



Karmaveer Bhaurao Patil University, Satara

**Syllabus for
B. Sc. III (Computer Science)**

**Under
Faculty of Science and Technology
(As per NEP 2020)**

With effect from Academic Year 2025-2026

Syllabus for Bachelor of Science (Computer Science) Part III

PREAMBLE:

Bachelor of Science is an integrated academic degree in the faculty of science. The revision of existing syllabus of Computer Science subject in Science Faculty is essential. This is a humble endeavor to initiate the process towards an era of knowledge. The students from science faculty should also be competent for this change in the technology. In this year, a student will be able to handle computers, develop the programs in languages and other peripherals with confidence. In the subject, the student will also get a basic and proper knowledge in the field of Programming skills.

GENERAL OBJECTIVES OF THE COURSE:

1. To learn basics of Computer, hardware, software, networking.
2. To inculcate the software development attitude and generate interest in the field of Technology.
3. To develop programming skills, management skills, writing skills, Project Analysis skill among students.
4. To inculcate research attitude among students.

PROGRAM SPECIFIC OUTCOMES (PSO's):

1. Ability to design, develop, implement computer programs and use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.
2. To work with and communicate effectively with professionals in various fields and pursue lifelong professional development in computing.
3. To analyse, design, develop, test and apply management principles, mathematical foundations in the development of computational solutions, make them to expert in designing the computer software and hardware.
4. Develop their skills to solve problems in the broad area of programming concepts and appraise environmental and social issues with ethics and manage different projects in inter- disciplinary field.

1. **TITLE: COMPUTER SCIENCE**

2. **YEAR OF IMPLEMENTATION: 2025-2026**

3. **DURATION: 01 year**

4. **PATTERN: Semester examination**

5. **MEDIUM OF INSTRUCTION: English**

6. STRUCTURE OF COURSE:

As per NEP-2020 (1.0)

Level	Sem.	Subject				VSC	FP	CEP	OJT	Total
		DSC		DSE*						
		T	P	T	P					
5.5	V	DSC I (2)	DSCP (4)	DSE I (2)	DSE P (2)	VSC (4)	FP (2)	CEP (2)	--	22
		DSC II (2)		DSE II (2)						
5.5	VI	DSC I (2)	DSCP (4)	DSE I (2)	DSE P (2)	VSC (2)	FP (2)	--	OJT (4)	22
		DSC II (2)		DSE II (2)						

* DSE - 2 Courses out of four for each semester.

7. COURSE TITLE

1) Fifth Semester

Sr. No.	Subject	Theory				Practical		
		Course No. & Course Code	Title of Paper	No. of lectures per week	Credits	Course No. & Course Code	No. of Practical Per week	Credits
1.	DSC I	BCST 351	C# Programming -I	2	2	BCSP 357	1	2
2.	DSC II	BCST 352	Programming Essentials in Python	2	2	BCSP 358	1	2
3.	DSE I	BCST 353	Introduction to mobile Application	2	2	BCSP 359	1	2
4.	DSE I	BCST 354	IOT					
5.	DSE II	BCST 355	Advanced JAVA Programming	2	2			
6.	DSE II	BCST 356	Multimedia Computing					
7.	VSC	--	--	--	--	BCSPVSC III	1	2
		--	--	--	--	BCSPVSC IV	1	2
8.	FP	--	--	--	--	BCSPFP I	1	2
9.	CEP	--	--	--	--	BCSPCEP I	1	2

2) Sixth Semester

Sr. No.	Subject	Theory				Practical		
		Course No. & Course Code	Title of Paper	No. of lectures per week	Credits	Course No. & Course Code	No. of Practica l Per week	Credits
1.	DSC I	BCST 361	C# Programming -II	2	2	BCSP 367	1	2
2.	DSC II	BCST 362	Introduction to Artificial Intelligence (AI)	2	2	BCSP 368	1	2
3.	DSE I	BCST 363	Advanced mobile Application development	2	2	BCSP 369	1	2
4.	DSE I	BCST 364	Web technology					
5.	DSE II	BCST 365	Computer Graphics	2	2			
6.	DSE II	BCST 366	E – Commerce					
7.	VSC	--	--	--	--	BCSP VSC V	1	2
8.	FP	--	--	--	--	BCSP FP II	1	2
9.	OJT	--	--	--	--	BCSP OJT I	2	4

8. EVALUATION STRUCTURE:

➤ B.Sc. III NEP 1.0

Semester V (5.5)

Course	Course Category	Course Code	Internal Evaluation			ESE	Total Marks	Credits
			CCE-I	Mid Semester	CCE-II			
DSC	T	BCST 351	05	10	05	30	50	02
	T	BCST 352	05	10	05	30	50	02
	P	BCSP 357	--	--	--	50	50	02
	P	BCSP 358	--	--	--	50	50	02
DSE (2 Theory Papers Out of Four)	T	BCST 353	05	10	05	30	50	02
	T	BCST 354	05	10	05	30	50	02
	T	BCST 355	05	10	05	30	50	02
	T	BCST 356	05	10	05	30	50	02
	P	BCSP 359	--	--	--	50	50	02
VSC	P	BCSPVSC III	--	--	--	50	50	02
	P	BCSPVSC IV	--	--	--	50	50	02
FP	P	BCSPFP I	--	--	--	50	50	02
CEP	P	BCSPCEP I	--	--	--	50	50	02
Total							550	22

Semester VI

Course	Course Category	Course Code	Internal Evaluation			ESE	Total Marks	Credits
			CCE-I	Mid -Semester	CCE-II			
DSC	T	BCST 361	05	10	05	30	50	02
	T	BCST 362	05	10	05	30	50	02
	P	BCSP 367	--	--	--	50	50	02
	P	BCSP 368	--	--	--	50	50	02
DSE (2 Theory Papers Out of Four)	T	BCST 363	05	10	05	30	50	02
	T	BCST 364	05	10	05	30	50	02
	T	BCST 365	05	10	05	30	50	02
	T	BCST 366	05	10	05	30	50	02
	P	BCSP 369	--	--	--	50	50	02
VSC	P	BCSPVSC V	--	--	--	50	50	02
FP	P	BCSPFP II	--	--	--	50	50	02
OJT	P	BCSPOJT I	--	--	--	100	100	04
Total							550	22

DSC: Discipline Specific Course; DSE: Discipline Specific Elective VSC: Vocational Skill Course; OJT: On Job Training; FP: Field Project; CEP: Community Engagement Program, T: Theory; P: Practical

9. OTHER FEATURES:

A) LIBRARY:

Reference books, Textbooks, journals, and Periodicals are available in Institute and Departmental Library. (Separate reference lists are attached along with the respective course syllabus)

B) EQUIPMENT:

a) Computer, LCD projector, visualizer, smart board

B. Sc. Part III, Semester V Computer Science

BCST 351: C# Programming -I

(Total Credit – 02)

Course Objectives: Students should be able to...

1. To Students will gain the ability to implement the algorithms in C#.net, VB.net and ASP.net.
2. To provide the knowledge of Dot Net Frameworks along with C#.
3. To analyze object-oriented paradigm in the C # programming language.
4. To understand different windows-based applications using standard controls and components.

UNIT	Content	No. of hours per unit
I	.NET Architecture	08
	1.1 Block diagram of .net framework, The Common Language Runtime, Advantages of Managed Code 1.2 A Closer Look at Intermediate Language & Assemblies, Support for Object Orientation and Interfaces 1.3 Distinct Value and Reference Types 1.4 Strong Data Typing, Garbage Collection	
II	C# Basics	07
	2.1 Getting Started with Introducing C#, Inside a C# Program 2.2 Compiling and Running the Program, Variables, Data Types, FlowControl, Enumerations, Namespaces 2.3 Namespace Aliases, The Main() Method, Multiple Main() Methods, defining & using functions & its scope, Passing Arguments to Main(), Parameter passing technique, Console 2.4 I/O, Classes and Structs, ClassMembers, DataMembers, Function Members, read-only Fields, properties and indexer 2.5 The Object Class, System.Object Methods, The ToString() Method, Delegates, Types of Delegates, Events, The Receiver's View of Events, Generating Events.	
III	Inheritance and Polymorphism	07
	3.1 Introduction-Types of Inheritance, Implementation Inheritance, Abstract Classes and Functions Sealed Classes and Functions 3.2 Constructors in Derived Classes, Interfaces, Defining and Implementing Interfaces, Derived Interfaces 3.3 Polymorphism, Method overloading, Operator overloading	

IV	Windows base application	08
	4.1 Creating a Windows Form Application, Standard Controls and Components, Properties and Events of the controls 4.2Forms, FormClass, Multiple Document Interface (MDI), Custom Controls (user Controls)	

Course outcomes - Students should be able to

1. Use the features of Dot Net Framework along with the features of C#.
2. Develop correct, well-documented programs using the C# programming language.
3. Learn to develop object-oriented programs using C# classes and objects.
4. Learn to use Windows Forms and WPF to create GUI-based programs.

Reference Books

1. Robinson, S., Nagel, C., Watson, K., Glynn, J., Skinner, M., & Evjen, B. Professional C#. Indianapolis: Wrox Publications, 2008.
2. Archer, T. Inside C#. Redmond: Microsoft Press, 2001.
3. Liberty, J., Maharry, D., & Hurwitz, D. Programming ASP.NET 3.5. Sebastopol: O'Reilly Media, 2008
4. Blackburn, P.D. ADO.NET Examples and Best Practices for C# Programmers. Berkeley: Apress, 2002.
5. Thomsen, C. Database Programming with C#. Berkeley: Apress, 2002.

