For Better or For Worse: Is What We Are Doing Right?

Jennifer Wong
Department of Computer Science
University of Victoria
jwong@cs.uvic.ca
http://www.cs.uvic.ca/~jwong

ABSTRACT
Many initiatives have risen in response to an enrolment decrease in Computer Science. However, many questions and challenges still remain unresolved. Initiatives have been designing and deploying activities and pedagogy, but how do we know we are actually recruiting and retaining instead of dejecting students? This poster will address various issues and attempt to answer some of these issues and challenges.

1 INTRODUCTION
While there is a significant increase in enrolment for Canadian university programs, there has been a clear decline in enrolment numbers in the Computer Science field [4]. To respond to this issue, numerous outreach initiatives have been formed and programs have been developed. When these programs are designed and executed, we often focus on teaching through fun and games; moreover, we are also allowing participants to learn through trial and error. However, when we are teaching Computer Science courses at the university level, we focus on teaching and often neglect the fun component of the equation.

On one hand, we are redesigning courses in such a way that it is more appealing and “less hard”; on the other hand, we can also be losing interested students who prefer the traditional method. Are we really retaining or are we dejecting? For this reason, we wish to answer questions such as “are we encouraging or discouraging students when we redesign courses”, “what are the variables that actually motivates students”, “how do we accommodate diversity”, and “how much can we extract from outreach for teaching university level Computer Science”. Are we doing it right?

2 BACKGROUND AND RELATED WORK
While designing fun activities for outreach is important, we must also look at evaluating these activities as well as outreach as a whole. Recent published work by Wong et al. suggested ways to conduct research in outreach evaluation, attempting to determine the success of the outcome [5].

Since the Computer Science field is relatively new, we must rely on information from other disciplines to help us understand what we need and develop suitable activities/pedagogy for the various settings. At the university level, Rao and Mitra have applied software engineering approach when teaching first year Computer Science courses [3] so that students are able to breakdown the problem into smaller chunks before attacking it, while Dodds et al. replaced Java with Python [2] in attempt to offer students, a taste of what Computer Science is about. At the K-12 level, we have used tools such as LEGO Mindstorms NTX as well as Computer Science unplugged activities written by Bell et al. [1] to stimulate interest in the field.
3  APPROACH AND UNIQUENESS

From literature in Computer Science education, we can see that time is mostly spent on finding new ways to teach Computer Science. But how do we know the change is really retaining and recruiting students instead of dejecting students from continuing? Little, if any, research has looked into this. When it comes to Computer Science education, many hours of research is still required to explore and answer some, if not all, of the challenges and questions.

As aforementioned, one of the major questions that we wish to address is whether redesigning a course to make it more appealing or “less hard” actually retains or dejects students. We first need to look into what type of activities attracts different age groups and try to find common patterns. If a pattern exists, then we can develop activities that are suitable for both K-12 and university level, thus, yielding a nice bridge for students who have participated in any of the K-12 outreach programs to smoothly transition into majoring Computer Science. While this can be true, we must also look into creating activities that will accommodate the diversity of students; we need to accommodate students who wish to be taught with the traditional method in order to be further challenged.

Another major challenge that we need to look into is the fact that we wish to teach students to be creative and think outside the box. If we start teaching languages at the beginning, are we already restricting the way students think? Is there a natural way for an untrained human mind to think (i.e. object-oriented or functional)? If we are teaching students to think in a certain way, is it reversible when we wish to teach new concepts/techniques.

Other than attempting to search for answers, we also need to see if our solution can be widely adaptable or not. Often literature reports on what has been done, but often neglects to mention some of the prerequisites for the approaches. If too many prerequisites are required, then others may not be able to easily adopt the method.

4  RESULTS AND CONTRIBUTIONS

At the outset of this study, we are expecting to gain insight into teaching Computer Science to students in the K-12 and at the undergraduate levels. In addition, we can create a general guide for all outreach initiatives and universities such that they can create the best possible programs to accommodate the diversity of people.

5  REFERENCES