SYLLABUS FOR BCA (MAJOR)

SEM. V&VI

Under Single Major Single Minor (FYUGP)

(To be implemented from Session 2024-25)

Proposed Syllabus for four years BCA (Major) Programme							
Year	**Semester**	**Paper Code**	**Paper**	**Credits**	**Periods/Week**	**Exam. Marks**	**Total Marks**
3rd Year	**V**	BCAPMAJ509	Operating System	3	3	60	80
		BCAPMAJ3509L	Operating System (Lab)	1	2	20	
		BCAPMAJ510	Software Engineering	3	3	60	80
		BCAPMAJ510T	Software Engineering (Tutorial)	1	1	20	
		BCAPMAJ511	Computer Networks	3	3	60	80
		BCAPMAJ511T	Computer Networks (Tutorial)	1	1	20	
		BCAPMAJ512	Website Design with HTML and PHP	3	3	60	80
		BCAPMAJ512L	Website Design with HTML and PHP (Lab)	1	2	20	
	VI	BCAPMAJ613	Database Management System	3	3	60	80
		BCAPMAJ613L	Database Management System (Lab)	1	1	20	
		BCAPMAJ614	Design and Analysis of Algorithm	3	3	60	80
		BCAPMAJ614T	Design and Analysis of Algorithm (Tutorial)	1	1	20	
		BCAPMAJ615	Python Programming	3	3	60	80
		BCAPMAJ615L	Python Programming (Lab)	1	2	20	
		BCAPMAJ616	Introduction to Internet of Things using Arduino Sensors	3	3	60	80
		BCAPMAJ616L	Introduction to Internet of Things using Arduino Sensors (Lab)	1	2	20	

NOTE:Tutorials should involve problem solving session/activity related to the subject taught.

3rd Year
Semester-V

Course-MAJOR Paper:	**Paper Code-BCAPMAJ509 Operating System**	**Credits-3**	**Lectures/Week-3**

Prerequisite(s) and/or Note(s):

1. Basic understanding of computer organization and architecture
2. Familiarity with data structures and fundamental programming concepts
3. Note(s): This course builds foundational knowledge essential for advanced topics in system-level programming and distributed computing.

Course Objectives:

Knowledge Acquired:

1. Understanding the fundamental principles and functions of modern operating systems
2. Ability to analyse and solve common problems related to process management, memory, storage, and device control
3. Appreciation of the role operating systems play in bridging computer hardware and user-level applications
4. Foundational knowledge that supports further studies in areas such as systems programming, distributed systems, and cloud computing

Skills Gained:

1. Proficiency in writing and understanding system-level code
2. Ability to manage memory, schedule processes, and understand file systems and I/O operations
3. Skills in debugging and performance analysis of operating system components
4. Enhanced problem-solving abilities relevant to concurrent programming and resource management

Competency Developed:

1. Ability to design, implement, and optimize operating system components
2. Understanding of security, access control, and basic networking from an OS perspective
3. Competency in solving real-world system-level problems involving concurrency, deadlocks, and scheduling
4. Readiness for roles in software development, system programming, system administration, cybersecurity, and research in OS and distributed systems

Syllabus Overview

Unit 1:	**Introduction**	**8 Lectures**

Basic OS functions, resource abstraction, types of operating systems, multi-programming systems, batch systems, time sharing systems, multiprocessing system, operating systems for personal computers & workstations, process control & real time systems.

Unit 2:	**Operating System Organization**	**5 Lectures**

Processor and user modes, kernels, system calls and system programs

Unit 3:	Process Management	12 Lectures

System view of the process and resources, process abstraction, process hierarchy, process life cycle, threads, threading issues, thread libraries, thread life cycle, Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms

Unit 4:	ConcurrencyandSynchronization	10 Lectures

Process synchronization, criticalsection, semaphores, methods for inter-process communication. Deadlocks: System model, deadlock characterization, deadlock prevention, detection and avoidance, recovery from deadlock, banker's algorithm.

Unit 5:	Memory Management	10 Lectures

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, demand paging, page replacement algorithms, allocationoffframes, thrashing, segmentation, and virtual memory.

Suggested Readings

1. A.Silberschatz,P.B.Galvin,G.Gagne,OperatingSystemsConcepts,8thEditio n, John Wiley Publications2008.
2. A.S.Tanenbaum,ModernOperatingSystems, 3rdEdition,PearsonEducation2007.
3. G.Nutt,Operating Systems:AModernPerspective,2ndEditionPearsonEducation1997.
4. W.Stallings, OperatingSystems,Internals&DesignPrinciples, 5thEdition, PrenticeHallof India.2008.
5. M.Milenkovic,OperatingSystems-Conceptsanddesign, TataMcGrawHill1992.
6. OperatingSystems,A.KSharma, UniversityPress.