B. Sc. Part III, Semester V Computer Science
BCST 352: Programming Essentials in Python
(Total Credit – 02)

Course Objectives: Students should be able to...

1. To understand programming skills in core Python.
2. To acquire Object Oriented Skills in Python
3. To develop the skill of designing Graphical user Interfaces in Python.
4. To inculcate Problem solving and programming capability.

UNIT	Content	No. of hours per unit
I	Overview of Programming	08
	1.1 History, features, installation, Writing and executing Python scripts, Python IDEs (IDLE, VS Code, Jupyter) 1.2 Structure of a Python Program, Elements of Python, Hello world application, and a more interesting program 1.3 Interpreters, modules, Variables, Names and Assignment, 1.4 Types Input and Output Statements, 1.5 Indentation, Atoms, Identifiers and keywords,	
II	Datatypes and operators	07
	2.1 Data types- int, float etc., Literals, Strings, 2.2 Basic Data Structures: - Lists, Tuples, Sets, Dictionaries, 2.3 Indexing, slicing, built-in functions, 2.4 Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator).	
III	Creating Python Programs and Structures	07
	3.1 Control statements (Looping- while Loop, for Loop , Loop Control, 3.2 Conditional Statement- if...else, Difference between break, continue and pass), 3.3 Date & Time, Modules, 3.4 Functions: - Defining and calling functions, Arguments, return values, *args, **kwargs, lambda functions, Recursion, 3.5 Modules & Packages:- Importing modules (import, from, as), Standard libraries (math, random, datetime), Creating user-defined modules 3.6 Exception Handling:- try, except, finally, else, Built-in exceptions, custom exceptions, 3.7 File Handling: Opening/closing files, Reading/writing text	

	and CSV files, File methods and modes	
IV	Advanced Python	08
	4.1 Objects and Classes, Constructor and destructor 4.2 Inheritance, polymorphism, encapsulation, abstraction 4.3 Regular Expressions, 4.4 Event Driven Programming, GUI Programming.	

Course outcomes - Students should be able to

1. Understand why Python is a useful scripting language for developers.
2. Learn how to design and program Python applications.
3. Explain basic principles of Python programming language
4. Implementing database and GUI applications.

Reference Books:

1. Brown, M.C. Python: The Complete Reference. New Delhi: McGraw Hill Education, 2018.
2. Budd, T. Exploring Python. New Delhi: Tata McGraw-Hill, 2011.
3. Severance, C. Python for Informatics: Exploring Information. Ann Arbor: CreateSpace Independent Publishing, 2013.
4. Lutz, M. Learning Python. Sebastopol: O'Reilly Media, 2009.
5. Fangohr, H. Introduction to Python for Computational Science and Engineering (A Beginner's Guide). Southampton: University of Southampton, 2005.
6. Gutttag, J.V. Introduction to Computation and Programming Using Python. New Delhi: Prentice Hall of India, 2013.

B. Sc. Part III, Semester V Computer Science
BCST 353: Introduction to mobile application development
(Total Credit – 02)

Course Objectives: Students should be able to...

1. To introduce students to mobile application development using cross-platform frameworks.
2. To provide foundational knowledge of mobile UI/UX design, app architecture, and state management.
3. To enable students to develop simple, functional mobile apps with modern tools.
4. To expose students to testing and deployment practices on emulators and real devices.

UNIT	Content	No. of hours per unit
I	Mobile App Development and its development environment	08
	1.1 Overview of mobile platforms: Android, iOS 1.2 Native vs. Cross-platform development 1.3 Tools and environments (IDE, SDKs, Emulators) 1.4 Introduction to chosen framework (e.g., Flutter or React Native) , Installing and configuring tools (Android Studio, VS Code, etc.)	
II	Building Basic UI	07
	2.1 UI components: Buttons, Text fields, Images, Lists. 2.2 Layouts and navigationss. Building User Interfaces: Fundamental Android UI design, Layouts – Linear, Relative, Grid and Table Layouts. User Interface (UI) Components.	
III	Fragments	07
	3.1 Fragments – Creating fragments, 3.2 Lifecycle of fragments, Fragment states 3.3 Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions 3.4 Interfacing between fragments and activities.	
IV	Core App Functionality, Testing and Deployment	07
	4.1 State management basics. 4.2 Data storage (local/shared preferences) 4.3 Basic use of APIs (fetching JSON from an endpoint) 4.4 Testing on emulators and physical devices 4.5 Debugging and performance checks 4.6 Building and deploying the app	

Course outcomes - Students should be able to

1. Understand the basics of mobile platforms, development environments, and frameworks.
2. Design and implement user interfaces for mobile applications.
3. Handle user inputs, events, and state management effectively.
4. Interact with device features and external APIs for enhanced app functionality.

References

1. Napoli, M.L. “Beginning Flutter: A Hands-On Guide to App Development”. Hoboken: Wiley, 2020.
2. Phillips, B., Stewart, C., Hardy, K., & Marsicano, B. “Android Programming: The Big Nerd Ranch Guide (4th ed.)”. Atlanta: Big Nerd Ranch Guides, 2019.
3. Gregoire, J.F. “Learning Android Application Programming for the Kindle Fire: A Hands-On Guide to Building Your First Android Application”. Boston: Addison-Wesley, 2014.
4. Mednieks, Z., Dornin, L., Meike, G., & Nakamura, M. “Programming Android: Java Programming for the New Generation of Mobile Devices”. Sebastopol: O’Reilly Media, 2012.
5. Darcey, L., & Conder, S. “Android Wireless Application Development (3rd ed., Volume I and II)”. Boston: Addison-Wesley, 2012.

B. Sc. Part III, Semester V Computer Science

BCST 354: IOT

(Total Credit – 02)

Course Objectives: Students should be able to...

1. To learn the basics of Introduction to IoT.
2. To understand the state of Art -IoT Architecture & Market perspective model.
3. To understand the Android OS architecture and understand the application development lifecycle.
4. To study Android's APIs for data storage, retrieval, user preferences, files and content providers.

UNIT	Content	No. of hours per unit
I	Introduction to IoT Architecture	08
	1.1 Introduction to IoT: - Definition and characteristics. 1.2 Web of Things V/s Internet of Things: -Two pillars of the web, 1.3 Architecture standardization for IoT, 1.4 Platform middleware for IoT, 1.5 Unified multitier IoT architecture, 1.6 IoT portals and Business Intelligence. 1.7 M2M to IoT: M2M Communication, 1.8 Trends in Information and Communication Technology, 1.9Implications for IoT, IoT Architecture: Building architecture, 1.10 Main design principles and needed capabilities, An IoT architectural overview. IoT Reference Model: IoT domain model.	
II	Embedded IoT devices	07
	2.1Embedded IoT devices: Sensors and actuators for IoT applications, 2.2IoT components and implementation, 2.3 Programming of Node MCU and Raspberry PI, 2.4Implementation of IoT with Edge devices, 2.5Reading sensor data and transmit to cloud, 2.6 Controlling devices through cloud using mobile application and web application	
	IoT Protocols	07

III	3.1 Link layer protocols, 3.2 Network/internet layer protocols, 3.3 Transport layer protocols, Application layer protocols: Hypertext transfer protocol (HTTP), 3.4 Systematic HTTP access methodology, 3.5 Web Socket, Constrained application protocol CoAP), Message Queue Telemetry Transport Protocol (MQTT), XMPP, DDS, AMQP	
IV	IoT Applications and case study 4.1 Broad categories of IoT applications: Consumer IoT, 4.2 Commercial IoT, 4.3 Industrial IoT, Infrastructure IoT, 4.4 Military Things (IoMT), 4.5 IoT Case studies: Home automation with IoT, River water pollution monitoring, Smart city street light control and monitoring, Health care monitoring, Voice Apps on IoT device	08

Course outcomes - Students should be able to

1. Learn IoT architecture.
2. Understand program Embedded IoT devices.
3. Use IoT protocol to upload sensor data and to control devices.
4. Design IoT application

References

1. Dubey, R. "An Introduction to Internet of Things: Connecting Devices, Edge Gateway, and Cloud with Applications". New Delhi: Cengage India, 2019.
2. Kamal, R. "Internet of Things: Architecture and Design Principles". New Delhi: McGraw Hill Education, 2017.
3. Madiseti, V., & Bahga, A. "Internet of Things (A Hands-on Approach)". New York: VPT, 2015.
4. McEwen, A., & Cassimally, H. "Designing the Internet of Things". Chichester: Wiley, 2013.
5. Kanetkar, Y. "21 Internet of Things Experiments". Kindle edition, 2016.
6. Javed, A. "Building Arduino Projects for Internet of Things". Berkeley: Apress, 2017.
7. Noris, D. "The Internet of Things: Do it Yourself Projects with Arduino, Raspberry Pi, and Beagle Bone Black". New York: McGraw Hill Education, 2015.

B. Sc. Part III, Semester V Computer Science
BCST 355: Advanced JAVA Programming
(Total Credit – 02)

Course Objectives: Students should be able to...

