B.Sc. (Computer Science)

PROGRAME OUTCOME	COURSE NAME	COURSE CODE	COURSE OUTCOME
❖ An ability to apply knowledge of computing and mathematics appropriate to the discipline. ❖ An ability to identify, formulates, and develops solutions to computational challenges. ❖ An ability to design, Implements, and evaluate a computational system to meet desired needs within realistic constraints. ❖ An ability to function effectively on teams to	1. Mathematics-I	BCS-101	1. Analyze real world scenarios to recognize when ordinary differential equations (ODEs) or systems of ODEs are appropriate, formulate problems about the scenarios, creatively model these scenarios (using technology, if appropriate) in order to solve the problems using multiple approaches, judge if the results are reasonable, and then interpret and clearly communicate the results.
			2. Recognize ODEs and system of ODEs concepts that are encountered in the real world, understand and be able to communicate the underlying mathematics involved to help another person gain insight into the situation.
			3. Work with ODEs and systems of ODEs in various situations and use correct mathematical terminology, notation, and symbolic processes in order to engage in work, study, and conversation on topics involving ODEs and systems of ODEs with colleagues in the field of mathematics, science or engineering.
			4. Identify a general method for constructing solutions to inhomogeneous linear constant-coefficient second-order equations
			5. show an awareness of initial and boundary conditions to obtain particular values of constants in the general solution of second-order differential equations
	2. Computer Fundamental &	BCS-102	1. Understanding the concept of input and output devices of Computers and how it works and recognize the basic

accomplish shared computing design, evaluation, or implementation goals. ❖ An understanding of professional, ethical, legal, security, and social issues and responsibilities for the computing profession. ❖ An ability to communicate and engage effectively with diverse stakeholders.	Office Automation		terminology used in computer programming.
			2. Analyze and understand in-depth training in use of office automation packages, internet etc.
			3. Enhance the ability of essential for common man for day to day office management, and e-governance.
			4. To understand what computer is? To understand the Basics of Operating systems.
			5. To evaluate how to use software packages in day to day activities.
	3. Applied Physics	BCS-103	1. Analyze the intensity variation of light due to Polarization, interference and diffraction.
			2. To develop the understanding of laws of thermodynamics and their application in various processes.
			3. To formulate and solve the engineering problems on Electromagnetism.
			4. Determine gradient, divergence and curl of scalar and vector fields.
			5. Demonstrate a working knowledge of the basic concepts and theories of physics.
	4. MS-Office Lab	BCS-102-P	1. Design and Develop a vocabulary of key terms related to the computer and to software program menus.
			2. Describe and identify the components of a personal computer system.
			3. Demonstrate mouse and keyboard functions.
			4. Analyze and Deduce window and menu commands and how they are used.

			5. Demonstrate how to organize files and documents on a USB/hard drive.
	5. Physics Lab	BCS-103-P	1. Apply the various procedures and techniques for the experiments.
			2. Use the different measuring devices and meters to record the data with precision.
			3. Apply the mathematical concepts/equations to obtain quantitative results.
			4. Develop basic communication skills through working in groups in performing the laboratory experiments and by interpreting the results.
	6. English Communication	AEC-1	1. Understand professional writing by studying management communication contexts and genres, researching contemporary business topics, analyzing quantifiable data discovered by researching, and constructing finished professional workplace documents.
			2. Recognize, explain, and use the formal elements of specific genres of organizational communication: white papers, recommendation and analytical reports, proposals, memorandums, web pages, wikis, blogs, business letters, and promotional documents.
			3. Understand how to critically analyze data from research; incorporate it into assigned writing clearly, concisely, and logically; and attribute the source with proper citation.
			4. Compare different format features in both print, multimedia and html documents, and develop document design skills.
			5. Develop professional work habits, including those necessary for effective collaboration and cooperation with other students, instructors and Service Learning contact representatives.
	7. Mathematics-II	BCS-201	1. Describe fundamental properties of the real numbers that lead