Course-MAJOR Paper:	Paper Code-BCAPMAJ509L Operating System(Lab)	Credits-1	Lab hours/Week-2

Studentsareadvisedto dolaboratory/practicalpractice notlimitedto,but includingthefollowing types of problems:

1. WAPinCfor implementationofPriorityscheduling.
2. WAPinCfor implementationofpreemptiveandnon-preemptivescheduling.
3. WAPinCfor implementationofRoundRobinscheduling.
4. WAPinCfor implementationofFCFSscheduling.
5. WAPinCfor implementationofSJFscheduling.
6. WAPinCforimplementationofproducer–consumerproblemusingsemaphores.
7. WAPinCforimplementationofBanker'salgorithmfordeadlockavoidance.
8. WAPinCforimplementationofdeadlockavoidance.
9. WAPinCforimplementationofmemoryallocationmethodsforfixedpartitioning(Firstfit)
10. WAP in C to create a child process using fork (), display parent and child process id. Child process shoulddisplaythemessage"Childprocess"andtheparent process shoulddisplaythe message "Parent process".
11. WAP inC toaccept nintegers tobesorted. Mainfunctioncreates childprocess usingfork systemcall. Parentprocesssortstheintegersinascendingorder andwaitsfor childprocess using wait system call. Child process sorts the integers in descending order.
12. WAP inCtoimplementtheprocesssystemcalls.

Course-MAJOR Paper:	Paper Code-BCAPMAJ510 Software Engineering	Credits-3	Lectures/Week-3

Brief Course Description:

This course provides a comprehensive introduction to the principles, methodologies, and practices ofsoftware engineering. Students will learn the entire software development lifecycle, from requirementsgathering to design, implementation, testing, deployment, and maintenance.

Prerequisite(s) and/or Note(s):

1. Basic understanding of programming concepts
2. Basic familiarity with software development principles and terminology
3. **Note(s):**Course content may be revised during the term to align with academic priorities.Students will be evaluated only on the topics covered in class.

Course Objectives:

Knowledge Acquired:

Upon completion of the course, students will:

1. Understand and apply software engineering principles.
2. Effectively manage the Software Development Life Cycle (SDLC).
3. Contribute to the development of high-quality software systems.
4. Be prepared for roles in software development, project management, and related fields.
5. Gain knowledge in key software engineering areas, including:
    - SDLC phases
    - System design principles
    - Software testing techniques
    - Software configuration management

Skills Gained:

1. Application of SDLC models in real-world projects
2. Designing structured and modular software systems
3. Performing software testing and quality assurance
4. Managing software versions and configurations
5. Documenting and communicating software requirements and specifications

Competency Developed:

1. Strong programming skills relevant to software development
2. Enhanced problem-solving abilities
3. Application of algorithmic thinking to software design
4. Ability to participate in and contribute to software development projects
5. Working knowledge of software maintenance and evolution
6. Understanding the importance of team collaboration and project management in software engineering

Syllabus Overview

Unit 1: Introduction 4 Lectures

Definition of software and software engineering, the Evolving Role of Software, Software Characteristics,Softwarequality,ChangingNatureofSoftware,Softwareprocess.

Unit 2: Software Development Models 6 Lectures

Software Development Life Cycle (SDLC), Types of Software Models- Waterfall Model, Iterative Model,SpiralModel,PrototypeModel.CapabilityMaturityModel Integration(CMMI).

Unit 3: SoftwareRequirementAnalysisandPlanning 10 Lectures

Software Requirement Analysis and Modeling Techniques, Software Requirement Specification (SRS)-NeedforSRS,CharacteristicsandComponentsofSRS,Factfindingtechniques,CostEstimation-COCOMOModel,DataFlowDiagram(DFD)andEntity-RelationshipDiagram (ERD),

Unit 4: SoftwareProjectManagement-SPM 8 Lectures

Define SPM, Needs of SPM, Software Risks, Risk Identification, RMMM Plan, Software Quality ControlandQualityAssurance,ISO(InternationalStandardsOrganization)9000Certification,Softwar eMetrics.

Unit 5: Coding 4 Lectures

Define Coding, Goals of Coding, characteristics of Programming Language, Programming style, CodingStandards,CodingGuidelines.

Unit 6: TestingStrategies&Tactics 13 Lectures

Software Testing Fundamentals-Test Plan, Test Execution, Test Review, Inspection and Auditing, Testingprinciples, software Verification and validation, Types of software testing - Black-Box Testing, White-BoxTesting and their type, Other testing- Regression Testing, Mutation Testing, Acceptance testing, Alphatesting,Betatesting.

Suggested Readings

1. R.S.Pressman,SoftwareEngineering: APractitioner'sApproach(7thEdition),McGraw-Hill, 2009.
2. P.Jalote,AnIntegrated ApproachtoSoftwareEngineering(2ndEdition),NarosaPublishingHouse,2003.
3. K. K. AggarwalandY. Singh, SoftwareEngineering(2ndEdition),NewAgeInternationalPublishers,2008.
4. I.SommerVille, SoftwareEngineering(8thedition),AddisonWesley, 2006.
5. D.Bell, SoftwareEngineeringforStudents(4thEdition), Addison-Wesley, 2005.R.Mall,FundamentalsofSoftwareEngineering(2ndEdition),Prentice-HallofIndia,2004

Course-MAJOR Paper: Paper Code-BCAPMAJ510T Credits-1 Tutorial hours/Week-2
Software Engineering (Tutorial)

Software Engineering tutorial as assigned and advised by teacher(s).

Course- MAJOR Paper:	Paper Code-BCAPMAJ511 Computer Networks	Credits-2	Lectures/Week-2

Prerequisite(s) and/or Note(s):

1. Basic Computer Literacy: Students should be familiar with fundamental computer operations
such as using the keyboard, mouse, files, folders, and basic software applications.
2. No Prior Networking Knowledge Required: This is an introductory course, designed for beginners.
3. Recommended for: Undergraduate students from Computer Science, IT, Electronics, or any discipline with interest in Information Technology.

