language (Microsoft Press) by Donis Marshall

Type: DSE2-A

Course Title: Advanced Python Programming

(Paper Code: Paper XII)

Course Objectives:

1. To learn Multithreaded Programming.
2. To learn GUI programming using different types of python modules.
3. To study database programming using MySQL.
4. To study Web server programming using CGI and XML.
5. To learn socket programming concept using Networking.

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

1. Design, Create, Build, and Debug Python applications.
2. Explore Integrated Development Environment (IDE).

Unit 1: Threading and Networking **[20]**

Understanding threads, Class and threads, Difference between Process and a Thread, Creating Threads, Thread Synchronization, Treads Life cycle, Multi-threading, Deadlock of Threads, Avoiding Deadlock in a Program, use cases.

Networking in Python: Introduction to Sockets Programming, Server Socket Methods, Client Socket Methods, IP Address, URL, TCP/IP Server, TCP/IP Client, Sending E-mail Application.

Unit 2: GUI Programming and **[20]**

GUI in Python, Advantages of GUI, Introduction to GUI library, Basic Operations using Tkinter, Root Window, Working with Containers: Frame, Canvas, Layout Management, Events and Bindings, Font, Colors, drawing on Canvas (line, oval, rectangle, etc.), Widgets: Label, Button, Checkbutton, Entry, Listbox, Message, Radiobutton, Text, Spinbox, Scrollbar, Menu etc., different dialog boxes and message boxes, Writing Python Programs for GUI applications

Unit 3: Database Connectivity [5]

Database Connectivity using MySQL: Installation of MySQL, Installing MySQL connector, Steps for Database Connectivity, Inserting, Retrieving, Deleting and updating the data working with Stored Procedure.

Unit 4: Introduction to Django [15]

Django Project and App life cycle,Creating Project and App, Difference between app and project, The Project Structure, Setting Up Your Project,

Django Templates and Static, Django Models, Django Forms, Django Views, Django Sessions and Cookies, Django Serialization, Deserialization and Mixins, Django Authentication and Authorization, Django Middleware, Send Email in Django and CSV,PDF files and GIT and Github, Django Rest Framework Introduction, Django Rest Framework Serializer and Deserializer, Django Rest Framework Views, Django Rest Framework Authentication And Authorization, Django Rest Framework Pagination and Routing

Reference Books:

- Advanced Python Programming-By Richard Ozer, 2017
- Core python Programming- Dr. R NageswaraRao
- Expert Python Programming,: Become a master in Python-By Michał Jaworski, TarekZiade
- MySQL for Python: Database Access MadeEasy- A.Lukaszewski

l

Type: DSE2-B

Course Title: Theory of Computer Science

(Paper Code: Paper XII)

Course Objectives

1. Course should provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (and less magical) view towards algorithmic design and in general computation itself.
2. The course should in addition clarify the practical view towards the applications of these ideas in the engineering part of CS.

Course Outcomes: After learning this course, the students should be able to-

1. Understand the basic concepts and application in Theory of Computation.
2. Apply this basic knowledge of Theory of Computation in the computer field to solve computational problems and in the field of compiler also.

Unit 1: Preliminaries and Finite Automata [18]

Basic Definitions, Sets, Various ways of describing a Set, Subsets, operations on Sets, Infinite Sets Relations, Properties of relations, Equivalence of relations.

Finite Automata: Introduction, Deterministic Finite Automata, Non Deterministic Finite Automata, The Equivalence of DFAs and NFAs, Finite Automata with ε- Moves, Equivalence of NFA with ε- Transitions and NFA without Transitions, Finite Automata with output, Moore Machine, Melay Machine Equivalence of Moore and Melay Machine.

Unit 2: Regular Expression and Properties of Regular Sets [12]

Regular Expression Operations on set of strings, Regular Expression, Regular Sets, Equivalence of finite automata and regular expression Properties of Regular Sets Closure properties, The pumping lemma of regular sets, Application of pumping lemma.

Unit 3: Regular and Context Free Grammars [12]

Context Free Grammars (CFG) Derivation and Language generated by grammar, Derivation Trees, Ambiguity of CFG, Simplification of CFG, Normal forms of CFG Regular Grammars Equivalence of regular grammars and finite automata Closure properties of CFG.

Unit 4: Pushdown Automata and Turing Machine [18]

Introduction, Definitions, Equivalence of acceptance by final state and empty stack, Definition of DPDA and NPDA their correlation and examples of NPDA, CFG to PDA: Method and example, Closure properties of Regular language, Application of PDA.

Introduction of Turing Machine:

Turing Machine model and definition of TM, Language accepted by TM, Design of TM and examples.

Reference Books:

- J.P. Hopcroft, Rajeev Motwani, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, II Edition, Pearson Education, 2001.
- John Martin, Introduction to Languages and Theory of Computation, Tata McGrawHill, 2003.
- Daniel I.A., Cohen, Introduction to Computer Theory, 2 nd Edition, John Wiley and Sons, Inc, 2000.

Type: DSE2-C

Course Title: Mobile Application Development

(Paper Code: Paper XII)

Course Objectives:

1. To understand Android platform and its architecture.
2. To learn about mobile devices types and different modern mobile operating systems.
3. To learn activity creation and Android User Interface designing.
4. To learn basics of Intent, Broadcast and Internet services.
5. To learn about different wireless mobile data transmission standards.
6. To understand and learn how to integrate basic phone features, multimedia, camera and Location based services in Android Application.
7. To learn about different systems for mobile application development, deployment and distribution in Mobile market place (Android, IOS).
8. To understand and carry out functional test strategies for mobile applications.

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

1. Create, Design, Debug and Deploy Android applications.
2. Explore Integrated Development Environment (IDE).

Unit 1: Introduction **[15]**

What is Android, Android Versions and its Feature Set, Various Android Devices on the Market, Android Market Application Store, Android Development Environment, System Requirements, Android SDK, Installing Java, and ADT bundle - Eclipse Integrated Development Environment (IDE), Creating Android Virtual Devices (AVDs).