			to the formal development of real analysis.
			2. Comprehend rigorous arguments developing the theory underpinning real analysis.
			3. Demonstrate an understanding of limits and how they are used in sequences, series, differentiation and integration.
			4. Construct rigorous mathematical proofs of basic results in real analysis.
			5. Appreciate how abstract ideas and rigorous methods in mathematical analysis can be applied to important practical problems.
	8. Computer Graphics & multimedia Applications	BCS-202	1. Know and understand the structure and technologies needed in a multimedia system and be able to discriminate which technology may be more useful in order to best achieve the expected end result.
			2. Classify, Correlate and implement or effectively use advanced techniques in animation, modeling, visualization and graphics animation.
			3. Analyze and compare the different kinds of user interfaces in order to be able to decide which one will be more efficient and ergonomic according to the required specifications of the application to be developed.
			4. Recall the main concepts and operating techniques of Virtual Reality and Augmented Reality, so they can effectively decide and implement the most appropriate algorithms to suit the purpose of the application of Virtual Reality or Augmented Reality that they would like to develop.
			5. Analyze the graphics library OpenGL 3.1 (Open Graphics Library), used for graphical visualization in Windows, Linux and Mac OS platforms, as well as in workstations (Sun, Silicon Graphics) and GLSL shaders language, which willallow the students to implement shaders, graphic effects or optimization displays in scenes with lots of geometry.
	9. C-programming	BCS-203	1. Understand the basic terminology used in computer

			programming.
			2. Write, compile and debug programs in C language.
			3. Use different data types in a computer program.
			4. Design programs involving decision structures, loops and functions.
			5. Evaluate the usability of File and pre-processors of c Programming terminology.
	10. Computer Graphics Lab	BCS-202-P	1. Know and understand the structure and technologies needed in a multimedia system and be able to discriminate which technology may be more useful in order to best achieve the expected end result.
			2. Classify, Correlate and implement or effectively use advanced techniques in animation, modelling, visualization and graphics animation.
			3. Analyze and compare the different kinds of user interfaces in order to be able to decide which one will be more efficient and ergonomic according to the required specifications of the application to be developed.
			4. Recall the main concepts and operating techniques of Virtual Reality and Augmented Reality, so they can effectively decide and implement the most appropriate algorithms to suit the purpose of the application of Virtual Reality or Augmented Reality that they would like to develop.
			5. Analyze the graphics library OpenGL 3.1 (Open Graphics Library), used for graphical visualization in Windows, Linux and Mac OS platforms, as well as in workstations (Sun, Silicon Graphics) and GLSL shaders language, which willallow the students to implement shaders, graphic effects or optimization displays in scenes with lots of geometry.
	11. C-Programming Lab	BCS-203-P	1. Understand the basic terminology used in computer programming.
			2. Write, compile and debug programs in C language.

			3. Use different data types in a computer program.
			4. Design programs involving decision structures, loops and functions.
			5. Evaluate the usability of File and preprocessors of c Programming terminology.
	12. **Discrete Mathematics**	**BCS-301**	1. Demonstrate the ability to write and evaluate a proof or outline the basic structure of and give examples of each proof technique described.
			2. Understand the basic principles of sets and operations in sets.
			3. Demonstrate an understanding of relations and functions and be able to determine their properties.
			4. Acquire ability to describe computer programs (e.g. recursive functions) in a formal mathematical manner
			5. Have substantial experience to comprehend formal logical arguments.
	13. Operating System	**BCS-302**	1. Analyze the structure of OS and basic architectural components involved in OS design.
			2. Analyze and design the applications to run in parallel either using process or thread models of different OS.
			3. Demonstrate the various device and resource management techniques for timesharing and distributed systems
			4. Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system.
			5. Interpret the mechanisms adopted for file sharing in distributed. Applications and Conceptualize the components involved in designing a contemporary OS.
	14. Data structure & Programming with "C	**BCS-303**	1. Analyze algorithms and algorithm correctness.
			2. Summarize searching and sorting techniques.
			3. Describe stack, queue and linked list operation.
			4. Ability to have knowledge of tree and graphs concepts.
			5. To develop application using data structures.