Course Objectives:

1. To introduce students to the basic concepts and purpose of computer networks.
2. To explain the different types of networks, devices, transmission media, and communication protocols.
3. To provide a foundational understanding of network architectures such as the OSI and TCP/IP
models.
4. To familiarize students with basic IP addressing, configuration of small networks, and networking tools.
5. To develop practical skills for designing, setting up, and troubleshooting basic wired and wireless networks.

Knowledge Acquired:

By the end of this course, students will have gained theoretical and practical understanding of:
1. The role and components of computer networks in modern communication.
2. Types of networks (LAN, WAN, MAN), topologies, and data transmission media.
3. Hardware components like routers, switches, hubs, access points, NICs, and their uses.
4. Networking protocols such as TCP/IP, HTTP, FTP, SMTP, DNS, and DHCP.
5. Network security fundamentals including firewall basics, encryption, and safe internet practices.

Skills Gained:

1. Identify and differentiate between types of networks, hardware, and topologies.
2. Setup, configure, and troubleshoot a basic wired or wireless network.
3. Share resources like printers and files over a local network, Configure Wi-Fi routers and manage basic internet sharing settings.
4. Implement safe browsing and basic network security practices.

Competency Developed:

1. Understanding the foundational structure and operation of computer networks.
2. Performing hands-on tasks involved in configuring small network environments.
3. Communicating effectively using networking terminology and concepts.
4. Applying practical problem-solving techniques in network setup and troubleshooting.
5. Enhancing employability for roles such as IT support technician, helpdesk executive, or junior
network administrator.

Syllabus Overview

Unit 1:	**DataCommunicationFundamentalsandTechniques**	**10 Lectures**

Introduction:DataCommunications,Networks,Categoriesofnetwork-LAN, MAN, WAN,ModesofCommunication-Simplex,HalfDuplex,FullDuplex,Networktopologies,InternetHistory,LayeredNetwork Architecture-OSI model andTCP/IP protocol suite. Data and Signals- Analog and Digital signals,ParallelandSerialtransmission,Transmissionimpairment-Attenuation,DistortionandNoise,Multiplexingtechniques- FDM,TDM, WDM,Transmission media - Guided: Twisted Pair, Coaxial Cable, Fiber Optic, Unguided -Wi-Fi, Bluetooth, Infrared, Satellite.

Unit 2:	**NetworksSwitchingTechniquesandAccessmechanisms**	**5 Lectures**

Switching-Circuitswitching,Packetswitching;Messageswitching,ConnectionlessandConnection-orienteddatagramswitching.

Unit 3:	**DataLinkLayerFunctionsandProtocol**	**7 Lectures**

Errordetectionanderrorcorrectiontechnique-HammingCode,CRC,Checksum.Flowcontrolmechanism-SlidingWindowprotocol, go-back-nARQ,stopandwaitARQ

Unit 4:	**MultipleAccessProtocolandNetworks**	**7 Lectures**

Multiple Access Protocols: Radom Access - ALOHA, CSMA, CSMA/CD, CSMA/CA, Controlled Access -Reservation, Polling, Token Passing, Channelization - FDMA, TDMA, CDMA. Network devices- repeaters,hubs,switches, bridges,router andgateways;

Unit 5:	**NetworksLayer& Transport Layer FunctionsandProtocols**	**10 Lectures**

Routing- Routingalgorithms,classesofIP, IPv4andlPv6 Addresses, IP Addressing schemes, subnetting (basic concepts);Transport services-Congestion control, Connection establishment and release- three- way handshaking,NetworkSecurity-Cryptography

Unit 6:	**Overview ofApplicationlayerprotocol**	**6 Lectures**

OverviewofDNSprotocol,overviewofWWW, Important networking protocols: HTTP/HTTPS, FTP, SMTP, POP3, DNS and DHCP; Understanding firewalls and antivirus for network protection.

Suggested Readings

1. Behrouz A. Forouzan, “Data Communications and Networking”, 5th Edition, McGraw HillEducation,2013.
2. AndrewS.Tanenbaum,DavidJ.Wetherall,“ComputerNetworks”,5thEdition,PrenticeHall,2011.
3. Larry L. Peterson and Bruce S. Davie, “Computer Networks A System Approach”, 5th Edition,MKP,2012.
4. James F. Kurose, Keith W. Ross, “Computer Networking, A Top-Down Approach”, 5thEdition,Pearson,2012.
5. Dr.RakeshKumarMandal:Computer NetworksforStudents, FirstEdition,SPD,2018

Course-MAJOR Paper:	**Paper Code-BCAPMAJ511T** **Computer Networks (Tutorial)**	**Credits-1**	**Lab hours/Week-2**

Computer Networks tutorial as assigned and advised by teacher(s).

Course-MAJOR	Paper Code-BCAPMAJ512	Credits-3	Lectures/Week-3
Paper:	Website Design with HTML and PHP		

Brief Course Description:

The Website Design with HTML and PHP course introduces students to the fundamentals of web development. The course focuses on designing web pages using HTML and creating dynamic and interactive websites using PHP. Students will learn how front-end and back-end technologies work together to develop modern web applications. Through theory and practical exercises, students will gain the skills required to design, develop, and maintain professional websites.

Prerequisite(s) and/or Note(s):

(1) Basic knowledge of computer operations and internet usage.
(2) Prior programming experience is helpful but not mandatory.
(3) Note(s): Syllabus may be modified during the academic session depending on academic requirements and technological developments. Students will be evaluated only on the topics covered during the course.

