# Security Injections 2.0: Increasing Engagement and Faculty Adoption Using Enhanced Secure Coding Modules for Lower-Level Programming Courses

Sagar Raina, Blair Taylor, Siddharth Kaza

# Security Injections 2.0: Increasing Engagement and Faculty Adoption using Enhanced Secure Coding Modules for Lower-level Programming Courses

Sagar Raina[1,*], Blair Taylor[1], and Siddharth Kaza[1]

[1]Department of Computer & Information Sciences, Towson University
sraina1@students.towson.edu, {btaylor,skaza}@towson.edu

**Abstract.** Learning interventions based on modules are common in computer science education. Traditional learning modules that present a large amount of content in a linear format can lead to students skimming and skipping content resulting in lower student engagement and effectiveness. In this paper, we present theoretical support for increasing engagement and effectiveness of learning modules, describe a system that implements these principles, and discuss the results of a study across four sections of CS0. Using the Security Injections @Towson cybersecurity modules, we enhanced select modules by incorporating the e-learning design principles of segmentation and interactivity. The study compares student engagement between the current (1.0) modules and the enhanced (2.0) modules. The use of the enhanced (2.0) modules significantly increased student engagement and these results persisted across gender and race. Feedback from instructors indicates higher student and instructor interest in the enhanced modules; in spring 2015, more than 20 instructors are using the enhanced (2.0) modules.

**Keywords:** integer overflow, buffer overflow, input validation, cs0, learning sciences, interactive learning modules, instant-feedback, auto-grading.

## 1 Introduction

The recent focus on cybersecurity education has led to the development of cybersecurity learning materials across various academic institutions and organizations in the United States [19]. Among such initiatives are web-based learning modules developed by the Security Injections @Towson project [5, 13, 17]. These learning modules target key secure coding concepts including integer error, buffer overflow, and input validation in various programming languages, for Computer Science 0 (CS0), Computer Science 1 (CS1), and Computer Science 2 (CS2); and general security concepts, such as phishing, passwords, and cryptography for use in Computer Literacy courses. The Security Injections @Towson modules have been found to be highly effective at improving security awareness and the ability to apply secure coding concepts [13]. In over six years of dissemination to over 150 institutions, the following issues have

been observed: 1) instructors noted that students tended to skip content and proceed directly to lab exercises, and 2) large-scale adoption by instructors remains a challenge.

The modules originally developed in the project are web-based learning modules that follow a traditional linear format (we refer to these as 1.0 modules hereafter). This type of module design, that presents a large amount of content at one time, is common in many disciplines, but can lead to issues like skimming and skipping [11, 16]. In this paper, we discuss the key issues that lead to skipping and skimming, including the amount of content presented on a screen and decreased student engagement. Presenting less content allows students to process information more easily and improves learning; increasing student engagement motivates students to learn [1, 2, 9, 11, 16, 18, 20]. We describe enhanced modules, Security Injections 2.0, that address these issues by utilizing segmentation and interactivity, and a study that tests the engagement of the new modules. In addition, the enhanced modules include an auto-grading functionality to facilitate easier instructor adoption.

To assess the effectiveness of enhanced learning modules towards student engagement and instructor adoption, we compare the Security Injections 1.0 modules to 2.0 and explore the following questions:

**Q1:** How can we improve the traditional web-based learning modules to reduce content skipping, increase engagement and instructor adoption?

**Q2:** Can the use of 2.0 modules with segmentation and interactivity increase student engagement compared to the 1.0 modules?

**Q3:** Can the use of 2.0 modules with auto-grading increase instructor adoption compared to the 1.0 modules?

## 2 Literature Review

Web-based learning modules present content using hypertext. Individuals reading hypertext have a tendency to skip or skim the content [14]. In this section, we discuss: 1) why hypertext readers skip or skim the content, and 2) how can we improve interactivity in learning modules to engage learners.

### 2.1 Hypertext reading and content skipping

Hypertext is text that contains links to text, audio, video, or graphics in other documents or media resources, giving flexibility to readers to click any of the links to gain knowledge [3]. Hypertext readers often adopt a reading strategy which determines what to read and what to skim [3, 7]. This may lead more selective reading for lengthy documents. Readers who skim have a tendency to read the first half of the paragraph and if they determine that the gain of information is low, they may skip the other half of the paragraph and jump to the next paragraph [4]. This skipping may lead readers to lose important information [4, 11]. Selective reading may also lead to less in-depth reading, less concentration and attention towards the content [8]. In addition, skipping of the content leads to poor learning [12].