Android Architecture Overview and Application:

Android Software Stack, The Linux Kernel, Android Runtime - Dalvik Virtual Machine, Android Runtime – Core Libraries, Dalvik VM Specific Libraries, Java Interoperability Libraries, Android Libraries, Application Framework, Creating a New Android Project, Defining the Project Name and SDK Settings, Project Configuration Settings, Configuring the Launcher Icon, Creating an Activity, Running the Application in the AVD, Stopping a Running Application, Modifying the Example Application, Reviewing the Layout and Resource Files.

Unit 2: Android Software Development Platform and Framework [15]

Understanding Java SE and the Dalvik Virtual Machine, The Directory Structure of an Android Project, Common Default Resources Folders, The Values Folder, Leveraging Android XML, Screen Sizes , Launching Mobile Application: The AndroidManifest.xml File, Android Application Components, Android Activities: Defining the UI, Android Service s: Processing in the Background, Broadcast Receivers: Announcements and Notifications Content Providers: Data Management, Android Intent Objects: Messaging for Components, Android Manifest XML: Declaring Your Components.

Unit 3: Understanding Android User Interfaces, Views and Layouts [15]

Designing for Different Android Devices, Views and View Groups, Android Layout Managers, The View Hierarchy, Designing an Android User Interface using the Graphical Layout Tool, Displaying Text with Text View, Retrieving Data from Users, Using Buttons, Check Boxes and Radio Groups, Getting Dates and Times from Users, Using Indicators to Display Data to Users, Adjusting Progress with Seek Bar, Working with Menus using views, Gallery, Image Switcher, Grid View, and Image View views to display images, Creating Animation.

Unit 4: Databases, Intents, Location-based Services and Development [15]

Saving and Loading Files, SQLite Databases, Android Database Design, Exposing Access to a Data Source through a Content Provider, Content Provider Registration, Native Content Providers Intents and Intent Filters: Intent Overview, Implicit Intents, Creating the Implicit Intent Example Project, Explicit Intents, Creating the Explicit Intent Example Application, Intents with Activities, Intents with Broadcast Receivers.

Application Development:

Sending SMS Messages Programmatically, Getting Feedback after Sending the Message, Sending SMS Messages Using Intent Receiving, Sending email, Introduction to location-based service, Configuring the Android Emulator for Location -Based Services, Geocoding and Map-Based Activities Multimedia: Audio, Video, Camera: Playing Audio and Video, Recording Audio and Video, Using the Camera to Take and Process Pictures.

Reference Books

- Android Programming Unleashed (1st Edition) by Harwani.
- Beginning Mobile Application Development in the Cloud (2011), Richard Rodger.
- Learn Android App Development by Wallace Jackson.
- Professional Android Application Development by Reto Meier.

Type: SEC2

Course Title: Artificial Intelligence

(Paper Code: Paper XIII)

Course Objectives:

1. To impart knowledge about Artificial Intelligence.
2. To give understanding of the main abstractions and reasoning for intelligent systems.
3. To enable the students to understand the basic principles of Artificial Intelligence in various applications.

Course Outcomes:

1. To develop semantic-based and context-aware systems to acquire, organize process, share and use the knowledge embedded in multimedia content.
2. To maximize automation of the complete knowledge lifecycle and achieve semantic interoperability between Web resources and services.
3. To motivate the students to take up higher studies in Computer Science and Artificial Intelligence.

Unit 1: Introduction and Searching **[15]**

AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

Searching: Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A*, AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

Unit 2: Knowledge Representation **[15]**

Issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Baye’s probabilistic interferences and dempstershafer theory.

Unit 3: Order of Logic [15]

Inference in first order logic, propositional vs. first order inference, unification and lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods ,Reinforcement Learning.

Unit 4: Expert Systems [15]

Introduction, basic concepts, structure of expert systems, the human element in expert systems how expert systems works, problem areas addressed by expert systems, expert systems success factors, types of expert systems, expert systems and the internet interacts web, knowledge engineering, scope of knowledge, difficulties, in knowledge acquisition methods of knowledge acquisition, machine learning, intelligent agents, selecting an appropriate knowledge acquisition method, societal impacts reasoning in artificial intelligence, inference with rules, with frames: model based reasoning, case based reasoning, explanation and meta knowledge inference with uncertainty representing uncertainty.

Reference Books

- S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, SecondEdition, Pearson Education
- David Poole, Alan Mackworth, Randy Goebel, ”Computational Intelligence : a logical approach”, Oxford University Press.
- G. Luger, “Artificial Intelligence: Structures and Strategies for complex problem solving”, Fourth Edition, Pearson Education.
- J. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers.

Type: DSC 2E

Course Title: System Security

(Paper Code: Paper XIV)

Course Objectives:

1. To learn cryptographic tools.
2. To learn security issues regarding user Authentication.
3. To understand the various access control mechanisms.
4. To learn various types of malicious softwares and Denial-of-Service attacks.

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

1. Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.
2. Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.
3. Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
4. Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.

Unit 1: Fundamental of Security **[20]**

Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks – Active and Passive, Security services, Security Mechanisms, A model for Network Security , Access Control Mechanisms: Access Matrix, ACL and capabilities, Access Control Models.

Unit 2: Cryptography Concepts and Techniques **[20]**

Introduction to Cryptography, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks, Symmetric Key Cryptographic Algorithms: Algorithm Types and Modes, An overview of Symmetric Key Cryptography, DES, International Data Encryption Algorithm (IDEA), RC5, Blowfish, AES, Asymmetric Key Cryptography: Brief History of Asymmetric Key Cryptography, An overview of Asymmetric Key Cryptography, The RSA Algorithm, Symmetric and Asymmetric Key Cryptography Together.

Unit 3: Digital Signatures and Internet Security Protocols [12]

Digital Signatures: Introduction, Message digests, MD5, SHA-512,MAC, HMAC, Knapsack Algorithm, Elliptic curve Technology, ELGamal Algorithm, Internet Security Protocols: Secure Socket Layer/TLS, Secure Electronic Transaction, SSL versus SET, E-mail Security- PGP, S/MIME.

Unit 4: User Authentication and Network Security [8]

Introduction User Authentication, Authentication basics, Passwords, use of smart cards, Biometrics, Kerberos, Introduction Network Security, Firewalls, types of firewalls, IP Security, Intrusion: Intruders, Audit Records, Intrusion Detection, honey pots.