Course Objectives:

The primary objective of this course is to provide students with a strong foundation in website development using HTML and PHP. Students will learn how to create structured web pages, design interactive user interfaces, develop dynamic web applications, and manage user data through web forms and databases. The course aims to prepare students for further studies and practical work in web technologies.

Knowledge Acquired:

1. HTML Fundamentals: Students acquire knowledge of HTML elements, attributes, page structure, hyperlinks, tables, forms, and multimedia integration.
2. PHP Programming Concepts: They learn PHP syntax, variables, operators, control structures, functions, arrays, and file handling.
3. Dynamic Web Development: Students understand how HTML and PHP work together to create dynamic web pages, process user input, and interact with databases.
4. Web Deployment Concepts: They gain knowledge of web hosting, domain names, server environments, and website deployment.

Skills Gained:

1. Web Page Design: Students develop the ability to design and create well-structured web pages using HTML.
2. Dynamic Website Development: They learn to develop interactive websites using PHP and server-side scripting techniques.
3. Form Handling and Data Management: Students gain practical skills in creating web forms, validating user input, and managing data through databases.
4. Debugging and Testing: They learn to identify, troubleshoot, and resolve errors in web applications effectively.

Competency Developed:

1. Website Development Competency: Students develop the ability to design, develop, and maintain complete websites using HTML and PHP.
2. Problem-Solving Ability: They learn to apply programming concepts and logical thinking to solve web development challenges.
3. Professional Web Development Skills: Students gain the competency to create responsive,

interactive, and user-friendly web applications that meet modern web standards.

4. Industry Readiness: The course prepares students for entry-level web development roles, freelance projects, and further specialization in advanced web technologies.

Syllabus Overview

Unit 1: Introduction to Programming **5 Lectures**

Introduction to Markup Languages and HTML, need and use; the Head, the Body, Colors, Attributes, Lists, orderedand unordered

Unit 2: Links,ImagesandTables **5 Lectures**

Introduction; Relative Links, Absolute Links; Link Attributes; Using the ID Attribute to Link Within a Document;Putting an Image on a Page, Using Images as Links, Putting an Image in the Background, Creating a Table, TableHeaders,Captions, SpanningMultipleColumns, StylingTable

Unit 3: IntroductiontoXML **5 Lectures**

IntroductiontoXMLanditsGoals,XMLStructureandSyntax,DocumentclassesandRules,Scripti ngXML,XMLasData, LinkingwithXML,XSL–Style SheetBasics,XSLbasics,XSLstyle sheets. (3L)

Unit 4: IntroductiontoPHP **5 Lectures**

PHP introduction, inventionsandversions,scope, importanttoolsandsoftwarerequirements(likeWeb Server,Database, Editors etc.), Basic Syntax, PHP variables and constants, Types of data in PHP, Expressions, scopes of avariable (local, global), PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwise, ternary andMODoperator, PHP operator Precedenceandassociatively.

Unit 5: HandlingHTMLformwithPHP **5 Lectures**

BasicInputandAttributes,OtherKindsofInputs,StylingformswithCSS,WheretoGofromHereCapturingF ormData,GET and POST formmethods,Dealingwithmulti value fields,Redirectingaformaftersubmission.

Unit 6: PHPconditionalevents,Loops and PHP Functions **10 Lectures**

PHPIFElseconditionalstatements(NestedIFandElse),Switchcase,while, ForandDoWhileLoops,Goto,Break, Continueand exit.Function,NeedofFunction,declarationandcallingofafunction,PHPFunctionwitharguments,Default ArgumentsinFunction, Functionargumentwithcallbyvalue, call byreference, Scopeof FunctionGlobalandLocal

Unit 7: StringManipulation, RegularExpression and Arrays **10 Lectures**

Creating and accessing String, Searching & Replacing String; Formatting, joining and splitting String , String RelatedLibrary functions; Use and advantage of regular expression over inbuilt function; Use of preg_match(), preg_replace(),preg_split()functionsinregularexpression AnatomyofanArray, CreatingindexbasedandAssociativearray, Accessingarray;LoopingwithIndexbasedarray,withassociative arrayusingeach()and foreach();Some useful Libraryfunction.

Suggested Readings:

1. VirginiaDeBolt , IntegratedHTMLand CSSASmarter,FasterWaytoLearn ,Wiley/Sybex, 2006
2. CassidyWilliams,CamrynWilliams IntroductiontoHTMLandCSS,O'Reilly,2015
3. XMLin actionweb technologybyWilliamJ. Pardi
4. Step byStep XMLbyMichaelJ.Young
5. StevenHolzner,"PHP:TheCompleteReferencePaperback",McGrawHill Education(India),2007.
6. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", Wiley India PrivateLimited,2008.
7. RobinNixon,"LearningPHP,MySQL,JavaScript,CSS&HTML5",3rdEditionPaperback,O'reilly,2014.

