AICTE Report

Model Curriculum

for

Undergraduate Degree Courses

in

Computer Science and Engineering

(Engineering & Technology)

Report submitted by the committee constituted by AICTE for model design.
March 2022

Framework for CSE Curriculum Design

Context and Background

Computer Science has changed considerably in the last few years with areas like Machine Learning and Cloud computing becoming much more important. At the same time, the technologies and underlying computing systems have also evolved, considerably easing executing some of the tasks that earlier took much more training and experience. These changes require that teaching of computer science ought to suitably adapt – to reflect the changed nature of the discipline, as well as to update courses with the more recent technology platforms. And this ought to be done while providing some flexibility to the HEIs to address their specific constraints and focus. With these in mind, the committee for model curriculum design for CSE established some of the guiding principles for the exercise:

- The focus of curriculum design is the 4 yr BE/BTech program in CSE and the target audience of the curriculum is the vast majority of universities/institutions, rather than the premier institutes (who design their own curriculum and have the expertise for it.)

- The existing CSE curriculum of AICTE will be the starting point. Suitable enhancements/modifications will be made.

- **Flexibility**. Even within the large non-top tier education system, which is the primary target, some have better infrastructure and capability, so it is desirable to provide a limited degree of flexibility to the HEIs on the curriculum.

  To provide this flexibility, for each course, the learning outcomes will be grouped in two – essential, and desired/advanced. The core courses will also be identified as essential and desired/advanced.

  This simple framework of grouping outcomes and courses as essential and desired provides limited flexibility to institutions to design their curriculum depending on their capabilities, resources, goals, etc, while still providing guidance for a sound curriculum. The flexibility can be leveraged by institutions to provide different pathways to students, and multiple exits.

- **Early Exits**. The new education policy (NEP) aims to provide multiple exit points to students. To support more than one exits for students, it is important to develop employability skills early, and not take the approach of first focusing on foundations and then develop practical skills. Also, we believe that all but a few students will go for full degree, hence while providing flexibility for early exit, the outcomes of the 4 yr degree should not be compromised. We propose:

  - Eliminate the separation of theory and labs, instead break courses topic or theme-wise. This will allow both basic theory and basic skills to be taught together and build skills early.

  - Introduce discipline courses early, so disciplinary skills can be developed early. For this have one sem common program, and allow some discipline courses in 2nd sem – this will also help students in getting internships / part-time jobs in summer. This will require branch change to be done after one semester. (For lateral entry, some make-up courses may be needed.)

  With this, and the flexibility provided in the curriculum, an HEI can design suitable exits.

- **Provide Guidance for labs/assignments.** A small analysis the committee did on usage of AICTE curriculum indicated that while the courses and topics specified in the curriculum are widely used, HEIs differ considerably in types of assignments/labs/projects they give. Also, this is where there is a perceived deficiency in education – while theory is covered suitable assignments/labs are not give for students to develop skills. To help in this, it was decided that for each course design, experts will also recommend nature of labs and assignments for each module in the course.
- **Pedagogy suggestions**. With the emergence of a host of online resources that can be used to support teaching, there may be different ways to teach a topic. To help teachers in pedagogy, for each module in a course, pedagogy suggestions have also been provided.
- **Multiple Pathways**. For supporting multiple pathways within the academic program, we propose to provide for micro specializations through thematic course streams. These can be further enhanced by HEIs with programs like honors for advanced students with more credits or advanced learning outcomes,etc.

Graduate Attributes

Curriculum of a program is finally a network of credit units – courses (core, disciplinary core, disciplinary elective, open), internships, practice, projects, etc. which help achieve program goals. Program goals can be stated as attributes the students should possess on graduation, i.e. statements about the learning, values, capabilities etc. of graduates. These are called Graduate Attributes (GAs). A program typically has:

- General GAs: which are often common across many similar programs (e.g. BTechs) and focus on generalized skills and capabilities in the graduate.
- Discipline GAs: are discipline specific attributes, which focus on understanding of different concepts and systems related to the discipline, and on competencies and skills in that discipline.

Together the GAs define the goals of the program. The aim of curriculum design is to evolve a curriculum that can develop in students the stated graduate attributes. While specifying the GAs and designing a curriculum for it, we must keep a basic constraint in mind: a full BTech program has 8 semesters, each with about 5 full courses. GAs should specify only what can be taught and absorbed in this timebox (i.e. we cannot push more simply by adding more.)

Desired Graduate Attributes for the CSE program are given below. The CSE curriculum design will focus more on delivering the discipline GAs, while strengthening the general GAs, where possible. Feedback on these Graduate Attributed was taken from many representatives from industry, as well as from HEIs. (GAs should be read by adding this at the start of each: *At graduation time, a student should have:*

General Graduate Attributes	**Discipline Graduate Attributes (for CSE)**

G1 Ability to identify a problem, analyze using design thinking techniques, and evolve innovative approaches for solving it.	CS1 Proficiency in writing in at least two dissimilar programming languages programs of modest complexity which are: readable, tested for correctness, efficient, and secure.
G2 Ability to apply mathematical concepts and techniques in problem solving.	CS2 Ability to design and apply appropriate algorithms and data structures for evolving efficient computing based solutions for new problems.
G3 Ability to function effectively in multi- cultural teams to accomplish a common goal.	CS3 Understanding of computing systems at computer architecture, operating systems, and distributed- computing levels, and how they affect the performance of software applications.
G4 Ability to communicate effectively with a wide range of audience.	CS4 Understanding of theoretical foundations, fundamental principles, and limits of computing.
G5 Ability to self-learn and engage in life-long learning and upgrade technical skills	CS5 Ability to analyse large volumes of data employing a variety of techniques for learning, better prediction, decision making, etc
G6 An understanding of professional and ethical responsibility	**ADVANCED/OPTIONAL GAs** CS6 Ability to design, implement, and evaluate computer based system or application to meet the desired needs using modern tools and methodologies
G7 Ability to undertake small research tasks and projects.	CS7 Ability to develop full stack applications using one commonly used tech-stack and modern tool.
G8 An entrepreneurial mindset for opportunities using technology and innovations.	CS8 Understanding of and ability to use advanced techniques and tools in a few different domain areas (e.g. parallel processing, image processing, IR, …)
G9 An understanding of impact of solutions on economic, societal, and environment context.	CS9 Exposure to emerging technologies such as Cloud Computing, IoT, etc
G10 Strong emotional intelligence, human and cultural values	