15. Operating System Lab	**BCS-302-P**	1.	Analyze the structure of OS and basic architectural components involved in OS design.
		2.	Analyze and design the applications to run in parallel either using process or thread models of different OS.
		3.	Demonstrate the various device and resource management techniques for timesharing and distributed systems.
		4.	Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system.
		5.	Interpret the mechanisms adopted for file sharing in distributed. Applications and Conceptualize the components involved in designing a contemporary OS.
16. Data Structure Lab	**BCS-303-P**	1.	Analyze algorithms and algorithm correctness.
		2.	Summarize searching and sorting techniques.
		3.	Describe stack, queue and linked list operation.
		4.	Ability to have knowledge of tree and graphs concepts.
		5.	To develop application using data structures.
17. Computer Hardware Technology & Installation	**BCS-304S-A**	1.	Understand what all the terms highlighted in bold in the text mean.
		2.	Understand the fundamental hardware components that make up a computer's hardware and the role of each of these components.
		3.	Understand the difference between an operating system and a application program, and what each is used for in a computer.
		4.	Describe some examples of computers and state the effect that the use of computer technology has had on some common products.
		5.	Identify the principal components of a given computer system and draw a diagram after the style of Figures 6 and 12 to represent the data flows between them.

18. Applications of Office Tools	**BCS-304S-B**	1.	Understanding the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming.
		2.	Analyze and understand in-depth training in use of office automation packages, internet etc.
		3.	Enhance the ability of essential for common man for day to day office management, and e-governance.
		4.	To understand what computer is? To understand the Basics of Operating systems.
		5.	To evaluate how to use software packages in day to day activities.
19. Computer Communication Networks	**BCS-401**	1.	Independently understand basic computer network technology.
		2.	Understand and explain Data Communications System and its components.
		3.	Identify the different types of network topologies and protocols.
		4.	Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer
		5.	Identify the different types of network devices and their functions within a network and understand and building the skills of sub netting and routing mechanisms.
20. DBMS	**BCS-402**	1.	Differentiate database systems from file systems by enumerating the features provided by database systems and describe each in both function and benefit.
		2.	Analyze an information storage problem and derive an information model expressed in the form of an entity relation diagram and other optional analysis forms, such as a data dictionary.
		3.	Demonstrate an understanding of the relational data model.
		4.	Demonstrate an understanding of normalization theory and

			apply such knowledge to the normalization of a database.
			5. Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database. Use a desktop database package to create, populate, maintain, and query a database.
	21. Object Oriented Programming & C++	**BCS-403**	1. Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
			2. Develop solutions for a range of problems using objects and classes.
			3. Programs to demonstrate the implementation of constructors, destructors and operator overloading.
			4. Apply fundamental algorithmic problems including type casting, inheritance, and polymorphism.
			5. Understand generic programming, templates, file handling.
	22. DBMS LAB	**BCS-402-P**	1. Differentiate database systems from file systems by enumerating the features provided by database systems and describe each in both function and benefit.
			2. Analyze an information storage problem and derive an information model expressed in the form of an entity relation diagram and other optional analysis forms, such as a data dictionary.
			3. Demonstrate an understanding of the relational data model.
			4. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
			5. Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database. Use a desktop database package to create, populate, maintain, and query a database.
	23. C++ LAB	**BCS-403-P**	1. Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
			2. Develop solutions for a range of problems using objects and classes.

			3. Programs to demonstrate the implementation of constructors, destructors and operator overloading.
			4. Apply fundamental algorithmic problems including type casting, inheritance, and polymorphism.
			5. Understand generic programming, templates, file handling.
24. Designing with Corel draw and Photoshop	**BCS-404S- A**		1. Basic knowledge of hardware and software used during graphic processing, work with vector and bitmap graphics software.
			2. Exploring DTP techniques, different graphic formats and pre-press jobs.
			3. Practical implementation of CorelDraw, Adobe Illustrator, Adobe Photoshop.
			4. Making 2D vector graphics, web graphics, text documents, posters for project presentations and technical drawings.
			5. Implementation of artistic text for the creation of logos, labels and any other one page print design material.
25. Principles of Animation	**BCS-404S- B**		1. Create animated sequences from the development of the original concept through design to final film or video production.
			2. Communicate ideas, believable action and emotion effectively by employing principles of animation and performance in all aspects of drawing.
			3. Integrate the concepts, principles and theories involved in the physics of animation in all aspects of drawing
			4. Refine personal narrative voice that holistically integrates the elements of storytelling and performance in order to actively engage the audience.
			5. Create 2D and 3D characters and environments that reflect the integration of graphic clarity, design principles, performance principles and theoretical constructs.
26. Java programming and web page	**BCS-501-A**		1. Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc.