Course-MAJOR	**Paper Code- COMSMAJ512L**	**Credits-1**	**Lab hours/Week-2**
Paper:	**Website Design with HTML and PHP (Lab)**		

Studentsareadvisedto dolaboratory/practicalpractice notlimitedto,but includingthefollowing types of problems:

1. Create an HTML document with the different formatting options i.e. Bold, Italics, Underline, Headings (UsingH1toH6headingstyles),Font(Type,SizeandColor),Background(Coloredbackground/ Imageinbackground),Paragraph, Line Break, HorizontalRule,Pretag.
2. Create an HTML document which consists of: Ordered List, Unordered List, Nested ordered and/or unorderedListand Image
3. CreateanHTMLdocument whichimplementsInternallinkingaswellasExternallinking.
4. Create a table using HTML demonstrating use of columns and rows, merging multiple rows and/or columnsetc.withdata andimage values andcontents with hyperlinking
5. Createatypicalstudent datacaptureformforthepurposeofadmissiontoyourcollegeusingdifferenttypesofHTMLcontrolsi.e.Text Box, Option/radiobuttons, Checkboxes,Resetand Submitbuttonsetc.
6. CreateHTMLdocumentshavingmultipleframesindifferentpossibleformats/organization
7. InformationStructure:Inthisexercise,studentwillpracticeidentifyingthestructureofaninformationobject
8. DeconstructinganXMLDocument:Inthisexercise,studentwillpracticeidentifyingtheexplicitstructurewithinanXML document. Ina sense, thisisthereverse ofwhatyoudidin Exercise #7.
9. Creating XML Markup: In this exercise, create some XML markup based on the tree representation frompreviousexercise and the contentfromthe originalsample document.
10. CreateaPHPpageusingfunctionsforcomparingthreeintegersandprint thelargest number.
11. Write a function to calculate the factorial of a number (non-negative integer). The function accept the numberas anargument.
12. WAPtocheckwhetherthegivennumber isprimeornot.
13. Create a PHP page which accepts string from user. After submission that page displays the reverse ofprovidedstring.
14. Writea PHPfunctionthat checks ifastringis alllowercase
15. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase,orsequence thatreadsthesamebackward as forward,e.g.,madamornurses run)
16. WAPto sort anarray.
17. Write a PHP script that removes the whitespaces from astring. Sample string: 'The quick " " brown fox'ExpectedOutput :The quick brown fox
18. Writea PHPscriptthatfindsoutthesumoffirstnoddnumbers.

19. Create a login page having user name and password. On clicking submit, a welcome message should bedisplayed if the user is already registered (i.e.name is present in the database) otherwise error message shouldbe displayed.
20. WriteaPHPscriptthat checksifastringcontainsanother string.
21. Create a simple 'birthday countdown' script, the script will count the number of days between current day andbirth day.
22. Createascripttoconstructthefollowingpattern,usingnestedforloop.

    *
    * *
    * * *

3rd Year
Semester-VI

Course-MAJOR Paper:	**Paper Code-COMSMAJ613 Database Management System**	**Credits-3**	**Lectures/Week-3**

Brief Course Description:

This course provides a comprehensive introduction to the principles, design, and implementation ofDatabaseManagementSystems(DBMS).Studentswilllearnthefundamentalconceptsofdatabaseorganization,datamodeling,andSQL(StructuredQueryLanguage).Thecourseemphasizesboththeoretical understanding and practical skills necessary for effective database design, implementation,andmanagement.

Prerequisite(s) and/or Note(s):

1. Basic understanding of:
    a. Computer systems
    b. Data structures and Databases
    c. Programming concepts
2. **Note(s):**The syllabus is subject to minor modifications depending on institutional or academic requirements.Students will be assessed only on the basis of topics covered in class during the semester.

Course Objectives

Knowledge Acquired:

Upon completion of the course, students will:

1. Understand the principles of relational database design.
2. Be able to implement and manage relational databases.
3. Gain practical knowledge of SQL for data querying and manipulation.
4. Learn about database optimization techniques.
5. Be prepared for roles in:
    - Database administration
    - Database development
    - Data-driven application development

Skills Gained:

1. Database design and schema creation
2. Proficiency in SQL for querying, updating, and managing data
3. Data modeling using Entity-Relationship (ER) diagrams
4. Query optimization for efficient data retrieval

5. Application of normalization techniques to database structures
6. Awareness of database security and access control measures

Competency Developed:

1. Database connectivity through APIs and tools
2. Implementing backup and recovery solutions
3. Performing database administration tasks
4. Understanding the basics of data warehousing
5. Introduction to data mining and knowledge discovery
6. Fundamentals of distributed databases and their management

Syllabus Overview

Unit 1: Introduction 8 Lectures

Definition of DBMS, file processing system Vs DBMS, Limitation of file processing system, databaseusers, Characteristics of database,database components,database system architecture- 3-layerand 2-layers, data independence, data dictionary. View of Data – Data Abstraction – Instances and Schemasdiagrams.DBA-DatabaseAdministrator-rolesandresponsibilitiesof aDBA

Unit 2: Database Models 7 Lectures

A brief overview of three models- Hierarchical model, Network model and relationalmodel,comparisonofthree models.

Unit 3: Database Design, ER- Diagrams 12 Lectures

ER-Model,Constraints,ER-Diagrams,different types of entities, attributes, relationships,E. F. Codd'srules,Keys-Primary,Candidate,Composite, Alternate, Foreign, Super, Relational Schemas, Normalization: Lossless decomposition, Lossydecomposition,Functionaldependencies,NormalForms- (Up toBCNF-1NF,2NF,3NF,BCNF).

Unit 4: Transaction Processing 6 Lectures

ACIDproperties,Concurrencycontrol.

Unit 5: SQLandORACLE 12 Lectures

Oracle commands, Oracle datatypes, operators- Arithmetic, Logical, Range searching, Pattern Matching,Oracle functions- Aggregate, Numeric, String, Conversion, Date and Time, Oracle table-Dual, Groupingdata-Concept of grouping, Group by Clause, Having clause Language-DDL, DML, DCL, SQL Functions-queriesandsub-queriesinSQL, ConstrainsandindexesinSQL.