1. To obtain the basic knowledge of object-oriented programming, concepts of basic JAVA, advanced JAVA and Server-side scripting (JSP).
2. To develop distributed business applications, develop web pages using advanced server-side programming through servlets and Java server pages.
3. To design and develop error-free, well-documented Java programs.
4. To Learn how to write, test, and debug advanced-level Object-Oriented.

UNIT	Content	No. of hours per unit
I	Graphic Programming using swing	08
	1.1 Working with 2D Basic shapes, using Color, Using Font, Displaying Images, 1.2 Swing-1 Concept. MVC architecture. 1.3 Component of swing: JFrame, JComponent, JLabel, JTextfields, JCheckbox, JPanel, JRadiobuttons, 1.4 JTabbedPane, JButton, JTree, JTable, Jmenu. 1.5 Difference between AWT and Swing.	
II	AWT and Event handling	07
	2.1 Introduction of servlet. 2.2 Life cycle of servlet. 2.3 Session, cookies, servlet jdbc connection. 2.4 Components of JSP: 2.5 Directives. 2.6 Tags. 2.7 Scripting elements, 2.8 Simple application.	
III	Servlet and jsp	07
	3.1 Introduction of servlet. 3.2 Life cycle of servlet. 3.3 Session, cookies, servlet jdbc connection. 3.4 Components of JSP: 3.5 Directives, Tags, Scripting elements. 3.6 Simple application using jsp.	
	JDBC	08

IV	4.1 JDBC Introduction. 4.2 JDBC Architecture. 4.3 Types of JDBC. 4.4 Drivers. 4.5 The Connectivity Model. 4.6 The java.sql package. 4.7 Navigating the Result Set object's contents and Manipulating records of a Result Set object through User Interface. 4.8 The JDBC Exception classes, Database connectivity, Data 4.9 Manipulation (using Prepared Statements, Joins, Transactions, Stored Procedures), Data navigation.	
----	--	--

Course outcomes - Students should be able to

1. To design a graphical user interface (GUI) with Java Swing API.
2. To use event handling in Java applications and to draw various shapes using AWT Components.
3. To develop JSP applications using JSP Tags, JSP Scriptlets and JSP Application Models.
4. To evaluate solid Java applications using Java Database Connectivity (JDBC) to interact with relational databases and how to do fundamental database activities utilizing JDBC (Java Database Connectivity) API.

References

1. Schildt, H. "Java: The Complete Reference". New York: McGraw-Hill Education, 2018.
2. Bloch, J. "Effective Java". Boston: Addison-Wesley, 2018.
3. Gosling, J., Joy, B., & Steele, G. "The Java Programming Language". Boston: Addison-Wesley, 2000.
4. Horstmann, C.S. "Core Java Volume I—Fundamentals". Upper Saddle River: Prentice Hall, 2018.
5. Carnell, J. "Java 8 in Action". Shelter Island: Manning Publications, 2014.

B. Sc. Part III, Semester V Computer Science
BCST 356: Multimedia Computing
(Total Credit – 02)

Course Objectives: Students should be able to...

1. Understand core concepts of multimedia systems and their components.
2. Work with various multimedia data types like audio, video, and images.
3. Learn compression techniques and file formats.
4. Develop simple multimedia applications.
5. Understand multimedia retrieval and streaming technologies.

UNIT	Content	No. of hours per unit
I	Introduction to Multimedia	08
	1.1 Definition and Components of Multimedia 1.2 Multimedia Systems Architecture 1.3 Applications of Multimedia (Education, Entertainment, Web, etc.) 1.4 Multimedia Authoring Tools	
II	Text and Image Processing	07
	2.1 Types of Text (Formatted, Unformatted) 2.2 Unicode Standards and Text Compression (Huffman, Run Length Encoding) 2.3 Digital Image Representation (Bitmap, Vector) 2.4 Image File Formats: JPEG, PNG, GIF 2.5 Basic Image Processing: Color models, Filtering, Histogram	
III	Audio Processing	07
	3.1 Basics of Sound and Digital Audio 3.2 Audio File Formats: WAV, MP3, AAC 3.3 Audio Sampling and Quantization 3.4 Audio Compression (Lossless & Lossy) 3.5 Speech Synthesis and Recognition	
IV	Multimedia Databases and Retrieval	08
	4.1 Multimedia Data Storage and Retrieval 4.2 Content-Based Image Retrieval (CBIR) 4.3 Indexing and Querying Multimedia Content 4.4 Metadata and Annotation	

Course outcomes - Students should be able to

1. Implement basic operations on images, audio, and video using appropriate software tools and programming libraries.
2. Design and develop simple multimedia applications integrating multiple media types.
3. Understand and evaluate multimedia delivery over networks, including streaming protocols and QoS requirements.
4. Demonstrate the ability to work with multimedia databases and apply basic content-based retrieval techniques.

References

1. Pratt, W. "Multimedia: Making It Work, 9th Edition". New York: McGraw-Hill, 2014.
2. Zhang, H. "Fundamentals of Multimedia, 2nd Edition". Berlin: Springer, 2014.
3. Ralf, K., & Shyamsundar, P. "Multimedia: Computing, Communications and Applications". Upper Saddle River: Pearson Education, 2004.
4. Vaughan, T. "Digital Multimedia, 3rd Edition". Hoboken: Wiley, 2009.
5. Stallings, W. "Multimedia Communications: Applications, Networks, Protocols and Standards". Upper Saddle River: Pearson Education, 2001.

B. Sc. Part III, Semester V Computer Science

BCSP 357: Practical's based on BCST 351 C# Programming -I

(Total Credit – 02)

Course Objectives: Student will be able to

1. Understand the .NET framework and its architecture.
2. Learn the fundamentals of C# syntax, data types, and control structures.
3. Implement object-oriented programming concepts using C# (e.g., classes, inheritance, polymorphism).
4. Develop Windows-based applications using forms and events.
5. Work with file input/output and exception handling in C#.

Practical no.	Content	No. of hours per practical
1.	Write a menu driven of a] Face value b] Armstrong c] Palindrome.	04
2.	Write a program to Show Garbage Collection.	04
3.	Write a program to demonstrate Value Type and Reference Type.	04
4.	Write a Basic C# Program with Input/Output.	04
5.	Write a program to calculate grade based on marks using if, else, and switch.	04
6.	Write a function using ref, out, and params parameters.	04
7.	Create a Book struct and a Library class to demonstrate both types.	04
8.	Write a program to overload method.	04
9.	Write a program for static class and partial class.	04
10.	Write a program for static property and indexer.	04
11.	Write a program to implement inheritance and interface.	04
12.	Write a program to overloading operator.	04
13.	Write a program to add controls like Label, TextBox, Button to accept and display user input	04
14.	Write a program for delegate.	04
15.	Write a program creating files & directories & display the following attribute-1] Name 2] Size 3] Get creation time by using windows application.	04

Course outcomes - Students should be able to

1. Describe the features of C# and .NET framework components.
2. Write, compile, and execute basic C# programs using appropriate syntax and control structures.
3. Apply object-oriented principles to design and implement real-world software solutions.
4. Create simple GUI-based applications using Windows Forms.
5. Manage data using files, streams, and basic serialization.

References

1. Guruswamy, B. "Programming in C#". New Delhi: McGraw-Hill Education, 2012.
2. Troelsen, A., and Japikse, P. "Pro C# 8 with .NET Core 3". New York: Apress, 2020.
3. Perkins, B., Hammer, J. V., and Reid, J. D. "Beginning C# and .NET". Indianapolis: Wrox, 2015.
4. Skeet, J. "C# in Depth, 4th Edition". Shelter Island: Manning Publications, 2019.
5. Stellman, A., and Greene, J. "Head First C#, 4th Edition". Sebastopol: O'Reilly Media, 2020.

B. Sc. Part III, Semester V Computer Science

BCSP 358: Practicals based on BCST 352 Programming Essentials in Python

(Total Credit – 02)

Course Objectives: Student will be able to

1. Understand the basics of Python programming language and its syntax.
2. Develop problem-solving skills using Python.
3. Use control flow statements, functions, and data structures effectively.
4. Apply file handling and exception handling techniques.

Practical no.	Content	No. of hours per practical
1.	Write a Python program to demonstrate the use of variables, data types, and type casting.	04
2.	Write a program to accept user input and display the result using different operators.	04
3.	Implement a Python script to check if a number is even or odd using if-else.	04
4.	Create a program to display the Fibonacci series using loops.	04
5.	Demonstrate list, tuple, set, and dictionary operations.	04
6.	Write a Python function to calculate factorial using recursion.	04
7.	Create a module with multiple functions and import it in another script.	04
8.	Write a program to handle division by zero using exception handling.	04
9.	Develop a program to read and write data to a text file.	04
10.	Write a Python script to read data from a CSV file and print it.	04
11.	Create a class 'Student' with attributes and methods to display student details.	04
12.	Demonstrate inheritance by creating a base class and a derived class.	04
13.	Implement a stack and queue using list and perform basic operations.	04
14.	Write a generator function to yield even numbers up to a limit.	04
15.	Write a program using regular expressions to validate email addresses.	04

Course outcomes - Students should be able to

1. Describe the features and applications of Python in various domains.
2. Write Python programs using variables, operators, input/output, and control structures.
3. Implement functions and use standard libraries to build modular programs.
4. Work with Python data structures such as lists, tuples, dictionaries, and sets.
5. Perform file I/O operations and handle exceptions gracefully.