Research suggests that instead of presenting large amounts of hypertext content at once, the content should be broken into smaller chunks and presented one idea at a time on a single screen [1, 2, 9]. This concept is referred to as segmentation. Segmentation improves processing of information in the working memory and makes recalling and retention of concepts easier [9].

### 2.2 Improving interactivity in learning modules

Interactivity in e-learning is the "responsiveness to the learner's actions during learning" [9, 18]. Interactivity improves engagement and motivates students to learn [18]. The types of interactivity in e-learning environments include: dialoguing, controlling, manipulating, searching and navigating [9]. In this paper, we focus on dialoguing and controlling.

*Increasing interactivity with Dialoguing* - Dialoguing occurs when the learner answers questions and receives feedback to his/her input. Dialoguing has been considered beneficial, as learners can relate the feedback to the current content [15]. In e-learning, dialoguing can be implemented using assessments (like formative questions) with appropriate feedback [15]. This assessment can be conducted using multiple-choice, fill-in-the-blank, short answer and essays formats. The feedback provided on the answers can be either immediate or delayed. The feedback provided can also be classified based on the amount of detail provided on the answers [6, 15] and is categorized as: 1) *Knowledge of Results (KR)* , 2) *Knowledge of Correct Response (KCR)* and, 3) *Elaborate Feedback (EF)*.

Knowledge of results (KR) informs the learner if their answer is correct or incorrect; knowledge of correct response (KCR) informs if the answer is correct or incorrect and includes the correct answer; and elaborate feedback (EF) informs if the answer is correct or incorrect, includes the correct answer, and also includes a concise explanation of the correct answer.

We plan to implement dialoguing in 2.0 modules using Multiple Choice Questions (MCQs), true or false and constructed response (CR) type of assessments with immediate knowledge of results (KR) and immediate elaborate feedback (EF).

*Increasing interactivity with controlling* - Controlling implies that the learner can determine the pace of the presentation. Controlling helps students learn better by allowing them to process information at their own pace [2, 9]. We plan to implement controlling in 2.0 modules using answer-until-correct [15] with immediate knowledge of results (KR) and immediate elaborate feedback (EF) to MCQs, true or false and constructed response (CR) type of assessments.

Segmentation breaks large content into smaller chunks and presents them one at a time which may result in less reading and less skipping of content. Less skipping of content leads to increased learning [12]. Interactivity (dialoging and controlling) on segmented chunks leads to engagement and enforces learning (see Fig.1.).

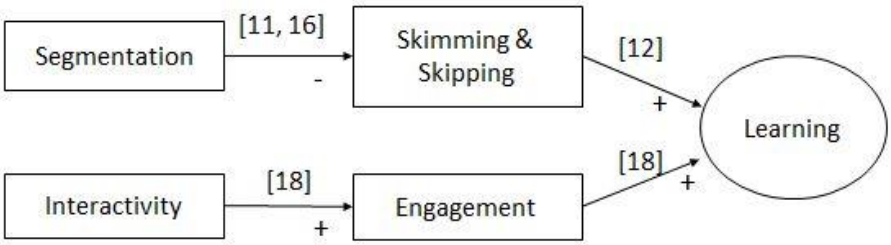



**Fig. 1.** Literature suggests that segmentation and interactivity in modules may increase learning

## 3 Incorporating Segmentation, Instant-feedback and Auto-grading in Security Injection Modules

The Security Injection learning modules 1.0 were developed on the cognitive learning principles of Bloom's taxonomy and adopt a uniform structure. Each module begins with a background section to describe the problem with examples, followed by a "Code Responsibly" section (that includes methods to avoid security issues), a laboratory assignment with a security checklist, and discussion questions. The module structure is designed to help students to first understand the problem through the background and code responsibly sections, remember it through the laboratory assignments, evaluate it through checklists, and apply the concepts learned through discussion questions.

In the 2.0 modules, we enhanced the 1.0 modules by applying segmentation to reduce skipping of content, and interactivity, using dialoging and controlling, to improve student engagement. We implemented segmentation by breaking up the module content per section (background, code responsibly, laboratory assignment, discussion questions) and presenting each section, one at a time, on the screen (see Fig.2.). We implemented dialoguing using formative assessment including true or false, multiple choice and constructed response that include immediate knowledge of results (KR) or elaborate feedback (EF). We implemented controlling using answer-until-correct giving the students control over the number of attempts. Students receive immediate knowledge of results (KR) feedback until third attempt and immediate elaborate feedback (EF) thereafter.