Suggested Readings:

1. R.Elmasri,S.B.Navathe,FundamentalsofDatabaseSystems6thEdition, PearsonEducation, 2010.
2. R.Ramakrishanan, J.Gehrke, DatabaseManagementSystems3rdEdition, McGraw-Hill,2002.
3. A.Silberschatz,H.F. Korth,S.Sudarshan, DatabaseSystemConcepts6thEdition, McGrawHill,2010.
4. R.Elmasri,S.B.NavatheDatabaseSystemsModels,Languages,DesignandapplicationProgrammi ng,6thEdition,PearsonEducation,2013.
5. IvanBayross, Teach YourselfSQL &PL/SQL UsingORACLE8i& 9iwithSQLJ,1st Edition, BPBPublications,2003

Course-MAJOR Paper:	Paper Code-BCAPMAJ613L Database Management System (Lab)	Credits-1	Lab hours/Week-2

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems.

1. Demonstratetablecreation
2. INSERTdataintothetable
3. SELECTtherecordfromthetable
4. SELECTallrowsandallcolumnsformthetable
5. Demonstratefilteringtabledata–Selectedcolumns andallrows,selectedrows andallcolumns,selectedcolumnsandselectedrows,Renametable.
6. SELECTallrecordsfromthetableeliminating duplicaterows.
7. Filteringtabledata
8. Sortingdatainatable
9. Creatingatablefromaanothertable
10. Insertingdataintoatablefromanothertable.
11. Demonstratedeleteoperations
12. Updatingthecontents ofatable
13. Modifyingthestructureof table
14. Destroyingtables
15. Findoutthetablecreatedbydifferentusers
16. Demonstratedifferenttypesof dataconstraints
17. DemonstratedifferentoperatorsinSQL-Arithmetic, logical,
18. Demonstratepatternmatching,rangesearching
19. Demonstrateoracletable'DUAL'

Demonstratedifferent oraclefunctions.

Course-MAJOR Paper:	Paper Code-BCAPMAJ614 Design and Analysis of Algorithm	Credits-3	Lectures/Week-3

Brief Course Description:

Design and Analysis of Algorithms is a core subject in computer science that focuses on developing efficient procedures for solving computational problems and assessing their performance. It involves crafting algorithms using techniques like divide and conquer, dynamic programming, and greedy methods, and evaluating their efficiency through time and space complexity using Big O notation. The subject also includes optimizing algorithms for better performance and exploring complexity classes, such as P and NP, to understand computational limits. Mastery of these concepts is essential for creating effective and efficient software and systems.

Prerequisite(s) and/or Note(s):

Students interested in this course should have had anexposure to introductory courses on programming and data structures. A basic of mathematical logic is essential for success in this course**.**

Course Objectives:

The objective of learning Design and Analysis of Algorithms is to develop efficient methods for solving computational problems and to evaluate their performance using time and space complexity. It aims to equip students with skills to optimize algorithms for better speed and resource usage while understanding problem complexity and classification. This foundational knowledge is crucial for advancing in computer science and creating effective software solutions.

Knowledge acquired:

1. Understanding how to measure and improve the efficiency of algorithms.
2. Proficiency in analyzing and comparing time and space complexity.
3. Skills to enhance algorithms for better performance and lower resource consumption.
4. Mastery of various algorithmic techniques such as divide and conquer, dynamic programming, and greedy methods.

Skills gained:

1. Ability to conceptualize and develop step-by-step procedures for solving complex problems.
2. Proficiency in analyzing the time and space complexity of algorithms to gauge their efficiency.
3. Skills in refining algorithms to enhance their performance and minimize resource usage.
4. Mastery of various Problem-Solving Techniques, such as divide and conquer, dynamic programming, and greedy strategies.
5. Insight into the classification of problems and algorithms based on complexity classes like P, NP,etc.

CompetencyDeveloped:

1. Ability to evaluate and compare the efficiency of different algorithms using time and space complexity metrics.
2. Proficiency in refining and optimizing algorithms to improve performance and resource utilization.
3. Expertise in applying various algorithmic strategies and techniques to effectively solve diverse computational problems.
4. Enhanced capability to analyze and reason about the computational limits and trade-offs of different algorithmic approaches.

Syllabus Overview

Unit 1:	**Introduction**	**6 Lectures**

Definitions, Algorithms vs Programs, Basic AlgorithmDesignTechniques,CorrectnessofAlgorithm,Algorithm classification: Iterativetechniques,DivideandConquer,Dynamic Programming,GreedyAlgorithms, Algorithm analysis (formal and informal), calculation of running time of an algorithm, loop invariants.

Unit 2:	**Growth of Functions**	**7 Lectures**

Complexity of Algorithms, Best, Worst and Average case complexity, growth rate of functions, Asymptotic Analysis, Analyzing algorithm control structures.

Unit 3:	**Recurrences**	**8 Lectures**

Introduction, Substitution method, Iteration method, Master method (Master theorem), Simple problems using Master theorem.

Unit 4:	**Analysis of simple SortingandSearchingalgorithms**	**16 Lectures**

Elementary sorting techniques: Bubble Sort, Selection sort, Insertion Sort, Shell sort, Merge Sort, Advanced Sorting techniques - Heap Sort,QuickSort,SortinginLinearTime-BucketSort,RadixSortandCountSort,SearchingTechniques.

