

## The Role of Automation in Undergraduate Computer Science Education

Full Text:  PDF  [Get this Article](#)

Author: [Chris Wilcox](#) Colorado State University, Fort Collins, CO, USA

Published in:



- Proceeding

[SIGCSE '15](#) Proceedings of the 46th ACM Technical Symposium on Computer Science Education

Pages 90-95

ACM New York, NY, USA ©2015

[table of contents](#) ISBN: 978-1-4503-2966-8 doi:>[10.1145/2676723.2677226](https://doi.org/10.1145/2676723.2677226)



 2015 Article

### [Bibliometrics](#)

- Downloads (6 Weeks): 10
- Downloads (12 Months): 105
- Downloads (cumulative): 170
- Citation Count: 1



[Recent authors with related interests](#) ▾

[Concepts in this article](#) ▾

powered by

**IBM Watson™**

Presented by: Garrett Dancik

September 17, 2018

<http://dl.acm.org/citation.cfm?id=2677226>

# Article: The Role of Automation in Undergraduate Computer Science

- Background:
  - Grading is a very time-consuming element of teaching
  - Automated grading has advantages
    - Students can submit programs online where they are automatically graded
    - Students receive relatively fast feedback
    - Students can submit multiple times, can make corrections and, ideally, better learn concepts
- Objective: Look at effect of automatic grading (+peer instruction) on learning and other outcomes

The screenshot shows a Mozilla Firefox browser window titled "Checkin - Mozilla Firefox". The address bar shows "CS1 Checkin" and the URL "https://www...". The page content is titled "Foundations in Programming" and includes a navigation menu with links to Home, Syllabus, Policies, Progress, Tutorials, Programs, Resources, and Checkin. A "SERVER TIME" section displays the current time. Below it, three items are listed: "P3: Tax Computation" (status: OPEN), "P2: Numbers and Strings" (status: OPEN), and "P1: Let's Get Started!" (status: CLOSE). Each item has a corresponding icon (red X, green checkmark, orange exclamation mark). Below these items, posted and due dates are listed: "Posted Date: Sunday, 08-03-2014 @ 08:00", "Due Date: Sunday, 08-17-2014 @ 12:00", and "Late Deadline: Sunday, 08-17-2014 @ 22:00". The current submission date is "Friday, 08-15-2014 @ 09:17". A "Submission Status: Present" section shows a "REVIEW" button, a "BROWSE..." field containing "No file selected.", a "SUBMIT" button, and two "DETAILS" buttons. At the bottom, testing and grading information is provided: "Preliminary Testing: 3 / 7" and "Final Grade: Not Graded".

Figure 3: Automated Program Grading Interface.

The screenshot shows a question titled "Classes, Methods, Data" with a coffee cup icon. The question asks: "What does the following code print?". The code is:

```
public class Peer {  
    static int i = 12;  
    int j = 23;  
    public static void main(...) {  
        Peer p1 = new Peer();  
        Peer p2 = new Peer();  
        p1.i++; p1.j = 34; p2.i++; p2.j = 45;  
        System.out.println(p1.i+" "+p1.j+" "+p2.i+" "+p2.j);  
    }  
}
```

Below the code, five options are listed: A. 13 34 14 45, B. 13 45 14 45, C. 14 34 14 45, D. 14 45 14 45, and E. Will not compile. The bottom right corner of the box contains the text "CS 180, Spring Semester 2014".

Figure 4: Peer Instruction Example Question.



| Semester    | Zeros Included | Mean | Median | Sample Size | Stddev | Variance | Increase | Statistical Significance   |
|-------------|----------------|------|--------|-------------|--------|----------|----------|----------------------------|
| Spring 2013 | Yes            | 66.1 | 71.0   | 234         | 20.3   | 410.4    | +0.0%    | base semester              |
| Spring 2013 | No             | 68.5 | 71.0   | 234         | 16.2   | 264.0    | +0.0%    | base semester              |
| Fall 2013   | Yes            | 70.3 | 76.3   | 252         | 24.1   | 582.7    | +6.4%    | $t = 2.07$<br>$p = 0.0388$ |
| Fall 2013   | No             | 75.7 | 79.0   | 252         | 14.7   | 216.3    | +10.6%   | $t = 5.04$<br>$p < 0.0001$ |
| Spring 2014 | Yes            | 73.9 | 78.0   | 232         | 20.5   | 422.3    | +11.8%   | $t = 4.11$<br>$p < 0.0001$ |
| Spring 2014 | No             | 77.6 | 78.0   | 232         | 12.5   | 156.8    | +13.3%   | $t = 6.63$<br>$p < 0.0001$ |

Figure 5: Analysis of Exam Scores.

# Conclusions

- Article contributions: the combined adoption of new pedagogic techniques (automated grading + peer instruction)
  - improves the average final exam scores and overall grade.
  - decreases the student withdrawal rate.
  - improves student attendance.
- Limitations
  - Automated grading and peer instruction are confounded
  - Are students just better over time (uncontrolled extraneous variable)?
- Additional observations
  - Automated grading has an unforgiving nature
  - Some students throw submissions at automated grading

Project idea: development of an automated tool for learning *R* programming

# *R* programming and *swirl*

- *R* (<http://www.r-project.org>) is a free environment for statistical computing and graphics
- *R* is an interpreted language
- Many packages are available for specialized analyses (<http://cran.r-project.org/web/packages/>)
- Swirl (<http://swirlstats.com>) is a package where you can "learn R, in R."
  - Questions are hard-coded
  - This makes *swirl* appropriate for learning but not for practice and/or assessment

# Proposed project

- Develop a swirl-based package that generates template-based problems to help students practice *R* programming and data analysis concepts
- Question templates that use random variable names and/or values:
  - Question: Create a vector named '**x**' that stores the values **3** and **11**.
  - Solution: **x <- c(3,11)**
- Improvement: values in red above are randomly generated each time.

# Questions:

1. What is the best way to learn a new programming language?
2. What is the best way to assess (grade) someone's programming ability?
3. Does anyone have experience creating R packages (how do you do this)?
4. Does anyone know how to generate random numbers in *R*?