Reference Books:

- Cryptography and Network Security by Atul Kahate, Tata McGraw-Hill
- Cryptography and Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Special Indian Edition, Tata McGraw-Hill.
- Network Security Essentials: Applications and Standards by William Stallings, Pearson Education.
- Fundamentals of Computer Security Technology: Edward Amoroso, Prentice-Hall.
- Cryptography and Data Security: Dorothy E. Denning, Addison-Wesley.
- Cryptography -Theory and Practice: Douglas R. Stinson, CRC Press.

Type: DSC 2F

Course Title: Data Warehouse and Data Mining

(Paper Code: Paper XV)

Course Objectives:

1. To understand the principles of Data warehousing and Data Mining.
2. To understand the Architecture of a Data Mining system.
3. To perform classification, association, and prediction of data.

Course Outcomes: The students should be able to:

1. Identify data mining problems and implement the data warehouse.
2. Write association rules for a given data pattern.
3. Choose classification and clustering solutions.

Unit 1: Introduction to Data Warehouse and Data Mining [15]

Differences between Operational Database Systems and Data Warehouses, Data Warehouse Architecture, Data Warehouse Components, A Multidimensional Data Model, Schemas, Data Warehouse Implementation, Data cube Technology, OLAP operations, Data mining query language, Data Mining:- What is data mining, Evolution, KDD, What kind of data, Architecture, data mining views, Data Mining Functionalities, Issues in Data Mining.

Unit 2: Data Preprocessing and Association Rule mining [15]

Data Preprocessing:An Overview, Extract, Transform, Load (ETL) Processes, Data Cleaning, Data Integration, Data Transformation and Data Discretization, Data Reduction, Frequent Patterns,

Associations, and Correlations: Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules, Frequent Itemset Mining Methods-Apriori Algorithm: Finding Frequent Itemsets, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A Pattern-Growth Approach for Mining Frequent Itemsets, Mining Multilevel and multidimensional Association Rules, Constraint-Based Frequent Pattern Mining

Unit 3: Supervised Learning Technique [15]

supervised and unsupervised learning, What Is Classification? What is regression, difference between classification and regressing, General Approach to Classification, Issues regarding Classification and Predication, Binary and Multiclass Classification, Types of classifications, Classification by Decision tree induction, Bayesian Classification, Classification by Back propagation, Logistic regression, k-Nearest-Neighbor Classifiers, SVM, Introducing Ensemble Methods-Bagging, Boosting, AdaBoost, Random Forests, Other classification methods, Prediction: regression. Model Evaluation and Selection-Metrics for Evaluating Classifier Performance, Cross-Validation, underfitting and overfittin.

Unit 4: Unsupervised Learning Technique and Applications [15]

Clustering: What is Cluster Analysis, Types of data in Cluster Analysis, A Categorization of Major Clustering Methods., Partitioning Methods, Hierarchical Methods, Density-Based Methods, Model-Based Clustering Methods: Statistical Approach, Neural Network Approach, Outlier Analysis, Applications and Trends in Data Mining: Data Mining Applications, Data Mining for Financial Data Analysis, Data Mining for Retail and Telecommunication Industries, Data Mining in Science and Engineering, Data Mining for Intrusion Detection and Prevention, Data Mining and Recommender Systems, Spatial Data Mining. Text Data Mining, Multimedia Data Mining, Web Data Mining, Privacy, Security, and Social Impacts of Data Mining, Data Mining and Intelligent Query Answering, Trends in Data Mining.

Reference Books:

- Data Mining – Concepts and Techniques – Jiawei Han, Michelinen Kamber, Morgan Kaufmann Publishers, Elsevier, 2nd Edition, 2006.
- Introduction to Data Mining, Pang – Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.
- Data Warehouse Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.

Type: DSC 2G

Course Title: ASP.Net Core MVC

(Paper Code: Paper XVI)

Course Objectives:

1. Demonstrate the creation of ASP.NET Core MVC Web Applications using .NET 8, covering project file structure, main method, hosting options, and configuration files.
2. Cover Models, Controllers, Views, and Dependency Injection in ASP.NET Core MVC applications.
3. Introduce Entity Framework Core and guide the installation process also explain DbContext in Entity Framework Core and database connection string configuration and database operations.
4. Discuss Transactions, Migration, and Database Seeding in Entity Framework Core.
5. Introduce Partial Views, View Components, and Razor View Engine.
6. Cover Action Results, Routing, Model Binding, HTML Helpers and Tag Helpers in ASP.NET Core MVC.

Course Outcomes: Students will be able to:

1. Demonstrate the creation of ASP.NET Core MVC Web Applications using .NET 8.
2. Understand project file structure and implement Models, Controllers, Views, and Dependency Injection in ASP.NET Core MVC applications.
3. Utilize Entity Framework Core for Data Access and perform database operations using Entity Framework Core.
4. Discuss transactions, migration, and database seeding in Entity Framework Core.
5. Implement Reusability in Views using Partial Views, View Components, and the Razor View Engine for efficient view management.
6. Cover Action Results, Routing, Model Binding, HTML Helpers, and Tag Helpers in ASP.NET Core MVC applications.
7. Data Annotations and Model Validations, including custom validations and remote validation.
8. Discuss different methods of State Management, including Cookies and Sessions.

Unit 1: Introduction to ASP.Net Core MVC [20]

Overview of Microsoft Web Technologies, Introduction to ASP.NET Core Framework. NET Core Environment Setup, Install .NET Core SDK, SQL Server 2022, SSMS, Creating ASP.NET Core Web Application using .NET 8, NET Core Project File Structure, NET Core Main

Method, NET Core In Process Hosting, out of Process Hosting, Launch Settings.json File, AppSettings .json file, Middleware Components, Web Root (wwwroot) Folder, Static Files Middleware, Configuring Default Page, Developer Exception Page Middleware Command Line Interface, Project Templates in ASP.NET Core Application, Introduction to ASP.NET Core MVC Framework, Set up MVC in ASP.NET Core, Models, Controllers and Views in ASP.NET Core MVC, ASP.NET Core Dependency Injection, Creating ASP.NET Core Application using MVC.