*Module Design* - In the background and code responsibly sections, students are required to go through the content and answer a set of checkpoint questions. Each question provides immediate feedback on submit (see Fig.3. and Fig.4.). The student cannot advance to the next section until all questions are answered correctly. In the laboratory assignment and discussion question, students answer text-based, multiple choice questions, and identify vulnerabilities based on a security checklist (see Fig.5.). These are also auto-graded.


**Fig. 2.** Modules broken into sections


**Fig. 3.** Instant-feedback on incorrect answer

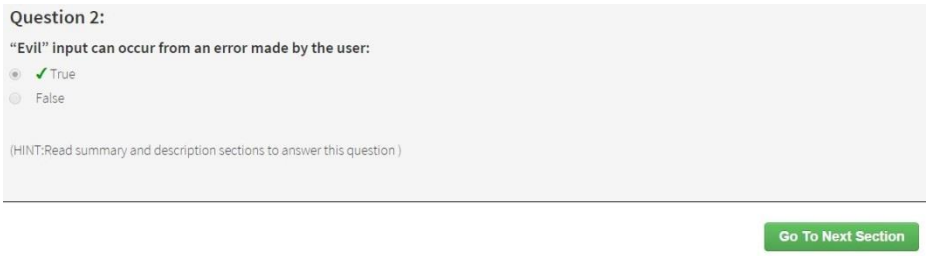
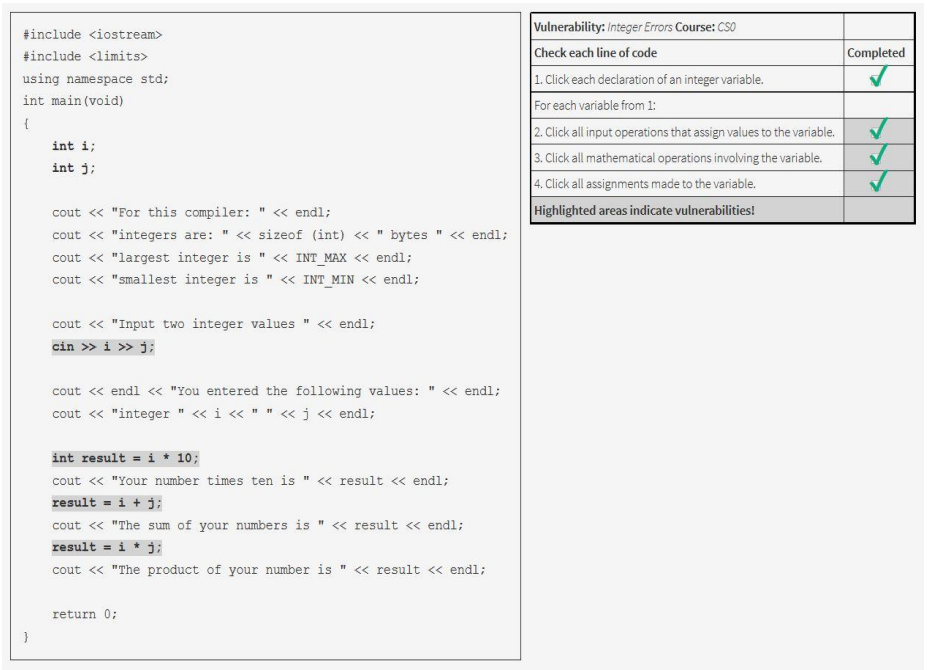



**Fig. 4.** Instant-feedback on correct answer



```
#include <iostream>
#include <limits>
using namespace std;
int main(void)
{
    int i;
    int j;

    cout << "For this compiler: " << endl;
    cout << "integers are: " << sizeof (int) << " bytes " << endl;
    cout << "largest integer is " << INT_MAX << endl;
    cout << "smallest integer is " << INT_MIN << endl;

    cout << "Input two integer values " << endl;
    cin >> i >> j;

    cout << endl << "You entered the following values: " << endl;
    cout << "integer " << i << " " << j << endl;

    int result = i * 10;
    cout << "Your number times ten is " << result << endl;
    result = i + j;
    cout << "The sum of your numbers is " << result << endl;
    result = i * j;
    cout << "The product of your number is " << result << endl;

    return 0;
}
```

| **Vulnerability:** *Integer Errors* **Course:** *CS0* | |
|---|---|
| **Check each line of code** | **Completed** |
| 1. Click each declaration of an integer variable. | ✓ |
| For each variable from 1: | |
| 2. Click all input operations that assign values to the variable. | ✓ |
| 3. Click all mathematical operations involving the variable. | ✓ |
| 4. Click all assignments made to the variable. | ✓ |
| **Highlighted areas indicate vulnerabilities!** | |