Unit 5:	**Graphs**	**8 Lectures**

GraphAlgorithms–BreadthFirst Search,DepthFirst SearchanditsApplications, Directed acyclic graphs, Single source shortest paths: Dijkstra's algorithm, MinimumSpanningTrees algorithms: Prim's algorithm, Kruskal's Algorithm.

Suggested Readings

1. T.H.Cormen,CharlesE.Leiserson,RonaldL.Rivest,CliffordSteinIntroductionto Algorithms,PHI,3rdEdition 2009
2. Sarabasse& A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher Pearson 3rdEdition 1999
3. RajeshK.Shukla,AnalysisandDesignofAlgorithms:ABegineersApproach,Wiley
4. Udit Agarwal, Algorithms design and analysis, Dhanpat rai & Co.

Course-SEC Paper:	**Paper Code-BCAPMAJ614T**	**Credits-1**	**Lab hours/Week-2**
	Design and Analysis of Algorithms (Tutorial)		

Design and Analysis of Algorithms tutorial as assigned and advised by teacher(s).

Course-MAJOR Paper:	**Paper Code-BCAPMAJ615**	**Credits-3**	**Lectures/Week-3**
	Python Programming (Lab)		

Prerequisite(s) and/or Note(s):

(1) High school mathematics.

(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives

Knowledge Acquired:

1. Fundamental Concepts: Students acquire knowledge of fundamental programming concepts such as variables, data types, loops, conditionals, and functions in Python.
2. Data Structures: They learn about essential data structures like lists, tuples, dictionaries, and sets, understanding their usage and implementation.

Skills Gained:

1. Coding Proficiency: Through hands-on practice and assignments, students develop coding proficiency in Python, enabling them to write clear, concise, and functional code.
2. Problem-Solving: They enhance their problem-solving skills by applying Python programming concepts to solve various computational problems and algorithms.
3. Debugging and Troubleshooting: Students acquire skills in debugging code and troubleshooting errors, learning how to identify and fix common programming mistakes effectively.

Competency Developed:

1. Logical Thinking: Python programming exercises require logical thinking and algorithmic problem-solving skills, helping students develop a logical mindset.
2. Attention to Detail: Writing code necessitates attention to detail to ensure accuracy and functionality. Students develop this competency through debugging and code review processes.
3. Collaboration and Documentation: Students learn to collaborate on coding projects using

version control systems like Git and to document their code effectively, enhancing their ability to work in teams and communicate technical concepts clearly.

Syllabus Overview

Unit 1:	Introduction to Programming	6 Lectures

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. Flowcharts, algorithms, Types of programming paradigms – unstructured, procedure oriented, object oriented. Programming methodologies - top-down and bottom-up programming

Unit 2:	Introduction to Python	12 Lectures

Structure of a Python Program, Elements of Python, Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables**.**

Unit 3:	Flow control and Functions	12 Lectures

Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit(), def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling.

Unit 4:	List, Dictionary, String and Tuples	15 Lectures

String, String functions, Manipulating Strings, Lists: Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists. Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries. Tuples and Sets: Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Built-in Functions on Sets; Set Methods.

Suggested Readings

1. Yashavant Kanetkar and Aditya Kanetkar , Let Us Python,BPB,3rd Edition.
2. Sheetal Taneja and Naveen Kumar, Python Programming- A modular approach with Graphics, Database,
3. Mobile and Web applications, Pearson, Sixteenth Impression-2023,
4. T. Budd, Exploring Python, TMH, 1st Ed, 2011
5. Python Tutorial/Documentation www.python.or 2015
6. Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist: learning withPython , Freely available online.2012

Course-MAJOR Paper:	Paper Code-BCAPMAJ615L Python Programming (Lab)	Credits-1	Lab hours/Week-2

Studentsareadvisedto dolaboratory/practicalpractice notlimitedto,but includingthefollowing types of problems:

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users' choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:

Grade A: Percentage >=80
Grade B: Percentage>=70 and <80
Grade C: Percentage>=60 and <70
Grade D: Percentage>=40 and <60
Grade E: Percentage<40

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.

4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to implement the use of arrays in Python.
7. WAP to implement String Manipulation in python in Python.
8. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3! - - - - - n/n!
9. WAP to calculate the sum and product of two compatible matrices.
10. WAP to create Class and Objects in Python.
12. WAP to implement Data Hiding in Python.
13. WAP to implement constructor and destructor for a class in Python.
14. WAP to implement constructor and destructor in Python.
15. WAP to implement different types of inheritance in Python.
16. WAP to implement concept of Overriding in Python.
17. Write programs to create mathematical 3D objects using class.
    a. curve b. sphere c. cone d. arrow e. ring f. cylinder
18. WAP to Check if a Number is Positive, Negative or 0
19. WAP to Check leap year or not.
20. WAP to display calendar of the given month and year.

Course-MAJOR	**Paper Code-BCAPMAJ616**	**Credits-3**	**Lectures/Week-3**
Paper:	**Introduction to Internet of Things using Arduino Sensors**		

Prerequisite(s) and/or Note(s):

(1) Basic knowledge of computers and electronics at the high school level.
(2) Note(s): Syllabus may be modified during the academic session depending on the availability of resources and emerging technologies. Students will be evaluated only on the topics covered during the course.

