

Nifty Assignment Proposal : Problem-Solving Workshop

Correspondance : Hillary Dawkins, University of Guelph, Canada, hdawkins@uoguelph.ca

ABSTRACT

Problem solving is a crucial skill for computer science students to master. Most instruction in problem solving either assumes latent uptake (through practice with course materials), or, when explicit, is presented in parallel with new course material (requiring high cognitive overhead). We present an assignment, designed as a one-day workshop, which aims to explicitly teach problem-solving skills using simple programming tasks. Preliminary trials of the workshop show promising student engagement.

ASSIGNMENT DESCRIPTION

As part of ongoing research on the instruction of problem solving in computer science, we have developed a problem-solving checklist tool. The checklist was developed by observing strategies used by advanced computer science students in practice, and matching the observations onto known processes from the literature. This assignment aims to familiarize students with the checklist tool through practice with two simple programming tasks.

Difficulty level:

The intended audience for this assignment is second-semester undergraduate students. It is assumed that the students have some knowledge of programming in C. However, the workshop structure could easily be adapted to various skill levels or programming languages. Additionally, we have found that even advanced undergraduate students benefit from completing this assignment as presented. Because the assignment aims to teach general problem-solving skills, all programming skill levels are welcome.

Suggested workshop structure:

This assignment is intended to be completed as a one-day (approximately 6 hour) workshop. The workshop begins by briefing the students on the goals of the workshop, and introducing them to the checklist tool. Students are then divided into smaller groups (ideally 3-6 students), and each group is assigned a facilitator (which could be a senior undergraduate, graduate student, or teacher). The first half of the workshop (approximately 2.5 hours) is spent working through programming task #1 (found in the 'Tic-tac-toe task' folder) within groups with the guidance of the facilitator using worksheet #1 ('Tic-tac-toe problem-solving worksheet'). The facilitator keeps all students working on the same portion of the worksheet at the same time and encourages discussion and collaboration. Students may not finish the entire implementation of the programming task, but the goal is to teach them how to use the checklist tool. After a snack break (approximately half an hour), groups reconvene to work on programming task #2 ('Robot finds kitten task' folder) for approximately 2.5 hours using worksheet #2 ('Blank problem-solving worksheet'). In the second half, students practice using the checklist tool more independently to reinforce what they have learned about the problem-solving process.

List of included materials:

Checklist tool – The complete problem-solving checklist; provided to students as a reference guide.

Tic-tac-toe task – A folder containing all the necessary materials for programming task #1; students must implement a hot-seat game of tic-tac-toe in C using the ncurses library; we have provided a library of helpful functions for using ncurses for this task since prior experience with ncurses is not assumed.

Tic-tac-toe problem-solving worksheet – The worksheet was constructed by taking the relevant sections of the problem-solving checklist, providing space for students to write notes, and adding helpful hints and examples relevant to the tic-tac-toe task; students use this worksheet while completing the tic-tac-toe task.

Robot finds kitten task – A folder containing all the necessary materials for programming task #2; students implement a simplified version of the game 'robot finds kitten' (available to play in a browser at <http://www.robotfindskitten.org/>) in C; as before we provide a library of useful functions for implementing this task.

Blank problem-solving worksheet – The problem solving worksheet as before but without any hints or annotations; students use this worksheet while completing the robot finds kitten task.

ACKNOWLEDGEMENTS

The problem-solving checklist tool was developed by Hillary Dawkins, Grant Douglas, Kevin Glover-Netherton, David Hudec, Sean Lunt, Dalton Polhill, Mostafa Rashed, Matthew Sampson, Alliyya Mohammed, James Mosley, Rhys Young, and Dr. Judi McCuaig (University of Guelph). The additional libraries supplied as part of the programming tasks were written by George Chapman-Brown (University of Guelph). We also give credit to the original developers of the robot finds kitten game. The ongoing development of problem-solving workshops is part of the research program of Dr. Judi McCuaig at the University of Guelph, and is supported by an institutional Learning Enhancement Fund. We thank the facilitators and students who participated in the trial workshops and provided valuable feedback.