CSE Core Courses

For identifying the core courses for a robust CSE BTech program, the following assumptions have been made:

- All CSE program have the first course on programming, generally called “Introduction to Programming”, with different institutions choosing different languages. This course is generally common across all engineering disciplines. It is assumed that this course is part of the common core (and hence is not included in the CSE core courses)

- It is assumed that there are at least two Math courses that are part of common core which all CSE students will do – a course on Probability and Statistics, and a course on Linear Algebra.

In addition to these, to support multiple exits, it is also suggested that In first year, the common program should have an engineering design course where a project to build a system for some purpose is the main goal for students, and lectures support the project.

With these assumptions, the list of core courses recommended for CSE are given below. As explained above, the core has been split into two – essential and additional. The essential core is what all programs must have. The additional core list those courses which HEIs can choose to make core or an elective, depending on their education approach and resources.

Suggested Year	**Course Name**	**CS GAs it Directly Contributes to (GAs it supports)**
	Professional Core Courses (Essential)	
2nd year	Data structures and Algorithms (DS)	CS1, CS2, CS4
2nd year	Discrete Mathematics (DM)	CS4 (CS2, CS4, CS6)
2nd year	Computer Organization and Architecture (COA)	CS3 (CS4, CS6)
2nd year	Advanced Programming (in lieu of OO prog.) (AP)	CS1, CS3, CS7, CS8 (CS2)
3rd year	Operating Systems (OS)	CS3 (CS6, CS8)
3rd year	Design and Analysis of Algorithms (Algo)	CS1, CS2, CS4
3rd year	Database Systems (DB)	CS3, CS5, CS6 (CS1, CS7, CS9)
3rd/4th year	Computer Networks (NW)	CS3 (CS4, CS6, CS8)
3rd year	Machine Learning (New) (ML)	
3rd year	Security (New)	CS1, CS3, CS6
	Extended Professional Core (Additional)	
3rd/4th year	Compiler Design	
3rd/4th year	Theory of Computation	CS4 (CS2)

Detailed syllabus of each of these courses follows – a separate subsection for each course. A common template has been used to specify the course design. Each course design lists essential learning outcomes for the course, and desired/advanced learning outcomes. As the terms suggest, all HEIs should ensure that the essential learning outcomes are achieved. The desired/advanced learning outcomes are those which those HEIs which have the capabilities and resources to deliver may include in their courses.

Coverage of CS Graduate Attributes by the Core Courses

This table summarizes which CS GAs different courses contribute to. For each GA, list of main courses is given – these are the courses which directly contribute to the GA. Also mentioned are the courses which support the GA (though perhaps less directly):

CS1 Proficiency in writing in at least two dissimilar programming languages programs of modest complexity which are: readable, tested for correctness, efficient, and secure	Main Courses: DS, AP, Algo, Security Supporting Courses: DB
CS2 Ability to design and apply appropriate algorithms and data structures for evolving efficient computing based solutions for new problems	Main: DS, Algo Supporting: DM, AP, ToC
CS3 Understanding of computing systems at computerarchitecture, operating systems, and distributed- computing levels, and how they affect the performance of software application	Main: CO/CA, AP, OS, DB, NW, Security Supporting:
CS4 Understanding of theoretical foundations, fundamental principles, and limits of computing	Main: DS, DM, Algo, ToC Supporting: NW, CO/CA
CS5 Ability to analyse large volumes of data employing a variety of techniques for learning, better prediction, decision making,	Main: DB Supporting: DM
ADVANCED/OPTIONAL CS6 Ability to design, implement, and evaluate computer based system or application to meet the desired needs using modern tools and methodologies	Main: AP, DB, Security Supporting: DM, CO/CA
CS7 Ability to develop full stack applications using one commonly used tech-stack and modern tools	Main: AP Supporting: OS, NW, Security
CS8 Understanding of and ability to use advanced techniques and tools in a few domain areas (e.g. parallel processing, image processing, IR, …)	Main: Supporting: DB
CS9 Exposure to emerging technologies such as Cloud Computing, IoT, etc	Main: Supporting: DB, Security, NW

Micro Specializations (and Professional Electives)

Besides the core courses, programs normally have professional elective courses, which HEIs decide. It is possible to use the electives to provide a limited specialization in some sub-area of CSE to a BTech student. We call these as micro-specializations. These also allow multiple pathways to students, as different students can graduate with different specializations (or not). A micro specialization is a set of a few full or half courses, which build upon the core CSE program. The report gives possible structure of a few micro specializations in:

- Software Engineering
- Machine Learning
- Distributed and Cloud Systems
- Human Computer Interaction (HCI)

More specializations definition may be added (e.g. in Security, High Performance Computing, Algorithms). Micro specializations are optional for HEIs – i.e. they can decide to offer them or not, and if they do, which ones they want to. If they decide not to have these, the list of courses mentioned in the micro specializations can be used as a list of suggested professional electives.