Course Objectives

Knowledge Acquired:

1. Introduction to IoT: Students learn the basic concepts of the Internet of Things (IoT), its applications, and its importance in modern technology.
2. Arduino Fundamentals: They understand the Arduino platform, Arduino UNO board, Arduino IDE, and basic programming concepts.
3. Sensors and Interfacing: Students gain knowledge of different sensors such as LED modules, IR sensors, and Ultrasonic sensors, and learn how they interact with Arduino.

Skills Gained:

1. Arduino Programming: Students develop the ability to write, compile, and upload Arduino programs (sketches) using the Arduino IDE.
2. Hardware Interfacing: They learn to connect and control LEDs, LCD displays, and various sensors using Arduino boards.
3. Project Development: Students gain practical experience in building simple IoT-based applications such as distance measurement, water level monitoring, and sensor-based

automation.

Competency Developed:

1. Problem-Solving Ability: Students learn to design simple solutions using sensors and microcontrollers to address real-world problems.
2. Hands-on Technical Skills: They develop confidence in assembling electronic circuits, testing components, and troubleshooting hardware and software issues.
3. Innovation and Application Development: Students gain the ability to create basic smart systems and IoT prototypes using Arduino and sensor technologies.

Course Outcomes

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

1. Explain the basic concepts, applications, and importance of the Internet of Things (IoT).
2. Use the Arduino IDE to write, compile, and upload programs to an Arduino board.
3. Interface LEDs, LCD displays, and sensors with Arduino and control them through programming.
4. Develop simple sensor-based applications using IR and Ultrasonic sensors.
5. Design and implement basic IoT and automation projects using Arduino and electronic sensors.

Syllabus Overview

Unit 1:	**Introduction IoT and Arduino**	**3 Lectures**

IoT - The Internet of Things Today, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, IoT Related Standardization, Recommendations on Research Topics - Concept of Arduino Basics, the Arduino platform, Block Diagram, Architecture, Understanding Arduino UNO Board and Components, Installing and work with Arduino IDE.

Unit 2:	**Arduino Basic programming**	**Lectures**

Arduino Control structure, Arduino Functions, Arduino operators, Arduino Sketch Structure

Unit 3:	**Interfacing**	**Lectures**

LED with Arduino, Working of LED, Sketch for blinking LED using delay function, Sketch Explanation, Interfacing LCD display with Arduino, Interfacing different sensors with Arduino Sensor

Unit 4:	**IR Sensor**	**Lectures**

Introduction to IR Sensor, Working of IR sensor, Pinouts of IR sensor, Connection of IR sensor with Arduino, Sketch showing working of IR sensor, Sketch explanation Sensor

Unit 5:	**Ultrasonic Sensor (HC-SR04)**	**Lectures**

Introduction to HC-SR04, Working of HC-SR04, Pinouts of HC-SR04, Connection of HC-SR04 with Arduino, Sketch for water level monitoring using HC-SR04, Sketch explanation Sensor.

Suggested Readings

1. Make: Sensors, Book by Kimmo Karvinen, Tero Karvinen, and Ville Valtokari
2. Getting Started with Sensors: Measure the World with Electronics, Arduino and Raspberry Pi by

Kimmo Karvinen and Tero Karvinen .
3. Simon Monk "Programming Arduino: Getting Started with Sketches" 2nd Edition, Kindle Edition, McGraw Hill; 2nd edition
4. Michael Margolis "Arduino Cookbook, 2e", O′Reilly; 2nd edition
5. Blum Richard "Arduino Programming in 24 Hours, Sams Teach Yourself", Sams Publishing; 1st edition
6. Internet of Things, RMD Sundaram Shriram K Vasudevan, Abhishek S Nagarajan, John Wiley and Sons.
7. Internet of Things, Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, John Wiley & Sons.
8. Massimo Banzi, Michael Shiloh Make: Getting Started with the Arduino, Shroff Publisher/Maker Media Publishers.

Course-MAJOR	**Paper Code-BCAPMAJ616L**	**Credits-1**	**Lab hours/Week-2**
Paper:	**Introduction to Internet of Things using Arduino Sensors (Lab)**		

Students are advised to do laboratory/practical practice not limited to, but including the following types of programs

1. Write a program to blink an LED with an interval of one second.
2. Write a program to continuously create a fading effect in an LED using PWM (Pulse Width Modulation).
3. Write a program to turn the LED ON when the button is pressed and OFF when released.
4. Write a program to activate the buzzer when the button is pressed.
5. Write a program to read analog sensor values of LDR and display them on the serial monitor/ LCD.
6. Write a program to interface a relay module with Arduino to control the turning ON and OFF of a 220V bulb when an LDR detects darkness/brightness
7. Write a program to read the temperature values using analog sensor LM35, calculate the temperature in Celsius and Fahrenheit and display on serial monitor/ LCD.
8. Write a program to simulate a traffic light sequence (Red → Green → Yellow) using LEDs.
9. Write a program to interface a relay module with Arduino to control the turning ON and OFF of a 220V bulb when an LDR detects darkness/ brightness.
10. Write a program to interface a Bluetooth module HC-05 with Arduino and send “1”/”0” commands from the mobile phone to control an LED ON/ OFF.
11. Write a program to interface a Bluetooth module HC-05 with Arduino and send any sensor data (e.g., temperature, light intensity) from Arduino to mobile phone.
12. Write a program to interface a relay module with Arduino to control the turning ON and OFF of a 220V bulb when an LDR detects darkness/brightness.