References

1. Zelle, J. M. "Python Programming: An Introduction to Computer Science, 3rd Edition". Franklin, Beedle & Associates, 2016.
2. Downey, A. "Think Python: How to Think Like a Computer Scientist, 2nd Edition". Needham, MA: Green Tea Press, 2015.
3. Severance, C. "Python for Everybody: Exploring Data Using Python 3". Charleston: CreateSpace Independent Publishing, 2016.
4. Lutz, M. "Learning Python, 5th Edition". Sebastopol: O'Reilly Media, 2013.
5. Ramalho, L. "Fluent Python: Clear, Concise, and Effective Programming". Sebastopol: O'Reilly Media, 2015.

B. Sc. Part III, Semester V Computer Science

BCSP 359: Practicals based on BCST 353 Introduction to mobile application development (Total Credits – 01)

Course Objectives: Student will be able to

1. To introduce students to mobile application development using cross-platform frameworks.
2. To provide foundational knowledge of mobile UI/UX design, app architecture, and state management.
3. To enable students to develop simple, functional mobile apps with modern tools.
4. To expose students to testing and deployment practices on emulators and real devices.

Practical no.	Content (Any eight of the following)	No. of hours per practical
1.	Set up the development environment and create a simple mobile app that displays "Hello World".	04
2.	Create a form with buttons and text fields for user input.	04
3.	Design a responsive layout using Linear, Grid, and Relative (or Flex in Kotlin) layouts.	04
4.	Display a list of items with images using List View or Flat List.	04
5.	Implement navigation between two or more screens/pages.	04
6.	Create multiple fragments/pages and switch between them dynamically.	04
7.	Implement a counter app to demonstrate state changes.	04
8.	Store and retrieve simple user data like username or settings locally.	04
9.	Call a public API and display results (e.g., news headlines or weather info).	04
10.	Test the app on both emulator and physical device, build APK/IPA.	04

Course outcomes - Students should be able to

1. Understand the basics of mobile platforms, development environments, and frameworks.
2. Design and implement user interfaces for mobile applications.
3. Handle user inputs, events, and state management effectively.
4. Interact with device features and external APIs for enhanced app functionality.
5. Test and deploy basic mobile apps using standard tools and practices.

References

1. Napoli, M.L. “Beginning Flutter: A Hands-On Guide to App Development”. Hoboken: Wiley, 2020.
2. Phillips, B., Stewart, C., Hardy, K., & Marsicano, B. “Android Programming: The Big Nerd Ranch Guide (4th ed.)”. Atlanta: Big Nerd Ranch Guides, 2019.
3. Gregoire, J.F. “Learning Android Application Programming for the Kindle Fire: A Hands-On Guide to Building Your First Android Application”. Boston: Addison-Wesley, 2014.
4. Mednieks, Z., Dornin, L., Meike, G., & Nakamura, M. “Programming Android: Java Programming for the New Generation of Mobile Devices”. Sebastopol: O’Reilly Media, 2012.
5. Darcey, L., & Conder, S. “Android Wireless Application Development (3rd ed., Volume I and II)”. Boston: Addison-Wesley, 2012.

B. Sc. Part III, Semester V Computer Science

BCSP 359: Practicals based on BCST 354 IOT

(Total Credits – 01)

Course Objectives: Student will be able to

1. Understand the fundamentals and architecture of the Internet of Things (IoT).
2. Explore IoT communication models, protocols, and data flow.
3. Learn about various sensors, actuators, and microcontrollers used in IoT.
4. Gain hands-on experience in interfacing hardware with software using platforms like Arduino and Raspberry Pi.

Practical no.	Content (Any eight of the following)	No. of hours per practical
1.	Getting started with Node MCU, Arduino with ESP8266 and ESP32 in the Arduino IDE.	04
2.	GPIO Interfacing and programming	04
3.	Digital on/off sensor (PIR and IR) Interfacing programming	04
4.	Analog sensor programming and uploading sensor data on cloud	04
5.	Controlling devices remotely using Bluetooth link, WiFi link	04
6.	Interfacing and programming of actuators, controlling devices remotely using cloud	04
7.	Web based device control.	04
8.	Development of Android applications suitable for IoT	04
9.	Experiments on Agriculture IoT (Soil moisture, PH monitor) IoT based home automation.	04

Course outcomes - Students should be able to

1. Explain the concept, architecture, and key technologies of IoT systems.
2. Identify and select appropriate sensors, actuators, and communication protocols for a given IoT application.
3. Interface microcontrollers (Arduino/Raspberry Pi) with sensors and devices.
4. Develop basic programs to collect, process, and transmit sensor data.
5. Implement cloud-based data logging and remote monitoring using IoT.

References

1. Dubey, R. An Introduction to Internet of Things: Connecting Devices, Edge Gateway, and Cloud with

Applications. New Delhi: Cengage India, 2020.

2. Kamal, R. Internet of Things: Architecture and Design Principles. New Delhi: McGraw Hill Education, 2017.
3. Madiseti, V., & Bahga, A. Internet of Things: A Hands-on Approach. Mumbai: Orient BlackSwan, 2015.
4. McEwen, A., & Cassimally, H. Designing the Internet of Things. Chichester: Wiley, 2013.
5. Kanetkar, Y. 21 Internet of Things Experiments. Kindle Edition: BPB Publications, 2019.

B. Sc. Part III, Semester V Computer Science
BCSP 359: Practicals based on BCST 355 Advanced Java Programming
(Total Credits – 01)

Course Objectives: Student will be able to

1. Understand the principles of object-oriented programming in Java.
2. Gain expertise in building Java applications using advanced features such as multithreading, exception handling, and file I/O.
3. Learn to create GUI-based applications using JavaFX or Swing.
4. Explore networking capabilities in Java for building client-server applications.
5. Work with Java Database Connectivity (JDBC) to access and manipulate databases.

Practical no.	Content (Any eight of the following)	No. of hours per practical
1.	Write a Program to perform Swing.	04
2.	Write a Program to implement on AWT.	04
3.	Write a Program on Database Connection.	04
4.	Write a Program to implement cookie and Session	04
5.	Write a Program on Servlet JDBC	04
6.	Write a program on Simple application using JSP.	04
7.	Write a java program to implement the SQL login ID commands using JDBC.	04
8.	Write a program to demonstrate the concept of SQL exception, SQL warning.	04
9.	Write a program to create a servlet to read the parameters	04
10.	Write a java program to implement the List.	04
11.	Write a java program to implement the SQL	04
12.	Commands using JDBC.	04
13.	Write a program to illustrate the use of JDBC connection.	04

Course outcomes - Students should be able to

1. Demonstrate proficiency in object-oriented programming using Java.
2. Apply exception handling, file streams, and multithreading concepts in Java applications.
3. Design and implement GUI-based desktop applications using JavaFX or Swing.
4. Develop networked applications using Java sockets and networking APIs.

5. Connect and interact with relational databases using JDBC.

References

1. Patel, M., & Patel, C. Advanced Java Programming. Ahmedabad: Charotar Publishing House, 2015.
2. Naughton, P., & Schildt, H. The Complete Reference: Java. New York: McGraw Hill Education, 2019.
3. Horstmann, C.S., & Cornell, G. Core Java Volume I – Fundamentals (11th Edition). Boston: Pearson Education, 2018.
4. Hall, M. Core Servlets and JavaServer Pages (Volume 1). Upper Saddle River: Prentice Hall, 2004.
5. Bai, X., & Perrone, M.P. Advanced Java Programming. Boston: Cambridge University Press, 2002.

B. Sc. Part III, Semester V Computer Science
BCSP 359: Practicals based on BCST 356 Multimedia Computing
(Total Credits – 01)

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

1. Understand the fundamental concepts of multimedia systems and components.
2. Learn about various multimedia data types such as text, image, audio, video, and animation.
3. Explore multimedia compression techniques and file formats.
4. Gain skills in multimedia authoring and presentation tools.
5. Understand multimedia communication standards and protocols.

Practical no.	Content (Any eight of the following)	No. of hours per practical
1.	Creating a Simple Multimedia Presentation Using Authoring Tools (e.g., PowerPoint, Adobe Animate).	04
2.	Demonstration of Multimedia Applications in Education, Entertainment, and Web.	04
3.	Text Compression Using Huffman and Run-Length Encoding Algorithms.	04
4.	Identifying and Converting Image File Formats: JPEG, PNG, GIF.	04
5.	Image Filtering and Enhancement Using OpenCV.	04
6.	Analyzing Audio Waveforms and Digital Sound Properties.	04
7.	Demonstration of Sampling and Quantization Effects on Audio Quality.	04
8.	Storing and Retrieving Multimedia Content from a Database	04
9.	Content-Based Image Retrieval Using Feature Matching (Color, Texture)	04

Course outcomes - Students should be able to

1. Explain the key concepts and components of multimedia systems.
2. Differentiate between various multimedia data types and their representation.
3. Apply basic compression techniques for images, audio, and video.
4. Utilize multimedia tools to create presentations and interactive content.
5. Demonstrate knowledge of multimedia file formats, standards, and transmission protocols.

References

1. Gonzalez, R.C., & Woods, R.E. Digital Image Processing. Boston: Pearson Education, 2008.
2. Reddy, S., & Reddy, R. Multimedia Computing, Communications, and Applications. New Delhi: Pearson Education, 2004.

