COURSE OBJECTIVES

B.Sc. 1St year Computer Science (CBCS) Honours

Paper Code: CSC-HC-1016 Programming Fundamentals using C/C++

After completing the course, the students are able to :

1. Develop simple algorithms and flow charts to solve a problem.
2. Develop problem solving skills coupled with top down design principles.
3. Learn about the strategies of writing efficient and well-structured computer algorithms/programs.
4. Develop the skills for formulating iterative solutions to a problem.
5. Learn array processing algorithms coupled with iterative methods.
6. Learn text and string processing efficient algorithms.
7. Learn searching techniques and use of pointers.
8. Understand recursive techniques in programming.
9. Learn Object Oriented programming
10. Develop skills to use the paradigms of Object Oriented Programming

Paper Code: CSC-HC-1026 Computer System Architecture

After completing the course, the students are able to :

1. Gather knowledge on Boolean Algebra, combinational & sequential circuits.
2. Understand the basic structure, operation and characteristics of digital computer.
3. Understand arithmetic and logic unit as well as the concept of pipelining.
4. Understand hierarchical memory system including cache memories and virtual memory.
5. Understand the communicating principles with I/O devices and standard I/O interfaces

Paper Code: CSC-HC-2016 Programming in JAVA

After completing the course, the students are able to :

1. Know the strength and weaknesses of Java Technololgy
2. Know of the structure and model of the Java programming language
3. Use the Java programming language for implementing various programming problems.
4. Develop software in the Java programming language,
5. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.

Paper Code: CSC-HC-2026 Discrete Structures

After completing the course, the students are able to :

1. Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.
2. Understand the basics of combinatorics, and be able to apply the methods from these subjects in problem solving.
3. Use effectively algebraic techniques to analyse basic discrete structures and algorithms.
4. Understand asymptotic notation, its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples.
5. Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.

****