	design		2.	Identify and analyze classes, objects, members of a class and relationships among them needed for a specific problem.
			3.	Demonstrate the concepts of polymorphism and inheritance.
			4.	Understand the principles of inheritance, packages and interfaces.
			5.	Understand the fundamentals of object-oriented programming in Java, including defining classes , objects, invoking methods etc and exception handling mechanisms.
	27. Digital electronics & computer organization	**BCS-501-B**	1.	Explain the generic principles that underlie the design of digital computer, including data representation, digital logic and process simulation.
			2.	Describe the structure and functioning of a digital computer, including its overall system architecture, operating system, and digital components.
			3.	Apply and Implement fundamental coding schemes.
			4.	Understand the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit.
			5.	Understand the evolution of processors, their present technology and inter-process communication.
	28. ERP- System	**BCS-502-A**	1.	To comprehend the technical aspects of ERP systems.
			2.	To understand concepts of reengineering and how they relate to ERP system implementations.
			3.	To be able to map business processes using process mapping techniques and to understand the steps and activities in the ERP life cycle.
			4.	To be able to identify and describe typical functionality in an ERP system.
			5.	To obtain practical hands on experience with one of the COTS ERP Software e.g. SAP, Oracle.
	29. Software Engineering	**BCS-502-B**	1.	Analyze and resolve information technology problems through the application of systematic approaches and diagnostic tools. Support the implementation and administration of computer systems.

			2. Demonstrate an understanding of and apply current theories, models, and techniques that provide a basis for the software lifecycle.
			3. Demonstrate and compare an ability to use the techniques and tools necessary for engineering practice.
			4. Know and understand classical and evolving software engineering methods, can select and tailor appropriate methods for projects, and can apply them as both team members and managers to achieve project goals.
			5. Analyze basic software quality assurance practices to ensure that software designs, development, and maintenance meet or exceed applicable standards.
	30. Neural Networks & Its Application	**BCS-503-A**	1. To understand the fundamental theory and concepts of neural networks, neuro-modeling, several neural network paradigms and its applications.
			2. To understand the concepts of fuzzy sets, knowledge representation using fuzzy rules, approximate reasoning, fuzzy inference systems, and fuzzy logic control and other machine intelligence applications of fuzzy logic.
			3. To understand the basics of an evolutionary computing paradigm known as genetic algorithms and its application to engineering optimization problems.
	31. Multimedia System	**BCS-503-B**	1. Analyze fundamentals principles of multimedia, including digitization and data compression for non-textual information.
			2. To understand issues in representing, processing, and transmitting multimedia data.
			3. Describe core multimedia technologies and standards.
			4. To gain hands-on experience in image, sound and video editing and in some aspects of various computer application.
			5. Knowledge about Multimedia authoring (incorporating images, sound, video, and animation) To design, capture, store and integrate sound, images and video to deliver multimodal information.

32. Software Testing Methodology	**BCS-504-A**	1. Define a range of different software testing techniques and strategies and be able to apply specific (automated) unit testing method to the projects.	
		2. Distinguish characteristics of structural testing methods.	
		3. Understand the concept of integration testing which aims to uncover interaction and compatibility problems as early as possible.	
		4. Discuss about the functional and system testing methods.	
		5. Explain various issues for object oriented testing.	
33. Web Designing	**BCS-504-B**	1. Design and implement solutions to problems encountered in all phases of the design process.	
		2. Explain visual communications through the application of design theories and principles to develop effective design solutions.	
		3. Define typographic skills and knowledge to create effective visual communications.	
		4. Discuss the working of web elements like buttons, banners & Bars and of course complete UI designs.	
		5. Understanding the debug JavaScript code, making use of good practice and debugging tools.	
		6. Explain JavaScript libraries (e.g. J Query) to create dynamic pages.	
34. Practical Lab-1	**BCS-501-P**	1. Understanding and implementation of Java programming, multi-threaded programs and Exception handling.	
		2. Knowledge of object-oriented paradigm in the Java programming language.	
		3. Understand the principles of inheritance, packages and interfaces.	
		4. Explain the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc and exception handling mechanisms.	
35. Practical Lab-2	**BCS-504-P**	1. Design and implement solutions to problems encountered in all phases of the design process.	
		2. Explain visual communications through the application of design theories and principles to develop effective design solutions.	
		3. Define typographic skills and knowledge to create effective visual	