3. Li, Z., & Lo, W.H. Multimedia Systems and Applications. Berlin: Springer, 2008.
4. Foley, J.D., & van Dam, A. Fundamentals of Interactive Computer Graphics. Boston: Addison-Wesley, 1990.
5. Perry, W.D., & Perry, K.E. Multimedia Communications: Applications, Networks, Protocols, and Standards. Upper Saddle River: Pearson Education, 2001.

B. Sc. Part III, Semester V Computer Science

BCSPVSC III: Advanced Computer Skills III

(Total Credits – 02)

(Practical Based Course: Open CV for AI solutions)

Course Objectives: Student will be able to

1. Understand the Fundamentals of Computer Vision and AI.
2. Learn to use OpenCV effectively for tasks such as image manipulation, object detection, feature tracking, and face recognition.
3. Implement AI-based Solutions for Real-World Applications.
4. Learn how to optimize OpenCV applications for speed and efficiency, focusing on real-time performance and deployment in resource-constrained environments.

Practical no.	Content (Any 15 of the following)	No. of hours per Practical
1.	Understanding the fundamentals of computer vision, Introduction and installation of the OpenCV library.	04
2.	Reading an image in OpenCV using Python.	04
3.	Load an image and use OpenCV's imwrite() function.	04
4.	Saving an Image using opencv.	04
5.	Program to define Detect and draw edges.	04
6.	Program to implement Arithmetic Operations on Images using OpenCV.	04
7.	Program to implement Bitwise Operations on Binary Images.	04
8.	Program to implement image in painting.	04
9.	Program to implement Eye, smile detection with pretrained XML models.	04
10.	Program to implement Grayscale Conversion.	04
11.	Program to Resizing an Image.	04
12.	Program to implement Rotating an Image.	04
13.	Program to define Detect geometric shapes (circle, triangle, square).	04
14.	Program to define Detect and decode QR codes or barcodes.	04
15.	Program to define Image blending using addWeighted().	04
16.	Program to Flip an image horizontally or vertically.	04

Course outcomes - Students should be able to

1. Understand and explain the core principles of computer vision and image processing using OpenCV.
2. Apply OpenCV libraries to perform real-time image and video processing tasks for AI-driven applications.
3. Implement feature detection, object tracking, and face recognition using classical and AI-based computer vision techniques.
4. Design and develop practical AI applications such as gesture recognition, smart surveillance, or automated POS systems using OpenCV.

References

1. Solem, J.E. “Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images”. Sebastopol: O'Reilly Media, 2012.
2. Vaishya, A. “Mastering OpenCV with Python: Use NumPy, Scikit, TensorFlow, and Matplotlib to Learn Advanced Algorithms for Machine Learning through a Set of Practical Projects”. New Delhi: Orange Education Pvt. Ltd., 2023.
3. Géron, A. “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”. Sebastopol: O'Reilly Media, 2019.

B. Sc. Part III, Semester V Computer Science

BCSPVSC IV: Advanced Computer Skills IV

(Total Credits – 02)

(Practical Based Course: Open-Source Operating System)

Course Objectives: Student will be able to

1. Understand the core concepts and principles of open-source operating systems (OS).
2. Learn about the architecture and components of popular open-source operating systems, such as Linux and BSD.
3. Install, configure, and manage open-source operating systems in a virtualized or real hardware environment.
4. Explore the underlying source code of open-source OS and gain a deeper understanding of how OS functions.

Practical no.	Content (Any 15 of the following)	No. of hours per Practical
1.	Navigating the File System Using Basic Linux Commands Commands: pwd, ls, cd, clear.	04
2.	Creating, Renaming, Copying, and Deleting Files and Directories Commands: mkdir, touch, cp, mv, rm, rmdir.	04
3.	Viewing and Editing File Contents Using Command-Line Tools Commands: cat, more, less, head, tail, nano, vi.	04
4.	Working with File Permissions and Ownership Commands: ls -l, chmod, chown, chgrp.	04
5.	Finding Files and Directories Commands: find, locate, which, whereis.	04
6.	Basic Networking Commands and Connectivity Checks Commands: ifconfig, ip, ping, netstat, traceroute, hostname.	04
7.	Working with Environment Variables and Aliases Commands: echo, export, set, alias, env.	04
8.	Writing a Shell Script to Print “Hello, World!” Objective: Understand script structure and execution basics.	04
9.	Creating a Script to Display the Current Date, Time, and Username Commands: date, whoami, echo.	04
10.	Shell Script to Add, Subtract, Multiply, and Divide Two Numbers Objective: Use variables and arithmetic operations.	04
11.	Shell Script to Check Whether a Number is Even or Odd Concept: Use conditional statements (if, else) and arithmetic.	04

12.	Shell Script to Check Whether a File Exists and Its Type Commands: -f, -d, -e in if conditions.	04
13.	Shell Script to Display a List of Files in the Current Directory Command: ls inside a script.	04
14.	Shell Script to Count the Number of Files and Directories in a Folder Commands: ls, wc, find, grep.	04
15.	Shell Script to Take User Input and Display a Message Command: read with echo.	04
16.	Shell Script using Loops: for, while, and until Loops	04

Course outcomes - Students should be able to

1. Explain the key features and benefits of open-source operating systems.
2. Compare different open-source operating systems (e.g., Linux, FreeBSD) in terms of architecture, functionality, and licensing.
3. Install and configure an open-source OS, setting up the environment for specific use cases.
4. Navigate and customize open-source OS at both user and administrative levels.

Reference Books

1. Tanenbaum, A. S., and Woodhull, A. S. "Operating Systems: Design and Implementation, 3rd Edition". Upper Saddle River, NJ: Prentice Hall, 2006.
2. Wheeler, D. A. "The Art of Unix Programming". Upper Saddle River, NJ: Addison-Wesley, 2004.
3. Negus, C. "Linux Bible, 9th Edition". Indianapolis: Wiley, 2019.
4. Sobell, M. G. "A Practical Guide to Linux Commands, Editors, and Shell Programming, 4th Edition". Upper Saddle River, NJ: Prentice Hall, 2013.
5. Vaughan-Nichols, S. J. "The Linux Command Line: A Complete Introduction". San Francisco: No Starch Press, 2012.

B. Sc. Part III, Semester VI Computer Science

BCST 361: C# Programming -II

(Total Credits – 02)

Course Objectives: Students should be able to...

1. To Streamline data-centric applications with C# extended features and the Entity Framework
2. To Integrate Microsoft Core with .NET Framework applications for high-performance data access.
3. To analyze the various stages in the processing of web forms and different types of controls.
4. To implement and deploy the website.

UNIT	Content	No. of hours per unit
I	Exception Handling and Threading	08
	1.1 Try, catch, throw, finally, Nested try, Custom exception 1.2 What is threading?, Applications with Multiple, Threads, Thread Priorities, Synchronization	
II	File I/O and Streams	07
	2.1 Working with Drives, Directories, and Files, The Drive Info Class, The Directory and Directory Info, Classes 1.2 File and FileInfo, Working with Paths, File and Directory Properties 1.3 Attributes, and Access, Control, Lists, Reading and Writing Files, Streams, Readers and Writers, Compressing Streams.	
III	ASP.NET	07
	3.1 Building a Web Application, Examples Using Standard Controls, Using HTML Controls 3.2 Validating form Input Controls using Validation Controls 3.3 Understanding Applications and State, Applying Styles, Themes, and Skins, Creating a Layout Using Master Pages 3.4 Binding to Databases using Controls, Creating a Site Navigation Hierarchy, Navigation Controls , Membership and Role Management 3.5 Login Controls, Securing Applications Caching For Performance, Working with XML, Using Crystal Reports in Web Forms	
IV	Database Management with ADO.net	08

	4.1 Databases: Introduction, Data Management with ADO.net, Using SQL to work with database, 4.2 retrieving and manipulating data with SQL, working with ADO.NET, ADO.NET architecture, 4.3 Introduction to MVC architecture, deploying the web site.	
--	--	--

Course outcomes - Students should be able to

1. Understand the useful and advanced concepts in C# like multithreading, error handling, reflection etc.
2. Understand the new features that are unique to C# such as properties, indexers, delegates, events and namespaces.
3. Learn the features of ASP.NET version 2.0 and the various stages in the processing of web forms and different types of controls such as server controls, web controls, HTML controls, validation controls, user control and data binding controls
4. Understand how ADO.NET is used in web development using ASP.NET and the concept of files and how database connection is established.

References

1. Robinson, S., Nagel, C., Watson, K., Glynn, J., Skinner, M., & Evjen, B. Professional C#. Indianapolis: Wrox Publications, 2008.
2. Archer, T. Inside C#. Redmond: Microsoft Press, 2001.
3. Liberty, J., Maharry, D., & Hurwitz, D. Programming ASP.NET 3.5. Sebastopol: O'Reilly Media, 2008.
4. Blackburn, P.D. ADO.NET Examples and Best Practices for C# Programmers. Berkeley: Apress, 2006.

B. Sc. Part III, Semester VI Computer Science

BCST 362: Introduction to Artificial Intelligence (AI)

(Total Credits – 02)

Course Objectives: Students should be able to...

1. To create appreciation and understanding of both the achievements of AI and the theory underlying those achievements.
2. To introduce the concepts of a Rational Intelligent Agent and the different types of Agents that can be designed to solve problems.
3. To review the different stages of development of the AI field from human like behavior to Rational Agents.
4. To impart basic proficiency in representing difficult real-life problems in a state space representation so as to solve them using AI techniques like searching and game playing.