**Fig. 5.** Auto-graded security checklist

*System Implementation* - To implement the enhanced modules, several solutions were considered (including writing the system from scratch) before determining that a modified version of Stanford University's class2go web-based application (https://github.com/Stanford-Online/class2go/) was most appropriate. Class2go is built using the Django framework. Class2go is an open-source framework that provides core functionality, including user registration, course creation, test administration, and some components for auto-grading. We wrote code to auto-grade the security checklist, and regular expressions that match keywords to verify answers for short-answer questions (constructed response). In addition, we developed an instructor dashboard with a progress grade-book to monitor student progress (see Fig.6.).

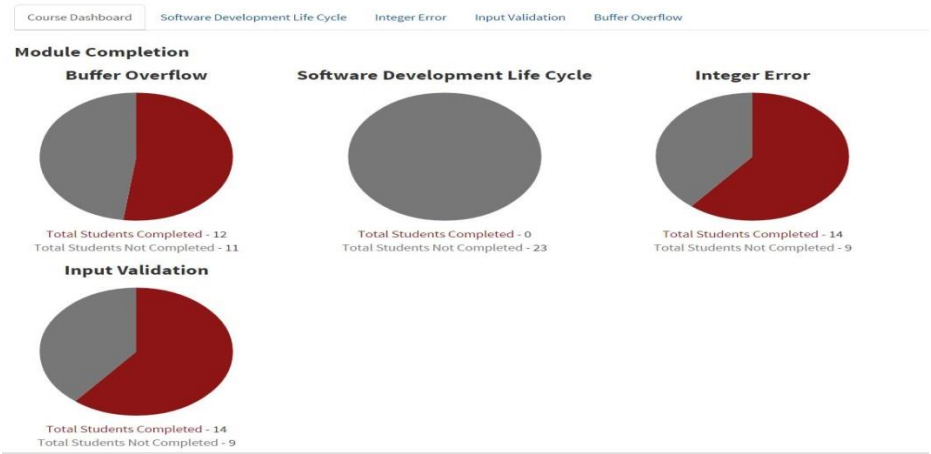



**Fig. 6.** Instructor grade-book

# 4 Study

## 4.1 Methodology

A quasi-experimental study was conducted in fall 2014 across four sections of a CS0 course (using C++) at a large public university, using a posttest only control group design. Two of the sections used the 1.0 version (control group) and the other used 2.0 (treatment group). The study was conducted during the laboratory sessions which were at different times for each section. Three modules - integer error, input validation and buffer overflow - were introduced, in that order, with approximately four weeks between the interventions. Both groups were administered a student engagement survey at the end of the semester.

The survey instrument used in this study measured student demographics and student engagement. While questions related to student demographics were derived from previous security injection studies [13], which measured students' gender, age-group, ethnicity and major, we adapted a set of eight item questions from a well-tested User Engagement Scale (UES) [10] to measure student engagement. The eight item student engagement questions were recorded on a five point Likert scale. (See Table 1 for sample survey questions.)

Based on the survey scores, we proposed the following hypothesis to compare Security Injections 1.0 and Security Injections 2.0 (treatment group) on the following dependent variable: student engagement score.

H1: The mean of survey scores for student engagement in the treatment group will be significantly higher than the mean of the survey scores for student engagement in the control group.