Unit 2: Entity Framework Core **[20]**

Introduction to Entity Framework Core, How to Install Entity Framework Core, DbContext in Entity Framework Core, Database Connection String in Entity Framework Core, CRUD Operations in Entity Framework Core, Entity States in Entity Framework Core, Data Annotation Attributes in Entity Framework Core- Table Attributes, Column Attributes, Key Attribute, ForeignKey Attribute, Index Attribute, InverseProperty Attribute, NotMapped Attribute, Required Attribute, MaxLength and MinLength Attribute, Database Generated Attribute, TimeStamp Attribute, ConcurrencyCheck Attribute, Relationships in Entity Framework Core- One-to-One Relationships, One-to-Many Relationships, Many-to-Many Relationships, Self-Referencing Relationship, Asynchronous Programming with Entity Framework Core, Disconnected Entities in Entity Framework Core, Stored Procedures in Entity Framework Core, Transactions in Entity Framework Core, Migration in Entity Framework Core, Database Seedd in Entity Framework Core, Entity Framework Core Database First Approach

Unit 3: Model, View, Controller and Routing **[10]**

ViewData, ViewBag, Strongly Typed View, ViewModel, TempData, Post-Redirect-Get (PRG) Pattern Example, Layout View, Sections in Layout View, ViewStart, ViewImports, Partial Views, Different Ways to Render Partial View, View Components, Razor View Engine and Razor Syntax, How to Install and use Bootstrap in ASP.NET Core MVC, Action Results in ASP.NET Core MVC- Action Results, View Result, Partial View Result, JSON Result, Content Result, File Result, Redirect Results, Status Results, Object Result, EmptyResult , Routing in ASP.NET Core MVC, Custom Routing, Custom Route Constraints in Web Application, Attribute Routing, Attribute Routing using Tokens, Attribute Routing vs Conventional Routing, Model Binding in ASP.NET Core MVC, Model Binding using- FromForm, FromQuery, FromRoute, FromHeader, FromBody, Complex Type, Custom Model Binding in ASP.NET Core MVC.

Unit 4: HTML, Tag Helper, Data Annotation Validation and State management [10]

HTML Helpers for-TextBox, TextArea, DropDownList, RadioButton, Check Box, ListBox, Password, Hidden, Custom HTML Helper in ASP.NET Core MVC, Creating Form Using HTML Helpers, Different Ways to Generate Links in ASP.NET Core MVC, Tag Helpers for-Image Tag , Environment Tag, Navigation Menus, Form Tag, Partial Tag, Creating Custom Tag Helper, View Component Tag Helper, Cache Tag Helper, Data Annotations, Model Validations, Data Annotation Attributes- Custom Data Annotation, Remote Validation, Blacklist and Whitelist Checks using Data Annotation, Displaying and Formatting Attributes, Real-Time Examples of Data Annotations in ASP.NET Core MVC, Cookies, Encrypt Cookies, Persistent vs Non-Persistent Cookies, Sessions, In-Memory vs Distributed Sessions, Differences Between Cookies and Sessions, Upload File, Restrict Uploaded File Size, Restrict Uploaded File Type, Save Uploaded file to Database, Display Images, Delete Images, Upload Multiple Files, Export Data to Excel File, Import Excel Data to Database, Generate PDF, Generate Password Protected PDF, Convert HTML to PDF, Send Email with Attachment.

Reference Books:

- Professional ASP.NET– Wrox Publication by Bill Evjen, Scott Hanselman, Farhan Muhammed, Sirnivasa Sivakumar, Devin Rader.
- Microsoft ASP.NET Step by Step - Microsoft Press by George Shepherd.

Type: DSE3-A

Course Title: ReactJS

(Paper Code: Paper XVI)

Course Objectives:

1. Understand Single Page React application is different than traditional web development frameworks.
2. Develop an application from scratch.
3. Understand the benefits of unidirectional data flow.

Course Outcomes:

1. To understand basic fundamentals of Front-end Technologies
2. To apply the advanced concepts of ReactJS and WebAPI
3. To develop ReactJS applications.

Unit 1: Introduction to ReactJS [20]

Introduction, Workflow, Scope, Pros and Cons, Difference between JS and JSX, React Components overview, Child Components, JSX expressions,

Building Blocks of ReactJS: JSX, Components, State and Props, Conditional Rendering, Why JSX, Advantages of JSX, Expressions in JSX, Implementation of JSX, Creating a react component with jsx

Environment Setups: Node setup, How to use NPM, Npm and Setting Environment for ReactJS projects, How to create package.json and purpose, IDE for ReactJS, ReactJS browser plugins overview.

Components: Types of components, Functional component vs Class Component, Converting Functional Components to Class Components, Component Life Cycles and its different methods.

Unit 2: Conditional Rendering and List [10]

if-else Statement, logical andand operator, operators, Preventing Component from Rendering, Switch case operator

List and Keys: react key prop, map function to iterate the List, References, use Refs, Create Refs, access Refs, Event Binding types: Bind () method, Arrow function

Props and State: What is a state, use and role of the state, what are props, Props validation, Passing data between multiple components, Managing Component State

Unit 3: Handling Events and Forms [10]

Lists of Form components, Setup Controlled and Uncontrolled form components, Control Input elements, Form Submission and Validation, how to set default values on all formats of Input elements, Form validations, writing Styles, Animations overview, Event, Event Binding, Event Handlers, Common React Events, Key Events, Event Pooling, Synthetic Event.

Unit 4: Routing and State Management [20]

Introduction to React Router, History of Router, Single Page Application Overview, configure React Router, Load the router library, Navigating between Routes, Route Parameters and Nested Routes, Dynamic Routing, Nesting Routes, Invalid URL, Handle Conditional statement in JSX

State Management: Local State vs. Global State, State Lift-Up, Context API for Global State

Redux: Introduction to Redux, Redux Architecture- Actions, Reducers, and Store, Provider Component, Dispatchers, View Controllers, Connecting React with Redux

Hooks: Introduction to Hooks, The useState hook, useEffect hook, Custom hook, useRef hook, useMemo hook, The useContext hook, The useReducer hook, Another Hooks.