UNIT	Content	No. of hours per unit
I	Introduction to AI	08
	1.1 Introduction to AI. 1.2 A brief history of AI and how it's used in daily life (e.g., voice assistants, smart devices). 1.3 Understanding the idea of intelligence in machines. 1.4 Introduction to machine “intelligent”. 1.5 The Turing Test: Can machines think like humans. Introduction to smart systems (Agents).	
II	Problem Solving with AI	07
	2.1 Understanding problems AI can help solve. 2.2 Simple ways to represent and solve problems (like puzzles and games) 2.3 Search Techniques: Breadth-First Search (BFS), Depth-First Search (DFS), Introduction to smarter searching using ideas and guesses (Heuristics), Introduction to Game Playing – How AI plays games like Tic-Tac-Toe.	
III	Internet Security	07
	3.1 Introduction to First Order Predicate Logic. 3.2 Resolution Principle, Unification, Semantic Nets. 3.3 Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. 3.4 Programming in Logic (PROLOG).	

IV	Dealing with Uncertainty and Inconsistencies	08
	4.1 Truth Maintenance System, Default Reasoning. 4.2 Probabilistic Reasoning, Bayesian Probabilistic Inference. 4.3 Possible World Representations. 4.4 Understanding Natural Languages: Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.	

Course outcomes - Students should be able to

1. Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
2. Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game-based techniques to solve them.
3. Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing.
4. Attain the capability to represent various real life problem domains using logic-based techniques and use this to perform inference or planning.

References

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell & Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd edition, 2005.
3. Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2nd edition, 1991.
4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.
5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd edition, 2000.

B. Sc. Part III, Semester VI Computer Science

BCST 363: Advanced Mobile Application Development

(Total Credits – 02)

Course Objectives: Students should be able to...

1. Understand advanced features and architecture of mobile platforms.
2. Design and develop secure, scalable, and user-friendly mobile applications.
3. Integrate cloud services and real-time databases.
4. Optimize mobile apps for performance and user experience.

UNIT	Content	No. of hours per unit
I	Advanced Mobile Architecture	08
	1.1 Architecture Patterns (MVC, MVP, MVVM) 1.2 Data Binding and Lifecycle Awareness 1.3 Injection (e.g., Hilt/Dagger, SwiftDI) 1.4 State Management in Flutter (Provider/Bloc)	
II	Data Storage and Persistence	07
	2.1 Local Databases (Room, SQLite, Core Data) 2.2 File Storage and Shared Preferences 2.3 Cloud Integration: Firebase Firestore/Realtime DB 2.4 Offline Data Syncing	
III	Cross-Platform Mobile Development	07
	3.1 Introduction to Flutter/React Native 3.2 Platform Channels & Native Modules 3.3 UI Development and Navigation 3.4 Performance Considerations in Cross-Platform Apps	
IV	Device Features & Sensor Integration	08
	4.1 Camera, GPS, Accelerometer, Gyroscope 4.2 Push Notifications and Firebase Cloud Messaging 4.3 Background Services and Job Scheduling 4.4 Accessing Contacts, Call, SMS APIs	

Course outcomes - Students should be able to

1. Design and develop advanced mobile applications using modern development frameworks (e.g., Flutter, React Native, Kotlin, or Swift).
2. Integrate device features such as GPS, camera, sensors, and notifications into mobile apps.
3. Implement advanced UI/UX designs following platform-specific design guidelines and best practices.
4. Work with local and remote data by integrating databases (e.g., SQLite, Firebase) and web services (RESTful APIs).

References

1. Zammetti, F.A. “Flutter for Dummies”. Hoboken: Wiley, 2020.
2. Lewis, P. “Flutter Projects: A practical, project-based guide to building real-world cross-platform mobile applications and games”. Birmingham: Packt Publishing, 2020.
3. Joseph, A. “Mastering Android Development with Kotlin”. Birmingham: Packt Publishing, 2018.
4. Meier, R. “Professional Android”. Indianapolis: Wrox, 2018.
5. Albahari, J., & Albahari, B. “C# 10 in a Nutshell: The Definitive Reference”. Sebastopol: O'Reilly Media, 2022.
6. Sill, D. “Android Development Patterns: Best Practices for Professional Developers”. Indianapolis: Addison-Wesley, 2015.

B. Sc. Part III, Semester VI Computer Science

BCST 364: Web technology

(Total Credits – 02)

Course Objectives: Students should be able to...

1. Understand client server architecture.
2. Study of web applications using java technologies
3. Gain the skills and project-based experience needed for entry into web application and development careers.
4. Learn different web related technologies.

UNIT	Content	No. of hours per unit
I	Web Essentials	08
	1.1 Introduction, 1.2 Web Essentials: Clients, Servers, Communication, 1.3 Basic Internet Protocols, HTTP Request Message, HTTP Response Message, 1.4 HTTPS protocol, Web Clients, 1.5 Generations of web applications	
II	Introduction to Client-Side Programming	07
	2.1 Introduction to JavaScript, Basic Syntax, 2.2 Variables and Data Types, Statements, Operators, literals, functions. 2.3 JavaScript Objects–properties, references, methods, 2.4 constructors, Arrays, other built-in objects, Debugging JavaScript, 2.5 Introduction to Host Objects, Document Object Model (DOM), Document tree, DOM event handling, 2.6 jquery, YUI Library	
III	Server-Side Programming	07
	3.1 Java servlet: architecture, life cycle. 3.2 The Client Request – form data, request headers. 3.3 The Server Response- HTTP Status Codes, HTTP Response Headers. Sessions, 3.4 Cookies, URL Rewriting, Concurrency in servlets, 3.5 Separating Programming and Presentation: Java server pages, Basic JSP, JavaBeans Classes and JSP, JSF, 3.6 Java Database Connectivity (JDBC), PHP	
IV	Representing Web Data	08

4.1 XML–Namespaces, AJAX–Overview, 4.2 basics, toolkits, security, DOM based XML processing, XSL, XPath, XSLT, 4.3 Content Management Frameworks (Drupal, Joomla, etc.) 4.4 Web configuration security: Apache Security, Nginx Security, jBoss. 4.5 Remote Command Execution, Tomcat Remote Command Execution, HTTP Parameter Pollution.	
---	--

Course outcomes - Students should be able to

1. Understand the fundamentals of web protocols.
2. Learn different web related technologies currently used.
3. Understand Studying data handling in web systems.
4. Analysing a wide range of web security vulnerabilities and issues.

References

1. "Core Web Programming", Marty Hall, Larry Brown ,Pearson Education, 2nd Edition,2001.
2. "Programming the World Wide Web", Robert. W. Sebesta, Pearson Education, 4thEdition, 2007.
3. "Internet & World Wide Web How To Program", H.M. Deitel, P.J. Deitel and A.B. Goldberg, Pearson Education, 3rd Edition, 2006.

B. Sc. Part III, Semester VI Computer Science

BCST 365: Computer Graphics

(Total Credits – 02)

Course Objectives: Students should be able to...

1. To understand the basics of various input and output computer graphics.
2. To make the student present the content graphically.
3. To Gain knowledge about graphics hardware and software.
4. To understand Various 2D and 3D objects transformation techniques.

UNIT	Content	No. of hours per unit
I	Basic of Computer Graphics	08
	1.1 Overview of Computer Graphics. 1.2 Computer Graphics Application and Software. 1.3 Description of some graphics devices, Input Devices for Operator Interaction, 1.4 Active and Passive Graphics Devices, Display Technologies 1.5 Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays. 1.6 Raster Refresh (Raster-Scan) Graphics Displays 1.7 Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics. 1.8 The Video Controller, Random-Scan Display Processor, LCD displays.	
II	Graphics Primitive	07
	2.1 Graphics programming, initializing the graphics 2.2 Graphical functions, simple Programs Point Plotting Techniques 2.3 Qualities of good line drawing algorithms. 2.4 The Digital Differential Analyzer (DDA) Bresenham's Algorithm, Generation of Circles.	
III	Two-Dimensional and Three-Dimensional Transformations	07
	3.1 Introduction of Transformations. 3.2 Types, Transformations and Matrices. 3.3 Transformation Conventions 3.4 Homogeneous Coordinates and Matrix Representation of 2D and 3D Transformations, Translations, Rotation, Reflection, Scaling, Shearing Combined Transformation, Transformation of Points	

IV	Clipping Techniques and OpenGL	08
	4.1 Clipping, Need for Clipping 4.2 Types of Clipping, Point Clipping, Line Clipping, Text Clipping, Mid-Point Subdivision Line Clipping Algorithm 4.3 Drawing Polygon, Introduction to OpenGL, Features in OpenGL, OpenGL operations 4.4 Abstractions in OpenGL – GL, GLU & GLUT, 3D viewing pipeline, viewing matrix specifications, a few examples and demos of OpenGL programs	

Course outcomes - Students should be able to

1. Analyse the basic principles, techniques, and applications of Artificial Intelligence.
2. Implement machine learning techniques and AI computing environment that are suitable for the applications under consideration.
3. Understand and apply scaling up machine learning techniques and associated computing techniques and technologies.
4. Use different machine learning techniques to design AI machine and enveloping applications for real world problems

Reference Books:

1. Joshi, V. Machine Learning and Artificial Intelligence. Cham: Springer, 2021.
2. Harrington, P. Machine Learning in Action. Shelter Island: Manning Publications, 2012.
3. Nagy, Z. Artificial Intelligence and Machine Learning Fundamentals. Birmingham: Packt Publishing, 2018.
4. Han, J., & Kamber, M. Data Mining: Concepts and Techniques. Burlington: Morgan Kaufmann, 2011.