**Table 1.** Survey questions

| | **Student engagement** |
|---|---|
| Q1 | I felt deeply engrossed while completing security injection modules using this web-based platform. |
| Q2 | I get so involved while completing security injection modules using this web-based platform that I forget everything. |
| Q3 | While completing the security injection modules using this web-based platform, I tend to block out conversations with others around me. |
| Q4 | The Security Injection modules presented on this platform hold my attention. |
| Q5 | Using this web-based platform excited my curiosity to learn cybersecurity principles. |
| Q6 | Time seemed to go by very quickly when I use this web-based platform for completing Security Injection module. |
| Q7 | The screen layout of this web-based platform for Security Injection modules was visually pleasing. |
| Q8 | Using this web-based platform for completing Security Injection modules was attractive. |

### 4.2 Initial Results

A total of 116 students participated in the study. After filtering missing data and outliers, 80 (42 in the treatment group and 38 in the control group) students including 54 (29 in the treatment group and 25 in the control group) males and 26 (13 in the treatment group and 13 in the control group) females completed the survey. For student engagement, each response was assigned codes from 1 to 5, on a five point Likert scale, 1 representing 'strongly disagree' and 5 representing 'strongly agree'. The engagement score for each respondent were calculated as the mean of codes for eight questions. The Cronbach's alpha, for eight-item engagement questions, was found to be 0.74, which suggested good internal consistency. The hypothesis was tested using independent samples t-test. We picked independent samples t-test because Shapiro-Wilk test showed that scores for student engagement in both the groups (treatment n=42, control n=38) satisfied the conditions of normal distribution (treatment p=.593, control p=.187) and homogeneity of variance ($F=2.554$, $p=.114 > 0.05$).

*Comparison of survey scores for student engagement in treatment and control groups*. In the survey results, the mean score for the treatment group (n=42, mean=3.43) was found to be significantly higher at 95% level ($t=-2.265$, $p=0.026$) than the mean score for the control group (n=38, mean=3.19). This implies that students found Security Injections 2.0 more engaging than Security Injections 1.0 (see Fig.6a.). This leads us to accept H1 and supports research question Q2. In addition, higher engagement persisted across gender (see Fig.6b.) and race (see Fig.6c.).

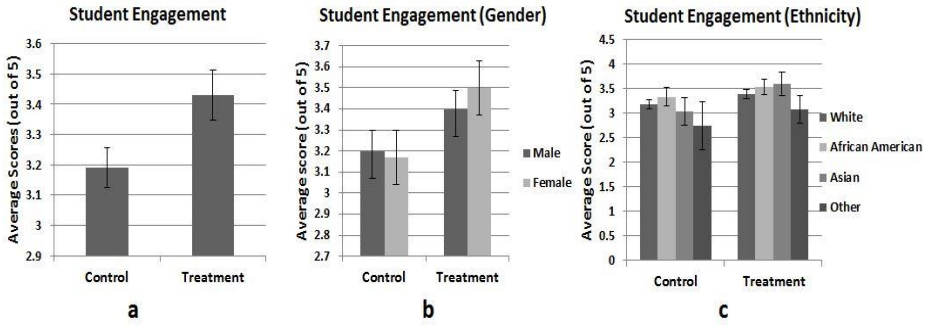



**Fig. 6.** (a) Average student engagement score in treatment and control group (b) Average student engagement score between males and females in treatment and control group (c) Average student engagement score between ethnic groups in treatment and control group

*Comparing results for individual survey questions between control and treatment groups.*

The student engagement mean score for the treatment group was found to be higher than the control group (see Table 2). In particular, the scores for Q6 and Q8 were found to be statistically significant at 95% level (refer to Table 1 for survey questions).

**Table 2.** Results of individual survey questions

| | **Mean Score** | |
|---|---|---|
| | **Control (n = 38)** | **Treatment (n = 42)** |
| Q1 | 3.39 | 3.48 |
| Q2 | 2.63 | 2.74 |
| Q3 | 2.89 | 3.07 |
| Q4 | 3.37 | 3.45 |
| Q5 | 3.34 | 3.33 |
| Q6 | 3.08 | 3.79* |
| Q7 | 3.50 | 3.79 |
| Q8 | 3.29 | 3.79* |
| Student Engagement Mean Score | **3.19** | **3.43*** |
| *p < 0.05 (statistically significant at 95% level) | | |

## 5 Instructor Feedback

Security Injections 2.0 modules were introduced to over 50 faculty members in workshops conducted last year. In the past year, five instructors have used the beta system for both in-class and online instruction to approximately 180 students. Introduction of the modules at the workshops has led to overwhelmingly positive response among instructors who look forward to incorporating the 2.0 modules in their curriculum,

primarily due to the auto-grading functionality. One workshop attendee indicated, "I am excited about the interactive modules (injection 2.0). I will be more likely to incorporate these modules." An instructor who used both versions of Security Injections modules in a class of 30 students in summer 2014 commented, "The students were able to access the materials easily, and seemed to enjoy the interactive modules more. They were also able to get feedback faster, and seemed more comfortable with it compared to the other version." In spring 2015, approximately 20 instructors are using 2.0 modules. Security Injections 2.0 modules are developed on the principles of learning theory and with the intention of facilitating wide-spread instructor adoption. Initial feedback indicates that segmented and instant-feedback based modules are likely to increase instructor adoption and student interest, which supports our initial result and third research question.