Reference Books:

- Introduction to React By Cory Gackenheimer, Apress
- React and React Native: A complete hands-on guide to modern web and mobile development with React.js By Adam Boduch, Roy Derks
- React 16 Essentials: A fast-paced, hands-on guide to designing and building scalable and maintainable web apps with React 16 By Artemij Fedosejev, Ada

Type: DSE3-B

Course Title: Compiler Construction

(Paper Code: Paper XVI)

Course Objectives:

1. To learn the process of translating a modern high-level language to executable code.
2. To learn an understanding of the fundamental principles in compiler design and to provide the skills needed for building compilers for various situations that one may encounter in a career in Computer Science.
3. To develop an awareness of the function and complexity of modern compilers.
4. To apply the code generation algorithms to get the machine code for the optimized code.
5. To represent the target code in any one of the code formats
6. To understand the machine dependent code
7. To draw the flow graph for the intermediate codes.
8. To apply the optimization techniques to have a better code for code generation

Course Outcomes:

1. To gives you with both theoretical and practical knowledge that is crucial in order to implement a programming language.
2. It gives you a new level of understanding of a language in order to make better use of the language (optimization is just one example).

Unit 1: Introduction to compiling and Lexical Analysis [15]

Compiler, self-compiler, cross compiler, boot strapping , phases of compiler, compiler construction tools, a simple one pass, two pass and multi pass compiler, factor affecting pass structure of compiler.

Role of lexical analyzer, input buffering, specification and recognition of tokens, finite automata implications, designing a lexical analyzer generator

Unit 2: Syntax Analysis Syntax Directed Translation [15]

Role of Parser, writing grammars for context free environments, top down parsing, recursive descent and predictive parsers (LL), Bottom-up parser, Operator precedence Parsing, LR, SLR and LALR parsers.

Syntax directed definitions, construction of syntax tree, bottom-up evaluation of S-attributed definitions, L-attributed definitions, Top-down translation and Bottom – up evaluation of inherited attributes, analysis of syntax directed definitions.

Unit 3: Run time environments and Intermediate code generation [15]

Source language issues, storage organization and location strategies, parameter passing, symbol table organization and generation, dynamic storage allocation.

Intermediate languages, declarations, assignments statements and Boolean expressions, case statements, back patching, procedure calls.

Unit 4: Code generation and Code Optimization [15]

Issues in design of a code generator and target machine, run time storage management ,basic blocks and flow graphs, next use information and simple code generator, issue of register allocation, assignment and basic blocs, code generation from DAG and the dynamic code generation algorithm.

Source of optimization, peephole optimization and basic blocks loop in flow graphs, data flow analysis and equations, code improving transformation and aliases, data flow analysis and algorithms, symbolic debugging of optimized code.

Reference Books:

- Compilers Priciple, Techniques, Tools by Aho, Lam, Sethi and Ulman
- Compiler Design by Wihelm, Mauer
- Compiler Design : Theory, Tools and Examples by Bergamann

Type: DSE3-C

Course Title: Internet of Things

(Paper Code: Paper XVI)

Course Objectives:

1. To understand the concept of IoT and its applications.
2. To reduce import dependency on IoT components and promote indigenization.
3. To Understand the the fundamentals of IoT with its architecture, and protocols.
4. It also covers the overview and programming of two widely used IoT platforms Arduino and Raspberry Pi.

Course Outcome: Students should be able to

1. Demonstrate the architecture and functioning of IoT systems including the sensors and micro controllers with their interfacing and software needs considering application areas.
2. Diagnose the various IoT protocols with detailing of their elements and overall functioning within IoT systems for efficient communication.
3. Design an IoT system to take the benefit of the Clouds for computing and storage considering security issues.
4. Leverage the benefits of IoT technologies for automating the various real-life challenges in various application areas.
5. Develop the software components of the IoT system using Arduino/Raspberry Pi Programming.

Unit 1: Introduction **[20]**

What is IoT, why important, Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Application areas of IoT, Characteristics of IoT, Things in IoT, IoT stack, Enabling technologies, IoT challenges, IoT levels, IoT and cyber-physical system, IoT and WSN, Overview of Governance, Privacy and Security Issues.

IOT ARCHITECTURE: IoT Open source architecture (OIC)- OIC Architecture and Design principles- IoT Devices and deployment models- IoTivity : An Open source IoT stack - Overview- IoTivity stack architecture- Resource model and Abstraction.

Sensors, Microcontrollers, and Their Interfacing: Sensor interfacing, Types of sensors, Controlling sensors, Microcontrollers, ARM.

Unit 2: Protocols **[10]**

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFIDProtocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security.

Unit 3: Cloud and Applications **[10]**

IoT and cloud, Fog computing, Security in cloud, Case study.

Application Building with IoT: Various applications of IoT: Food, Healthcare, Lavatory maintenance, Water quality, Warehouse, Retail, Driver Assistance, and Collision impact**,** IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

Unit 4: Arduino and Raspberry Pi **[20]**

Arduino: Architecture, Programming and Application Raspberry Pi : Architecture, Programming and Application.

IoT Security: Various security issues and needs, architecture, requirements, challenges and algorithms.

Reference Books:

- Internet of Things, Vasudevan, Nagrajan and Sundaram, Wiley India
- IoT Fundamentals, David Hence at el, Cisco Press
- 21 IoT Experiments, Yashavant Kanetkar, Shrirang Korde, BPB
- IoT Based Projects, Rajesh Singh at el, BPB
- Internet of Things with ARDUINO and BOLT, Ashwin Pajankar, BPB

Credits: Practical's – (4)

Total Lectures: 60 Hrs. **Contact Hrs. (Pr):5**

University Evaluation: 80 Marks **Internal Evaluation:** 20 Marks

Project (Practical-IV)

Instructions:

1. Team size for major project not exceed than two students.
2. Real time and live project followed by Presentation and Viva-Voce.

