

# Computational Thinking in CSC-235

## Course Description

CSC-235 (Problem Solving) combines a systematic approach, in which different problem-solving techniques and strategies are discussed, with hands-on examples. To this end, the course concentrates on problems and puzzles arising in mathematics, logic, reasoning, philosophy, games (such as chess and cards), everyday life (reading newspapers) and many other domains. The intent of the course is to teach students how to plan an informed approach to a new problem. The course topics include: Problem Modeling, Logical Inferences, Paradoxes and Limits of Reasoning, Classifying Action Sequences, Defining Subgoals, Searching and Pruning, Relations and Reductions, and Quantitative Reasoning.

## Computational Thinking Aspects in CSC-235

Computational Thinking manifests itself in several topics of CSC-235. In particular, it is emphasized when discussing the techniques of Defining Subgoals, and the technique of Searching and Pruning.

- **Defining Subgoals.** This technique aims at defining a simpler problem (or problems) than the original problem, whose solution leads to a solution to the original problem.

A beautiful example that is usually given in class, and that illustrates this technique as well as emphasizes Computational and Algorithmic Thinking, is the Tower or Hanoi problem. In this problem we are given 3 pegs labeled A, B, C and  $n$  disks (of different sizes) placed on top of each others on peg A in a decreasing order of their size (largest at the bottom). The problem is to transfer the disks from peg A to peg C under the restriction that no disk can be placed on top of a smaller one.

To solve the problem, one can define a subgoal, which is transferring  $n-1$  disks from peg A to peg B. If this subgoal is achieved, then clearly the original goal can be achieved by first moving the  $n-1$  top disks from A to B, moving the  $n$ th disk from A to C, and then moving the  $n-1$  disks from B to C. This solution is a *recursive* solution, which sits at the heart of Computational Thinking: a repeated hierarchal and mechanical use of a sequence of well-defined steps.

When this example is covered in class, it is covered from the above perspective. In particular, the recursive nature of the solution is emphasized, and the students see how the whole process can be automated once the subgoal has been defined.

A graphical applet is used to simulate the execution of a recursive solution to the problem.

- **Searching and Pruning**

This technique uses heuristics and exhaustive search methods to arrive at a solution. It is usually used to solve problems which do not possess “easy solutions”. For such problems one may need to perform a systematic search to check all possible choices that may lead to a solution.

A nice problem that illustrates this technique is the Queens of Chess problems. The problem asks to place 8 queens on a chessboard (in general, to place  $n$  queens on an  $n \times n$  chess board) such that no two queens can attack each other.

To solve the above problem, we can search systematically for positions on which to place the 8 queens. We start by placing a queen on the first row. We proceed to place the second queen on the second row in a square that does not conflict with the placement of the first queen, if such a square exists. In general, if we have placed the  $i^{\text{th}}$  queen on row  $i$ , we proceed to place the  $(i+1)^{\text{st}}$  queen on a square in row  $(i+1)$  that does not conflict with the placement of queens 1, ...,  $i$ . If such a square does not exist, then we backtrack to change the position of queen  $i$ , and proceed again.

The above systematic search emphasizes Computational Thinking through the process of automation. As the students try solving this problem on the chess board, they soon realize that this search process can be systemized, and automated.

- **Assessment**

The students will be presented with a set of problems and they are asked to identify which technique works best for each problem, and show how to apply it to the problem. For example, to test the students’ understanding on the Defining Subgoals technique, the students will be given a relevant problem and they need to identify the subproblem (a reduction) within it whose solution leads to a solution to the original problem. They also need to illustrate the process of going from a solution of the subproblem to a solution for the original problem.