B. Sc. Part III, Semester VI Computer Science

BCST 366: E – Commerce

(Total Credits – 02)

Course Objectives: Students should be able to...

1. To design web pages using html.
2. To Develop familiarity with the JavaScript language.
3. To develop, implement and creating Applications with C#.
4. To develop, implement, and demonstrate Component Services, Threading. in.

UNIT	Content	No. of hours per unit
I	An introduction to electronic commerce	08
	1.1 What is E-Commerce (Introduction and Definition) 1.2 Main activities E-Commerce, Goals of E-Commerce. 1.3 Technical Components of E-Commerce. 1.4 Functions of E-Commerce. 1.5 Commerce, Scope of E-Commerce, Electronic 1.6 Commerce Applications, Electronic Commerce 1.7 Electronic Business(C2C) (C2G, G2G, B2G, B2P, B2A, P2P, B2A, C2A, B2B, B2C).	
II	Internet Security	07
	2.1 Introduction of servlet. 2.2 Life cycle of servlet, Session. 2.3 Cookies, servlet jdbc connection. 2.4 Components of JSP: Directives, Tags, Scripting elements, simp Secure Transaction, Computer Monitoring . 2.5 Privacy on Internet, Corporate Email privacy, Computer. 2.6 Crime(Laws , Types of Crimes), Threats, Attack on Computer System, Software Packages for privacy, Hacking, Computer Virus(How it spreads, Virus problem, virus protection. 2.7 Encryption and Decryption. 2.8 Secret key Cryptography, DESPublic Key Encryption, RSA, Authorization and Authentication. 2.9 Firewall, Digital Signature(How it works)	
III	Electronic Data Exchange	07

	3.1 Introduction. 3.2 Concepts of EDI and Limitation. 3.3 Applications of EDI, Disadvantages of EDI, EDI Model. 3.4 Electronic Payment System: Introduction, Types of Electronic Payment System, Payment Types, Value Exchange System, Credit Card System. 3.5 Electronic Fund Transfer, Paperless bill, Modern Payment Cash, Electronic Cash.	
	Planning for Electronic Commerce	08
IV	4.1 Planning Electronic Commerce initiatives. 4.2 Linking objectives to business strategies, Measuring cost. 4.3 objectives, Comparing benefits to Costs, Strategies for Developing electronic commerce web site. 4.4 Internet Marketing: The PROS and CONS of online shopping, The cons of online shopping 4.5 Justify an Internet business, Internet marketing techniques. 4.6 The E-cycle of Internet marketing, Personalization e commerce.	

Course outcomes - Students should be able to

1. Understand the complexity of e-commerce and its many facts.
2. Explore how e-business and e-commerce fit together.
3. Recognize the benefits and limitations of e-commerce.
4. Identify the main barriers to the growth.

References

1. Laudon, K.C., & Traver, C.G. E-Commerce 2023: Business, Technology, Society (18th ed.). London: Pearson, 2023.
2. Chaffey, D. E-Business and E-Commerce Management: Strategy, Implementation and Practice (6th ed.). London: Pearson Education, 2015.
3. Belew, S., & Elad, J. Starting an Online Business All-in-One For Dummies (6th ed.). Hoboken: Wiley, 2021.

Bachelor of Science (B. Sc.) Part III

BCSP 367: Lab Course (Practical Based on BCST 361) C# Programming II

(Total Credits – 02)

Course Objectives: Student will be able to

1. Understand advanced features of the C# programming language, including exception handling, generics, collections, and file operations.
2. Develop Windows-based and console-based applications using object-oriented principles.
3. Explore data handling using file I/O and database integration with ADO.NET or Entity Framework.
4. Implement multithreaded applications and understand thread synchronization.

Practical no.	Content	No. of hours per unit
1.	Write a program for thread.	04
2.	Perform basic file operations using File and FileInfo.	04
3.	Write a program for Reading/Writing file by using byte stream class.	04
4.	Write a program that handles divide-by-zero and null reference exceptions.	04
5.	Write a program for copy one file to another file.	04
6.	Write a program to list all files and folders in a directory.	04
7.	Simple application using web controls Finding factorial Value Money Conversion Quadratic Equation Temperature Conversion Login control	04
8.	Write a program to display messages in a calendar control.	04
9.	Write a program to display vacation in a calendar control.	04
10.	Write a program to make selected day in a calendar control using style.	04
11.	Build a form with TextBox, DropDownList, and Button to collect user data.	04
12.	Write a program to demonstrate treeview control and datalist.	04
13.	Write a program to demonstrate treeview operations.	04
14.	Write a program to use of Validation controls.	04

15.	Implement login/logout using Session and Cookies without login control.	04
-----	---	----

Course outcomes - Students should be able to

1. Apply exception handling mechanisms to create robust and error-resistant applications.
2. Use generics and collection classes to manage data structures efficiently.
3. Perform file read/write operations using streams and file classes in the System.IO namespace.
4. Develop applications with multithreading capabilities to handle concurrent processing.

References

1. Price, M. C# 10.0 and .NET 6 – Modern Cross-Platform Development (6th ed.). Birmingham: Packt Publishing, 2021.
2. Troelsen, A., & Japikse, P. Pro C# 9 with .NET 5 (10th ed.). New York: Apress, 2021.
3. Griffiths, I. Programming C# 8.0: Build Cloud, Web, and Desktop Applications (1st ed.). Sebastopol: O'Reilly Media, 2019.
4. Skeet, J. C# in Depth (4th ed.). Shelter Island: Manning Publications, 2019.

Bachelor of Science (B. Sc.) Part III

BCSP 368: Lab Course (Practical Based on BCST 362) Introduction to Artificial Intelligence (AI)

(Total Credits – 02)

Course Objectives: Student will be able to

1. Introduce the foundational concepts and scope of Artificial Intelligence, including its history, goals, and applications.
2. Familiarize students with key AI techniques such as search algorithms, knowledge representation, and reasoning.
3. Explore problem-solving methods using state-space search, heuristics, and constraint satisfaction.
4. Understand the basics of machine learning, including supervised and unsupervised learning.

Practical no.	Content	No. of hours per unit
1.	Introduction of Artificial Intelligence and its application.	
2.	Write a Program to Implement Depth First Search using Python.	
3.	Write a Program to Implement Breadth First Search using Python.	
4.	Write a Program to Implement Tower of Hanoi using Python.	
5.	Write a Program to Implement Tic-Tac-Toe game using Python.	
6.	WAP to implement search problem of 3 x 3 puzzle.	
7.	Write a Program to implement 8-Puzzle problem using Python.	
8.	Write a Program to Implement Water-Jug problem using Python.	
9.	Write a Program to Implement Travelling Salesman Problem using Python.	
10.	Write a program to implement heuristic search procedure.	
11.	Write a prolog program to calculate the sum of two numbers.	
12.	Write a prolog program to find the maximum of two numbers.	
13.	Write a prolog program to calculate the factorial of a given number.	
14.	Write a Prolog program to implement append for two lists.	

15.	Write a Prolog program to count even and odd elements from list and count elements up to specific index in list.	
-----	--	--

Course outcomes - Students should be able to

1. Describe the core principles of AI and identify its key areas and real-world applications.
2. Apply uninformed and informed search strategies to solve AI problems.
3. Represent knowledge using logic-based and rule-based systems for reasoning and inference.
4. Implement basic machine learning algorithms such as decision trees and k-means clustering.

References

1. Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach (4th ed.). Boston: Pearson, 2020.
2. Rich, E., Knight, K., & Nair, S.B. Artificial Intelligence (3rd ed.). New Delhi: McGraw-Hill Education, 2017.
3. Poole, D., & Mackworth, A. Artificial Intelligence: Foundations of Computational Agents (2nd ed.). Cambridge: Cambridge University Press, 2017.
4. Luger, G.F. Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th ed.). Boston: Pearson, 2008.

B. Sc. Part III, Semester VI Computer Science

BCSP 369: Practical's based on BCST 363 Advanced Mobile Application Development (Total Credit – 01)

Course Objectives: Student will be able to

1. Understand the fundamentals of mobile platforms including Android and iOS.
2. Differentiate between native and cross-platform mobile application development.
3. Learn the tools and technologies used in mobile development environments such as IDEs, SDKs, and emulators.
4. Develop mobile applications using a chosen framework (e.g., Flutter or React Native).

Practical no.	Content (Any eight of the following)	No. of hours per practical
1.	Implementing MVVM Architecture in a Kotlin Mobile Application.	04
2.	Using Provider for State Management in a Multi-Screen Kotlin App.	04
3.	Building Custom Widgets and Responsive UI Layouts.	04
4.	Integrating RESTful APIs and Displaying Data using Future Builder.	04
5.	Implementing Local Database Storage with SQLite in Kotlin.	04
6.	Firebase Integration: Authentication and Cloud Fire store.	04
7.	Using Sensors (Accelerometer, Gyroscope) and Device Features.	04
8.	Background Services and Notifications in Mobile Apps.	04
9.	Platform-Specific Code Execution (Kotlin Platform Channels).	04
10.	Creating Animations and Transitions for Improved UX.	04

Course outcomes - Students should be able to

1. Describe the architecture and components of Android and iOS mobile platforms.
2. Compare and contrast native and cross-platform development strategies.
3. Install, configure, and utilize development environments for mobile application creation.
4. Build mobile user interfaces using buttons, text fields, images, lists, and layouts.
5. Use fragments to create modular and dynamic user experiences.