Credits: Practical's – (4)

Total Lectures: 60 Hrs. **Contact Hrs. (Pr):5**

University Evaluation: 80 Marks **Internal Evaluation:** 20 Marks

Practical based on DSC1F and DSC2F

DSC1F (Advance Java)

1. Write a programme which demonstrates life cycle of Servlet
2. Write a programme by using GenericServlet
3. Write a programme by using HttpServlet
4. Write a Servlet programme to send request to another page
5. Write a Servlet programme to track the user by using (Cookies, URL rewriting, Hidden form field and HttpSession)
6. Write Jsp programme which will display its life cycle
7. Write a Jsp programme by using its implicit objects like request, response, out, page, pageContext,application, session, config, exception
8. Write a Jsp programme which will use scriplet, expression and declarative tag.
9. Write a Jsp programme which will create bean and calculate simple interest
10. Write a Jsp programme to create bean to check account balance(from database)
11. Write a Jsp programme to insert data into database
12. Write a Jsp programme which will use JSTL core tag,JSTL SQL tags, JSTL formatting tags,JSTL xml tags, Customtag: empty tag, body content tag,iteration tag, simple tag
13. Write a programme to display a message in different languages (use java internationalization)
14. Write a simple Hibernate programme
15. Write a HB with annotation
16. Write a HB web application
17. Write a HB Inheritance mapping: Table per Hierarchy(TPH), TPH using annotation,
18. Table Per Concrete (TPC), TPC using annotation,Table Per Subclass (TPS), TPS using annotation. Collection mapping: Mapping list, one to many by list, one to many by bag, one to many by set, one to many by map.
19. Write a Spring programme to show Dependency injection: constructor Injection (CI),CI dependant object, CI with collection, CI with map, CI inheriting bean
20. Write a Spring Spring JDBC programme using : JDBC template, Result set Exactor, Named Parameter, Simple JDBC template. Spring with Hibernate

DSC2F (Tools: Weka or Python)

1. How to create and load data set in Weka? Apply pre-processing operations on given attributes.

(Note: Use following Excel data file (Name it as weather.arff) for doing above activity).

Outlook	**Temp**	**Humidity**	**Windy**	**Class**
Sunny	75	70	yes	Play
Sunny	80	90	yes	Dontplay
Sunny	85	85	no	Dontplay
Sunny	72	95	no	Dontplay
Sunny	69	70	no	Play
Overcast	72	90	yes	Play
Overcast	83	78	no	Play
Overcast	64	65	yes	Play
Overcast	81	75	no	Play
Rain	71	80	yes	Dontplay
Rain	65	70	yes	Dontplay
Rain	75	80	no	Play
Rain	68	80	no	Play
Rain	70	96	no	Play

1. Create and load following data set in Weka. Perform various pre-processing operations on given attributes.

(Note : Use following Excel data set (Name it as All_Electronics.arff)).

Table: Class-Labeled Training Tuples from the All Electronics Customer Database

RID	**age**	**income**	**student**	**credit-rating**	**class**
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-age	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-age	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-age	medium	no	excellent	yes
13	middle-age	high	yes	fair	yes
14	senior	medium	no	excellent	no

2. Use ID3(Iterative Dichotomiser 3) algorithm to classify weather data from the “weather.arff” file. Perform initial pre-processing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.

3. Create the appropriate Regression Model with Weka by using following data set(Name it as house_selling_rate.arff).

House size (square feet)	Lot size	Bedrooms	Granite	Upgraded bathroom?	Selling price
3529	9191	6	0	0	$205,000
3247	10061	5	1	1	$224,900
4032	10150	5	0	1	$197,900
2397	14156	4	1	0	$189,900
2200	9600	4	0	1	$195,000
3536	19994	6	1	1	$325,000
2983	9365	5	0	1	$230,000
3198	9669	5	1	1	????

   Calculate the selling price of last house by using regression model.

4. Create the appropriate Regression Model with Weka by using following data set(iris.arff). Use J-48 and select 66% training data set and the remaining for testing. Visualize the result by using decision tree and analyze the result.
5. Load the soybean.arff file in weka, and use the classification method (use J48), analyse the confusion matrix, visualize the decision tree.
6. Practical's on Mining Associations Rule
7. Practical's on Classifications and predictions
8. Practical's on Clustering
9. Load the weather.nominal dataset. Use the following filters in weka.
   a. unsupervised.instance.RemoveWithValues to remove all instances in which the humidity attribute has the value high.
   b. Convert numeric value to nominal
   c. Convert nominal to string
   d. Discretizes data
10. Load the iris dataset. Use the following filter of weka.
   a. Add noise to last column (i.e. Class).
   b. Randomize the data
   c. Normalize the data
   d. Reorder the data
11. Implement and evaluate different classification algorithms available in Weka (e.g., Decision Trees, Naive Bayes, k-Nearest Neighbors).
12. Train the models using the training set and evaluate their performance on the testing set using various metrics like accuracy, precision, recall, and F1-score.
13. Compare the performance of different classifiers and identify the most suitable one for the dataset.
14. Apply regression algorithms available in Weka (e.g., Linear Regression, Polynomial Regression).

15. Train the regression models using the training set and evaluate their performance on the testing set using metrics like Mean Squared Error (MSE) and R-squared.
16. Implement and evaluate clustering algorithms available in Weka (e.g., k-Means, Hierarchical Clustering).
17. Explore different clustering techniques and their impact on clustering quality.
18. Visualize the clusters using Weka's visualization tools and analyze their characteristics.
19. Apply association rule mining algorithms (e.g., Apriori) to identify interesting patterns in the dataset.
20. Adjust parameters such as minimum support and confidence to control the quality of the discovered rules.
21. Interpret and analyze the discovered rules to gain insights into the dataset.
22. Implement feature selection techniques to identify the most informative features in the dataset.
23. Evaluate the impact of feature selection and dimensionality reduction on the performance of classification or clustering algorithms.
24. Implement ensemble learning techniques (e.g., Random Forest, AdaBoost) using Weka's ensemble classifiers.
25. Compare the performance of ensemble methods with individual classifiers and analyze the benefits of ensemble learning.

