B.B.COLLEGE, ASANSOL

PROGRAM OUTCOMES, PROGRAM SPECIFIC

OUTCOMES AND COURSE OUTCOMES

	B.Sc. Computer Science(H)
Program Outcomes	The Computer Science Department's Bachelor of Science program must enable students to attain, by the time of graduation: ➢ An ability to apply knowledge of computing and mathematics appropriate to the discipline. ➢ An ability to identify, formulate, and develop solutions to computational challenges. ➢ An ability to design, implement, and evaluate a computational system to meet desired needs within realistic constraints. ➢ An ability to function effectively on teams to 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. ➢ An ability to analyze impacts of computing on individuals, organizations, and society. ➢ Recognition of the need for and ability to engage in continuing professional development. ➢ An ability to use appropriate techniques, skills, and tools necessary for computing practice. ➢ An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computational systems in a way that demonstrates comprehension of the tradeoffs

	involved in design choices. ➢ An ability to apply design and development principles in the construction of software systems of varying complexity.
Program Specific Outcomes	**PSO1**: On completion of the B.Sc.(H) in Computer Science degree the graduates will be able to ❖ Apply standard Software Engineering practices and strategies in real-time software project development using open-source programming environment or commercial environment to deliver quality product for the organization success. ❖ Design and develop computer programs/computer-based systems in the areas related to algorithms, networking, and web design. **PSO2**: ❖ **Theoretical Computer Science:** Students at the time of graduation will be able to apply fundamental knowledge of theoretical computer science and critically analyze problems to provide computer based solutions for engineering applications. ❖ **Hardware and software systems**: Students at the time of graduation will be able to design cost effective hardware/software systems and components for engineering/social applications using the knowledge of hardware and/or software architecture, programming and development. ❖ **Technology:** Students at the time of graduation will be able to apply appropriate technology to

find solutions for complex problems.

❖ **Research Capability:** Students at the time of graduation will be able to apply domain knowledge and expertise for enhancing research capability to transform innovative ideas into reality.

Course	Outcomes
Programming Methodology	Upon successful completion of this subject, students should be able to: ▪ Learn the fundamental programming concepts and methodologies which are essential to building good C/C++ programs. ▪ Practice the fundamental programming methodologies in the C/C++ programming language via laboratory experiences. ▪ Code, document, test, and implement a well-structured, robust computer program using the C/C++ programming language. ▪ Write reusable modules (collections of functions).
Operating System	On successful completion of this subject, the students will be able to: ▪ Describe the general architecture of computers. ▪ Describe, contrast and compare differing structures for operating systems. ▪ Understand and analyze theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files. ▪ Explain the role of an operating system in

	managing and interacting with computer system components including main and secondary memory.
Computer Architecture & Organization	After completion of this subject, the students will be able to: ▪ Understand the theory and architecture of central processing unit. ▪ Analyze some of the design issues in terms of speed, technology, cost performance. ▪ Design a simple CPU with applying the theory concepts. ▪ Use appropriate tools to design verify and test the CPU architecture. ▪ Learn the concepts of parallel processing, pipelining and interprocess communication. ▪ Understand the architecture and functionality of central processing unit. ▪ Exemplify in a better way the I/O and memory organization. ▪ Define different number systems, binary addition and subtraction, 2 s complement representation and operations with this representation.
System Analysis & Design	Upon successful completion of this subject, the students will be able to ▪ Gather data to analyze and specify the requirements of a system. ▪ Design system components and environments build general and detailed models that assist

	programmers in implementing a system. ▪ Design a database for storing data, a user interface for data input and output, and controls to protect the system and its data.
Software Engineering	On successful completion of this subject, the students will be able to: ▪ Obtain knowledge about principles and practices of software engineering. ▪ Define and develop a software project from requirement gathering to implementation. ▪ Focus on the fundamentals of modeling a software project. ▪ Obtain knowledge about estimation and maintenance of software system. ▪ Provide a professionally guided education in software engineering to transition into a broad range of career options: industry, government, computing graduate program, and professional education. ▪ Capable of diverse team and organizational leadership in computing project settings. ▪ Demonstrates ethical principles in the application of computing-based solutions to societal and organizational problems.
UNIX & Shell Programming Lab	Upon successful completion of this subject, students should be able to: ▪ To provide introduction to UNIX Operating System and its File System

	▪ To gain an understanding of important aspects related to the SHELL and the process ▪ To develop the ability to formulate regular expressions and use them for pattern matching. ▪ To provide a comprehensive introduction to SHELL programming, services and utilities. ▪ To write a shell script for specific problem definition ▪ To employ decision making and looping construct to write a shell script. ▪ To discuss various modes in which Vi editor operates. ▪ To differentiate between internal and external commands of UNIX. ▪ To discuss the importance of filters and their need in UNIX.
Computer Graphics	After completion of this subject, the students will be able to: ▪ Explain the core concepts of computer graphics, including viewing, projection, perspective, modeling and transformation in two and three dimensions. ▪ Apply the concepts of colour models, lighting and shading models, textures, ray tracing, hidden surface elimination, anti-aliasing, and rendering. ▪ Interpret the mathematical foundation of the concepts of computer graphics. ▪ Describe the fundamentals of animation, parametric curves and surfaces, and spotlighting.

	▪ Identify a typical graphics pipeline and apply graphics programming techniques to design and create computer graphics. ▪ Create effective programs to solve graphics programming issues, including 3D transformation, objects modeling, colour modeling, lighting, textures, and ray tracing.
Artificial Intelligence	Upon completion of this subject, the students will be able to: ▪ Understand the concept of knowledge representation and predicate logic and transform the real life information in different representation. ▪ Understand the state space and its searching strategies. ▪ Understand the machine learning concepts and range of problems that can be handled by machine learning. ▪ Apply the machine learning concepts in real life problems.
Core Java	On completion of the course, the student will be able to: ▪ Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs. ▪ Read and make elementary modifications to Java programs that solve real-world problems. ▪ Understanding the OOP s concepts, classes and objects, threads, files, applets, swings and act. ▪ Emphasis is placed on event-driven

	programming methods, including creating and manipulating objects, classes, and using Java for network level programming and middleware development
	▪ Build Java Application for distributed environment.
	▪ Design and Develop multi-tier applications.
	▪ Identify and Analyze Enterprise applications.
Cryptography & Network Security	By the end of the course, the students will be able to: ▪ Learn fundamentals of cryptography and its application to network security. ▪ Understand network security threats, security services. ▪ Acquire background on well known network security protocols. ▪ Understand vulnerability analysis of network security. ▪ Acquire background on hash functions; authentication, firewalls, detection techniques. ▪ Understand various Cryptographic Techniques. ▪ Apply various public key cryptography techniques. ▪ Implement Hashing and Digital Signature techniques. ▪ Understand the various Security Applications. ▪ Implement system level security applications.

Prepared by: SUTAPA MAJEE (Computer Science Department)