			communications.
			4. Discuss the working of web elements like buttons, banners & Bars and of course complete UI designs
			5. Understanding the debug JavaScript code, making use of good practice and debugging tools.
			6. Explain JavaScript libraries (e.g. J Query) to create dynamic pages.
	36. Mobile Computing	**BCS-601-A**	1. Describe the basic principles of Mobile Computing.
			2. Analyze the functionalities and components of mobile computing systems into different layers and apply various techniques for realizing the functionalities.
			3. Analyze requirements and solve problems using systematic planning and development approaches.
			4. Demonstrate mobile computing applications by analyzing their characteristics and requirements.
			5. Define the concepts and features of mobile computing technologies and applications.
	37. Information systems: analysis, design & implementation	**BCS-601-B**	1. Gather data to analyse and specify the requirements of a system.
			2. Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
			3. Build general and detailed models that assist programmers in implementing a system.
			4. Design a database for storing data and a user interface for data input and output, as well as controls to protect the system and its data.
			5. Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.
	38. Web Technology & Cyber Security	**BCS-602-A**	1. To master information security governance, and related legal and regulatory issues.
			2. Understanding external and internal threats to an organization

			with information security awareness.
			3. Knowledge about network security threats and countermeasures.
			4. Explain advanced security issues and technologies (such as DDoS attack detection and containment, and anonymous communications,)
			5. Discuss about network security threats and countermeasures.
	39. .Net framework and c#	**BCS-602-B**	1. Identify important events and individuals in the history of human-computer interfaces.
			2. Design and develop Windows application using different Windows technologies that use a variety of GUI controls and classes to fulfill specific user requirements.
			3. Analyze how event driven applications use threading to perform time- consuming operations.
			4. Demonstrate how to use specific features of the C# programming language to write object- oriented programs and handle run-time errors.
			5. Evaluate public setting how user interfaces should be designed to accommodate human physiology and limitations.
	40. Computer Architecture with CASE study of microprocessor	**BCS-603-A**	1. Identify the factors affecting performance in superscalar processors and the key components, options and tradeoffs that a designer has to consider when designing such processors.
			2. Compare a trace cache to conventional instruction cache and explain advantages and disadv antages of each approach.
			3. Compare checkpoint architectures to conventional speculative execution methods, such as reorder buffer.
			4. Describe and compare various latency tolerant architectures, such as: run-ahead execution, continual flow pipelines, and out-of-order commit processors.
			5. Explain the hardware support needed to implement virtualization technology and the benefits and usages of virtualization.
	41. Real Time System	**BCS-603-B**	1. Explain the real time system and present the mathematical model of the system.

			2. Define real-time algorithm for task Management.
			3. Understand the working of real-time operating systems and real-time database.
			4. Design and development of protocols related to real-time communication.
			5. Discuss the term of Fault classes and Fault Tolerant in real time system and also discuss about the issues in real time software design.
	42. Cyber Forensic	**BCS-604-A**	1. Understand the definition of computer forensics fundamentals.
			2. Describe the types of computer forensics technology.
			3. Analyze various computer forensics systems.
			4. Illustrate the methods for data recovery, evidence collection and data seizure.
			5. Summarize duplication and preservation of digital evidence.
	43. Artificial Intelligence	**BCS-604-B**	1. Describe the key components of the artificial intelligence (AI) field and its relation and role in Computer Science.
			2. Identify and describe artificial intelligence techniques, including search heuristics, knowledge representation, automated planning and agent systems, machine learning, and probabilistic reasoning.
			3. Identify and apply AI techniques to a wide range of problems, including complex problem solving via search, knowledge-base systems, machine learning, probabilistic models, agent decision making.
			4. Design and implement appropriate AI solution techniques for such problems.
			5. Analyze and understand the computational trade-offs involved in applying different AI techniques and models.
	44. Practical Lab-1	**BCS-602-P**	1. Understand the .Net language in the aspects of designing, coding and implementation.
			2. Knowledge about new ideas and advances, techniques, and tools and to use them effectively.

			3. Implement an interactive and effective student progress monitoring system.
	45. Project VIVA-VOCE	**BCS-605**	1. Understand the .Net language in the aspects of designing, coding and implementation.
			2. Knowledge about new ideas and advances, techniques, and tools and to use them effectively.
			3. Implement an interactive and effective student progress monitoring system.