Credits: Practical's – (4)

Total Lectures: 60 Hrs. **Contact Hrs. (Pr):5**

University Evaluation: 80 Marks **Internal Evaluation:** 20 Marks

Practical based on DSC 1G and DSC 2G

DSC 1G

Practical on .NET Core

1. Create a simple console application that prints "Hello, World!" to the console.
2. Compile and run the application using the .NET Core CLI or Visual Studio.
3. Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric variables.
4. Use string interpolation or concatenation to display variable values.
5. Implement conditional statements (if-else, switch-case) to control program flow based on different conditions.
6. Use loops (for, while, do-while) to iterate over arrays, collections, or sequences of data.
7. Create nested loops and conditional statements for more complex control flow logic.
8. Declare and initialize arrays of different data types.
9. Access array elements using index notation and perform array manipulation operations (sorting, searching, etc.).
10. Define and call methods with different access modifiers (public, private, protected).
11. Pass parameters to methods and return values from methods.
12. Overload methods with different parameter types and number of parameters.
13. Create classes and objects to represent real-world entities.
14. Implement encapsulation, inheritance, and polymorphism concepts in C#.
15. Use constructors, properties, and methods to define the behavior of objects.
16. Implement try-catch blocks to handle exceptions and prevent application crashes.
17. Throw custom exceptions to handle specific error conditions.
18. Use finally blocks to execute cleanup code regardless of whether an exception is thrown.
19. Read from and write to text files using StreamReader and StreamWriter classes.
20. Implement file input/output operations such as reading, writing, appending, and deleting files.
21. Handle file exceptions and ensure proper resource management using IDisposable interface.
22. Use LINQ queries to perform filtering, sorting, grouping, and aggregation operations on collections.
23. Create and manage multiple threads using the Thread class or ThreadPool.

DSC 2G

Practical on ASP.NET Core MVC

1. Create a new ASP.NET Core MVC project using Visual Studio or the .NET CLI.
2. Explore the project structure and understand the role of important files such as Startup.cs, Program.cs, and the Views folder.
3. Define model classes representing entities in the application domain.
4. Generate scaffolded controllers and views using Entity Framework Core for CRUD operations on the model classes.
5. Customize the generated views and controllers to meet specific requirements.
6. Define custom routes using attribute routing and convention-based routing.
7. Implement route constraints to restrict the format of URL parameters.
8. Deminstrate how routing works and how URLs map to controller actions.
9. Create HTML forms for user input and data submission.
10. Implement form validation using data annotations and ModelState.IsValid.
11. Bind form data to model properties using model binding techniques.
12. Working with Entity Framework Core:
13. Set up a database context and configure entity classes for use with Entity Framework Core.
14. Perform database migrations to create or update the database schema based on changes to the model classes.
15. Implement CRUD operations (Create, Read, Update, Delete) using Entity Framework Core methods.
16. Create RESTful API endpoints for accessing application data using ASP.NET Core MVC controllers.
17. Implement HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations on resources.
18. Use attribute routing and model binding to define API routes and handle incoming requests.
19. Validate form input and handle form submissions using JavaScript before sending requests to the server.

Credits: Practical's – (4)

Total Lectures: 60 Hrs. **Contact Hrs.(Pr): 5**

University Evaluation: 80 Marks **Internal Evaluation:** 20 Marks

Practical based on DSE 1A / DSE 1B / DSE 1C and DSE 2A / DSE 2B / DSE 2C

DSE 1A

Practical Assignments on DSE 1A: (Advanced Python)

1. Write a JDBC program to insert, update, delete and display records from databases using prepared statement.
2. Write a JDBC program to call stored procedures using callable statement.
3. Write a program to draw different shapes
4. Write a program to develop GUI applications
5. Write a program to implement Thread Synchronization.
6. Write a program to create simple Django app

2. Write a program to create simple Django project.
3. Write a program to create Django project which add, delete, update records.
4. Build a simple application with Tkinter that includes widgets such as labels, buttons, entry fields, and text areas and arrange widgets using different geometry managers (pack, grid, place) to create the desired layout. Validate user input and provide feedback for invalid entries using message boxes or labels.
5. Design a simple calculator application using Tkinter with buttons for numeric input and arithmetic operations.
6. Develop a CRUD (Create, Read, Update, Delete) application with Tkinter for managing a simple database-backed model.
7. Develop application which display employee records in tabular format.
8. Extend Tkinter by creating custom widgets with unique functionality or appearance. Implement custom widgets such as color pickers, sliders, or progress bars using Tkinter's canvas or frame widgets.
9. Design a menu-driven application with Tkinter that includes dropdown menus, cascading menus, and context menus.
10. Implement file dialogs, message boxes, and other common dialogs to interact with the user and perform actions such as opening, saving, and closing files.
11. Define URL patterns, views, and templates to render dynamic HTML pages using Django's templating engine.

12. Develop a CRUD (Create, Read, Update, Delete) application with Django for managing a simple database-backed model.
13. Create views and templates for listing, adding, editing, and deleting records from the database.
14. Implement form validation and error handling to ensure data integrity and user-friendly interactions.
15. Integrate user authentication and authorization into a Django application using Django's built-in authentication system. Create user registration and login forms, and implement password reset functionality. Restrict access to certain views or functionalities based on user roles and permissions.
16. Define models with relationships (one-to-one, one-to-many, many-to-many) to represent complex data structures. Implement CRUD operations for related models and navigate relationships using Django's ORM (Object-Relational Mapper).
17. Use Django's admin interface to manage database records and relationships.

DSE 1B

Practical Assignments on DSE 1B: (Theory of Computer Science)

Tool: JFLAP

The students are expected to understand JFLAP tools and designing suitable automata to recognize the following and test the output as string accepted or rejected.

1. Design a DFA which accept string is start with a over ∑={a,b}.
2. Design a DFA which accept number is even or odd.
3. Design a DFA which accept string either ending with ab or bc over ∑={a,b,c}.
4. Design a DFA which accept string does not having abc as substring over ∑={a,b,c}.
5. Design DFA which accept strings length is odd over ∑={0,1}.
6. Design Moore machine for finding binary is even or odd.
7. Design a Mealy machine for 1’s complement of binary number.
8. Design Mealy machine for accepting 2’s complement of binary number.
9. Removal unit production of following grammar;

