2- NARS CHARACTERIZATION OF COMPUTER SCIENCE

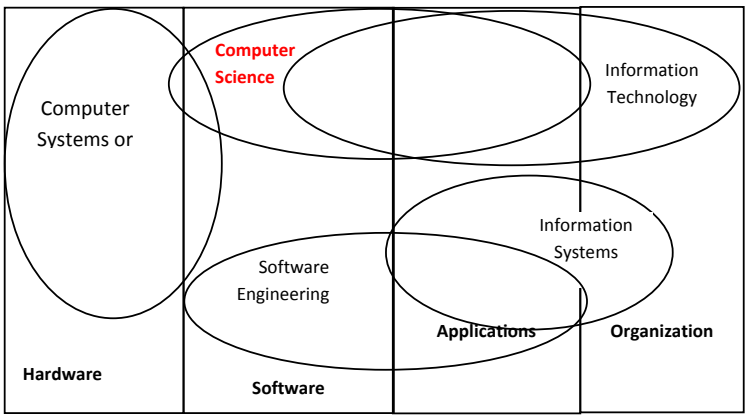
2.1 Introduction

Computer science spans a wide range, from its theoretical and algorithmic foundations to World Wide Web and its applications, computer vision, intelligent systems, bioinformatics, high performance computing, distributed systems, Object-oriented programming, grid and cloud computing and other exciting areas.

The work of computer scientists could be fallen into three categories.

- Design and implement software. Computer scientists take on challenging programming jobs.
- Supervise other programs by keeping them aware of new approaches.
- Devise new ways to use computers. Progress in the CS areas of networking, database, and human-computer-interface enabled the development of the World Wide Web.

Now CS researchers are working with scientists from other fields to make robots become practical and intelligent aides, to use databases to create new knowledge, and to use computers to help decipher the secrets of our DNA. They develop effective ways to solve computing problems. For example, computer scientists develop the best possible ways to store information in databases, send data over networks, and display complex images. Their theoretical background allows them to determine the best performance possible, and their study of algorithms helps them to develop new approaches that provide better performance.



2.2 The Attributes of Computer Science Graduate

The Computer Science program is designed to provide the student with the foundations of the discipline as well as the opportunity for specialization. After successfully completing the Computer Science program, the graduate should be able to:

1. Demonstrate knowledge and competence in fundamental areas of computer science such as: algorithms, design and analysis, computational theory, computer architecture and software based systems.
2. Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design, implementation, evaluation and evolution of computer-based systems.
3. Apply knowledge of mathematics and science to real world problems; as well as to analyze and interpret data.
4. Demonstrate the analytic skills necessary to effectively evaluate the relative merits of software and computer systems, and algorithmic approaches.
5. Understand and apply a wide range of principles and tools of software engineering, such as design methodologies, choice of algorithm, language, software libraries and user interface technique.
6. Understand and apply a wide range of principles and tools of natural language processing and data mining
7. Have a solid understanding of the used concepts in computer science to be able to pursue further learning, whether as graduate students or on their own.
8. Demonstrate an understanding of algorithms and data structures, computer organization and architecture, programming language concepts, compilers, networks, artificial intelligence, graphics, human computer interfaces, and databases, and identify and define the computing requirements for its solution.
9. Design, implement, and evaluate a computer-based systems, process, component or program.
10. Use knowledge and understanding in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.

2.4. National Academic Reference Standards for Computer Science

2.4.1 Knowledge and Understanding

In addition to **Knowledge and Understanding** of computing and information graduate, the Computer Science graduate should be able to:

1. Understand the essential mathematics relevant to computer science.
2. Use high-level programming languages.
3. Demonstrate basic knowledge and understanding of a core of analysis, algebra, applied mathematics and statistics.
4. Interpret and analyzing data qualitatively and/or quantitatively.
5. Know and understand the principles and techniques of a number of application areas informed by the research directions of the subject, such as artificial intelligence, natural language processing, data mining, databases and computer graphics.
6. Show a critical understanding of the principles of artificial intelligence, image, and pattern recognition.
7. Understand the fundamental topics in Computer Science, including hardware and software architectures, software engineering principles and methodologies, operating systems, compilers, parallel and distributed computing, systems and software tools.
8. Select advanced topics to provide a deeper understanding of some aspects of the subject, such as hardware systems design, object-oriented analysis and design, and artificial intelligence, and parallel and concurrent computing.

2.4.2 Intellectual Skills

In addition to Intellectual of computing and information graduate, the Computer Science graduate should be able to:

1. Define traditional and nontraditional problems, set goals towards solving them, and. observe results.
2. Perform comparisons between (algorithms, methods, techniques...etc).
3. Perform classifications of (data, results, methods, techniques, algorithms.. etc.).
4. Identify attributes, components, relationships, patterns, main ideas, and errors.
5. Summarize the proposed solutions and their results.
6. Restrict solution methodologies upon their results.
7. Establish criteria, and verify solutions.
8. Identify a range of solutions and critically evaluate and justify proposed design solutions.
9. Solve computer science problems with pressing commercial or industrial constraints.
10. Generate an innovative design to solve a problem containing a range of commercial and industrial constraints.

2.4.3 Professional and Practical Skills

In addition to Professional and Practical Skills of computing and information graduate, the Computer Science graduate should be able to:

1. Use appropriate programming languages, web-based systems and tools, design methodologies, and knowledge and database systems.
2. Communicate effectively by oral, written and visual means.
3. Perform independent information acquisition and management, using the scientific literature and Web sources.
4. Prepare and present seminars to a professional standard.
5. Perform independent information acquisition and management, using the scientific literature and Web sources.
6. Prepare technical reports, and a dissertation, to a professional standard; use IT skills and display mature computer literacy.
7. Specify, design, and implement computer-based systems.
8. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.
9. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video.
10. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, and multimedia systems.
11. Identify any risks or safety aspects that may be involved in the operation of computing equipment within a given context.
12. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.
13. Prepare technical reports, and a dissertation, to a professional standard.