Reference Books

1. Napoli, M.L. “Beginning Flutter: A Hands-On Guide to App Development”. Hoboken: Wiley, 2020.
2. Phillips, B., Stewart, C., Hardy, K., & Marsicano, B. “Android Programming: The Big Nerd Ranch Guide” (4th ed.). Atlanta: Big Nerd Ranch Guides, 2019.

3. Mednieks, Z., Dornin, L., Meike, G., & Nakamura, M. "Programming Android: Java Programming for the New Generation of Mobile Devices". Sebastopol: O'Reilly Media, 2012.
4. Darcey, L., & Conder, S. "Android Wireless Application Development" (3rd ed., Volume I & II). Boston: Addison-Wesley, 2012.
5. Gregoire, J.F. "Learning Android Application Programming for the Kindle Fire: A Hands-On Guide to Building Your First Android Application". Boston: Addison-Wesley, 2014.

B. Sc. Part III, Semester VI Computer Science

BCST 369: Practical's based on BCST 364 Web technology

(Total Credits – 01)

Course Objectives: Student will be able to

1. Understand the fundamentals of the internet and web architecture.
2. Learn HTML, CSS, and JavaScript for front-end web development.
3. Gain practical knowledge of client-server architecture and HTTP protocols.
4. Develop interactive and dynamic web pages using scripting and frameworks.

Practical no.	Content (Any eight of the following)	No. of hours per practical
1.	Create a college website using HTML tags.	04
2.	Create a calculator format using java script.	04
3.	Add a simple script using Click event.	04
4.	Create Employee details using schemas.	04
5.	Change the color of the old image to the new image and apply filter effects.	04
6.	Draw an image in flash and work with different layers.	04
7.	Creating a banner – Typography.	04
8.	Animation with different layers with adding sounds.	04
9.	Create an Image Gallery with Lightbox Effects using JavaScript	04

Course outcomes - Students should be able to

1. Design and build static and dynamic web pages using HTML, CSS, and JavaScript.
2. Implement client-side validation and interactivity with JavaScript.
3. Create responsive layouts using modern CSS frameworks (e.g., Bootstrap).
4. Develop web applications using front-end and back-end technologies.

Reference Books

1. Duckett, J. "HTML and CSS: Design and Build Websites". Indianapolis: Wiley, 2011.
2. Duckett, J. "JavaScript and JQuery: Interactive Front-End Web Development". Indianapolis: Wiley, 2014.
3. Robbins, J.N. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics". Sebastopol: O'Reilly Media, 2018.
4. Flanagan, D. "JavaScript: The Definitive Guide". Sebastopol: O'Reilly Media, 2020.
5. Ullman, L. "PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide". San Francisco: Peachpit Press, 2017.

B. Sc. Part III, Semester VI Computer Science
BCST 369: Practical's based on BCST 365 Computer Graphics
(Total Credits – 01)

Course Objectives: Student will be able to

1. Understand the fundamental principles and mathematical foundations of computer graphics.
2. Learn algorithms for 2D and 3D graphics transformations and projections.
3. Explore techniques for modeling, rendering, and animation.
4. Gain knowledge of graphic programming using libraries like OpenGL or WebGL.

Practical No.	Content (Any eight of the following)	No. of hours per practical
1.	Introduction to various graphics commands.	
2.	Develop the DDA Line drawing algorithm.	
3.	Draw basic graphics construction	
4.	Program to define Functions in Graphics.	
5.	Translation, Rotation, and Scaling using Composite Transformation	
6.	Implement Standard Perspective Projection in 3-Dimensions.	
7.	To implement Point Clipping.	
8.	To implement OpenGL programming.	
9.	Draw a rectangle, triangle, and ellipse using graphics primitives.	

Course outcomes - Students should be able to

1. Apply mathematical and geometric concepts to develop 2D and 3D graphics.
2. Implement basic algorithms for line drawing, clipping, and filling.
3. Perform transformations such as translation, rotation, scaling, and projections.
4. Use graphics libraries (e.g., OpenGL) to create visual content and animations.
5. Design and develop simple interactive graphics applications.

Reference Books:

1. Hearn, D., & Baker, M.P. "Computer Graphics with OpenGL". Boston: Pearson, 2014.
2. Foley, J.D., van Dam, A., Feiner, S.K., & Hughes, J.F. "Computer Graphics: Principles and Practice". Boston: Addison-Wesley, 2013.
3. Hill, F.S., & Kelley, S.M. "Computer Graphics Using OpenGL". Upper Saddle River: Prentice Hall, 2006.
4. Angel, E., & Shreiner, D. "Interactive Computer Graphics: A Top-Down Approach with WebGL". Boston: Pearson, 2020.
5. Marschner, S., & Shirley, P. "Fundamentals of Computer Graphics". Boca Raton: CRC Press, 2021.

B. Sc. Part III, Semester VI Computer Science

BCST 369: Practical's based on BCST 366 E- Commerce

(Total Credits – 01)

Course Objectives: Student will be able to

1. Understand the fundamental concepts and types of E-Commerce models.
2. Explore the infrastructure and technologies enabling E-Commerce.
3. Analyze legal, ethical, and security issues in online business.
4. Learn about electronic payment systems and digital currencies.

Section	Content (Any eight of the following)	No. of hours per unit
1.	Write a HTML program to design a form which should allow you to enter your personal data. (Hint: make use of text field, password field, e-mail, lists, radio buttons, checkboxes, submit button)	04
2.	Write html code to generate the following output. • Coffee • Tea o Black Tea o Green Tea • Milk. Write a HTML code to generate following output 	04
3.	Design a CSS to create menu.	04
4.	Design a webpage i.e. Bio data using CSS.	04
5.	Write a program to create table and list using CSS.	04
6.	Design a home page using javascript.	04
7.	Create a web page and display data from database.	04
8.	Create a simple XML document to display address book.	04
9.	Design a simple homepage using HTML and CSS with sections like product listings, navigation bar, and contact info.	04

Course outcomes - Students should be able to

1. Identify technical components required for E-Commerce infrastructure.
2. Evaluate E-Commerce platforms, hosting, and payment gateway integration.
3. Assess security measures, encryption techniques, and legal frameworks in E-Commerce.
4. Design and analyze E-Commerce websites or strategies using real-world tools and platforms.

Reference Books:

1. Laudon, K.C., & Traver, C.G. "E-Commerce 2021: Business, Technology, Society". Boston: Pearson, 2021.
2. Schneider, G. "Electronic Commerce". Boston: Cengage Learning, 2017.
3. Chaffey, D. "Digital Business and E-Commerce Management". Harlow: Pearson Education, 2019.
4. Turban, E., Outland, J., King, D., Lee, J.K., Liang, T.P., & Turban, D.C. "Electronic Commerce: A Managerial and Social Networks Perspective". Cham: Springer, 2018.
5. Rayport, J.F., & Jaworski, B.J. "Introduction to E-Commerce". New York: McGraw-Hill, 2003.

B. Sc. Part III, Semester VI Computer Science

BCSPVSC V: Lab Course (Practical Based on VSC)

(Total Credits – 02)

Course Objectives: Student will be able to

1. Understand the Basics of R Language.
2. Handle different data types, structures (vectors, matrices, data frames, lists), and indexing.
3. Perform Data Manipulation and Analysis.
4. Apply Statistical Methods.

Practical no.	Content (Any 15 of the following)	No. of hours per practical
1.	Learn all the basics of R-Programming (Data types, Variables, Operators).	04
2.	Using R execute the Vector, Array and list.	04
3.	Using R execute the Matrix computation.	04
4.	Implement data frames in R. Write a program to join columns and rows in a data frame using cbind() and rbind() in R.	04
5.	Implement decision making statements with different examples.	04
6.	Implement R-Loops with different examples.	04
7.	Write a program to read a csv file and analyse the data in the file in R.	04
8.	Data visualization using R.	04
9.	To find Measure of central tendency-I using R.	04
10.	To find Measure of central tendency-II using R.	04
11.	To find Measure of dispersion using R -I.	04
12.	To find skewness and kurtosis using R.	04
13.	To find Correlation using R.	04
14.	To find Regression using R.	04
15.	To find probabilities using R.	04
16.	Testing of hypothesis using R.	04

Course outcomes - Students should be able to

1. Show the installation of R Programming Environment.
2. Utilize and R Data types for developing programs.
3. Apply R programming for reading, cleaning, visualizing of data.
4. Apply R Programming for Statistical Analysis

References

1. Wickham, H., & Grolemund, G. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. Sebastopol: O'Reilly Media, 2017.
2. Matloff, N. The Art of R Programming: A Tour of Statistical Software Design. San Francisco: No Starch Press, 2011.
3. Grolemund, G. Hands-On Programming with R: Write Your Own Functions and Simulations. Sebastopol: O'Reilly Media, 2014.
4. McElreath, R. Statistical Rethinking: A Bayesian Course with Examples in R and Stan (2nd ed.). Boca Raton: CRC Press, 2020.