I) A → BB A→a B →b	II) S→ A S → bb A → B A → b B → a B → S

10. Removal ε-production of following grammar;

I) S ⟶ aSa S ⟶ bSb S⟶ ε	II) S ⟶ a S ⟶ Xb S ⟶ aYa X ⟶ Y X ⟶ ε Y⟶ b Y ⟶ ε

11. Convert following Context Free Grammar(CFG) to Chomsky Normal Form (CNF);

I) S⟶ABAB A⟶ Aa A⟶ a B⟶ b	II) S ⟶ ABA A ⟶ aA A ⟶ ε B ⟶ bB B ⟶ ε

12. Design a PDA for accepting palindrome string over ∑= {0,1}.

13. Design a PDA to check whether a given string over {a,b} ends in abb.

14. Design TM for L= {a^nb^n| n>1}.

15. Construct Turing machine for copy string over ∑={a,b}.

16. Construct Turing Machine that recognize the language:

17. L= {x ϵ $\{0,1\}^*$| x ends in 00}.

DSE 1C

Practical Assignments on DSE 1C: (Mobile Application Development)

1. Develop an application that uses GUI components, Font and Colours

2. Develop an application that uses Layout Managers and event listeners.

3. Develop a native calculator application.

4. Write an application that draws basic graphical primitives on the screen.

5. Develop an application that makes use of database.

6. Develop an application that makes use of RSS Feed.

7. Implement an application that implements Multi-threading

8. Develop a native application that uses GPS location information.

9. Implement an application that writes data to the SD card.

10. Implement an application that creates an alert upon receiving a message.

11. Write a mobile application that creates alarm clock.

DSE 2A

Practical Assignments on DSE 2A: (ReactJS)

1. How useEffect works in ReactJS?
2. How to concatenate unicode and variable in ReactJS?
3. How to pass data from one component to other component in ReactJS?
4. How to set input box to be a floating number in ReactJS?
5. How to use useState in arrow function instead of hook?
6. How to add theme to your React App?
7. How to add a function in JSX?
8. How to fetch data from APIs using Asynchronous await in ReactJS?
9. How to get cell value on React-Table?
10. How to access nested object in ReactJS?
11. How to set default value in select using ReactJS?
12. How to change the position of the element dynamically in ReactJS?
13. How to solve too many re-renders error in ReactJS?
14. How to publish a ReactJS component to NPM?
15. How to change the navbar color when you scroll in ReactJS?
16. How React Native is different from ReactJS?
17. How to Create a Countdown Timer Using ReactJS?
18. How to Create a Navigation Bar with Material-UI?
19. How to locally manage component’s state in ReactJS?
20. How to add a CSS class whenever the component is updated in ReactJS?
21. How to Create a Toggle Switch in React as a Reusable Component?
22. How to create a custom progress bar component in React.js?
23. How to pass data from one component to other component in ReactJS?
24. How to convert functional component to class component in ReactJS?
25. How to put ReactJS component inside HTML string?
26. How to use CssBaseLine Component in ReactJS?
27. How to use ToggleButtonGroup Component in ReactJS?
28. How to use componentWillMount() in React Hooks?
29. How to use Link Component in ReactJS?
30. How to use BottomNavigation Component in ReactJS?
31. How to use Breadcrumbs Component in ReactJS?
32. How to use TextField Component in ReactJS?
33. How to use Portal Component in ReactJS?
34. How to use ScopedCssBaseline Component in ReactJS?
35. How to use Popper Component in ReactJS?
36. How to use Slide Component in ReactJS?
37. How to use Grow Component in ReactJS?

38. How to use Collapse Component in ReactJS?
39. How to use Fade Component in ReactJS?
40. How to use Zoom Component in ReactJS?
41. How to use Popover Component in ReactJS?
42. How to apply validation on Props in ReactJS?
43. What is prop drilling and how to avoid it?
44. How to create new elements with ReactJS mapping props?
45. How to pass multiple props in a single event handler in ReactJS?
46. How to send state/props to another component in React with onClick?
47. How to create a translucent text input in ReactJS?
48. How to set an object key inside a state object in React Hooks?

DSE 2B

Practical Assignments on DSE 2B: (Compiler Construction)

1) Convert following Grammar to LL Grammar;

I) S→ ABcC A → aA A → ϵ B→ bbB B → ϵ C → BA	II) S→ ABcC A → aA A → ϵ B→ bbB B → ϵ C → BA

2) Build LL(1) parse table for following Grammar and find out LL(1) or not;

I. S → A A→ aaA A → b	II.E → E+T T → T*F F → (E) F → a
III. S → aA S → a A → a	IV. S →AS S → a A → SA A → b
V.S → BAc A → aA A → a B→ AB B → bB B → d	

3) Build SLR(1) parse table for following Grammar and find out LL(1) or not;

I. S → A S → B A → A A → b B → B B → b	II. E → E+T T → T*F F → (E) F → a
III. S → AA S → aA A → b	IV. S → A A → aA A → b

V. S → Ab A→ aA A → ab A → ϵ	

DSE2-C

Practical Assignments on DSE 2B: (Internet of Things)

1. Write a program to sense the available networks using Arduino.

2. Write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.

3. Write a program to detect the vibration of an object with sensor using Arduino.

4. Write a program to connect with the available Wi-Fi using Arduino.

5. Write a program to sense a finger when it is placed on the board Arduino.

6. Write a program to get temperature notification using Arduino.

7. Write a program for LDR to vary the light intensity of LED using Arduino.

8. Write a program to install MySQL database in Raspberry pi.

9. Write a program to work with basic MySQL queries by fetching data from database in Raspberry pi.

10. Write a program to switch light on when the input is 1 and switch the light off when the input is 0 using Raspberry pi.

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

Equivalent Subject for Old Syllabus B.Sc. (ECS) - III (Semester–V and VI)

Semester-V		
Sr. No.	**Name of the Old Paper (w.e.f. 2021-2022)**	**Name of the New Paper (w.e.f. 2024-2025)**
1	English (BusinessEnglish)	English (Business English)
2	Data Communicationand Networking	Data Communicationand Networking
3	Theory of Computer Science	Theory of Computer Computation
4	Visual Programming	Dot NET Core
5	Advanced Java	Advanced Java
6	Advanced Python Programming	Advanced Python
Semester-VI		
Sr. No.	**Name of the Old Paper (w.e.f. 2021-2022)**	**Name of the New Paper (w.e.f. 2024-2025)**
1	English (BusinessEnglish)	English (BusinessEnglish)
2	System Security	Network Security
3	Compiler Construction	Compiler Construction
4	Internet Programmingusing ASP.Net	ASP .Net Core MVC
5	Angular JS	ReactJS
6	Mobile Application Development	Mobile Application Development