## 6 Conclusion

In this study, we designed a system of enhanced secure coding learning modules that implement the principles of segmentation; by breaking content from 1.0 into individual sections per screen with checkpoint questions, and interactivity; by including instant-feedback. We conducted a study to test their effectiveness towards increasing student engagement across four sections of CS0 using the post only control group design. In addition, we collected instructor feedback.

Results from the study showed the segmented and interactive 2.0 modules led to significant increase in student engagement. Additionally, higher engagement persisted across gender and race. Feedback from instructors indicates both higher student and instructor interest. In addition, the increase in use of enhanced modules by instructors indicated higher instructor adoption.

Due to the limited instruments available to assess whether segmentation leads to less skipping and skimming, we plan an observational study in fall 2015 to record student behavior using segmented and interactive modules.

In addition, in future work, we plan a full scale experiment across multiple sections and courses, to examine the effectiveness of the enhanced modules, towards our goal of increasing secure coding knowledge among computer science students.

## Acknowledgements

Class2go (https://github.com/Stanford-Online/class2go/). This project is partially supported by NSF DUE-1241738.

## References

1. Al-Samarraie, H. et al.: Can structured representation enhance students' thinking skills for better understanding of E-learning content? Comput. Educ. 69, 463–473 (2013).

2. Clark, R.C., Mayer, R.E.: e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning (Google eBook). John Wiley & Sons (2011).
3. DeStefano, D., LeFevre, J.-A.: Cognitive load in hypertext reading: A review. Comput. Human Behav. 23, 3, 1616–1641 (2007).
4. Duggan, G.B., Payne, S.J.: Skim Reading by Satisficing: Evidence from Eye Tracking. CHI 2011. , Vancouver, BC, Cananda (2011).
5. Kaza, S. et al.: Injecting Security in the Curriculum : Experiences in Effective Dissemination and Assessment Design. The Colloquium for Information Systems Security Education (CISSE). (2010).
6. Van der Kleij, F.M. et al.: Effects of feedback in a computer-based assessment for learning. Comput. Educ. 58, 1, 263–272 (2012).
7. Lawless, K.A. et al.: Knowledge, Interest, Recall and Navigation: A Look at Hypertext Processing. J. Lit. Res. 35, 911–934 (2003).
8. Liu, Z.: Reading behavior in the digital environment: Changes in reading behavior over the past ten years. J. Doc. 61, 6, 700–712 (2005).
9. Moreno, R., Mayer, R.: Interactive Multimodal Learning Environments. Educ. Psychol. Rev. 19, 3, 309–326 (2007).
10. O'Brien, H.L., Toms, E.G.: The development and evaluation of a survey to measure user engagement. J. Am. Soc. Inf. Sci. Technol. 61, 1, 50–69 (2010).
11. Protopsaltis, A., Bouk, V.: Towards a Hypertext Reading/Comprehension Model, http://delivery.acm.org/10.1145/1090000/1085349/p159-protopsaltis.pdf.
12. Rudestam, K.E., Schoenholtz-Read, J. eds: Handbook of Online Learning. SAGE Publications (2010).
13. Taylor, B., Kaza, S.: Security injections: modules to help students remember, understand, and apply secure coding techniques. Proceedings of the 16th annual joint conference on Innovation and technology in computer science education - ITiCSE '11. p. 3 ACM Press, New York, New York, USA (2011).
14. Taylor, J.S.: Using the World Wide Web in Undergraduate Geographic Education: Potentials and Pitfalls. J. Geog. 99, 1, 11–22 (2000).
15. Thalheimer, W.: Providing Learners with Feedback—Part 1: Research-based recommendations for training, education, and e-learning. , Somerville, Massachusetts, USA (2008).
16. Tseng, M.: The Difficulties That EFL Learners Have with Reading Text on the Web. The Internet TESLJournal. 14, 2, (2008).
17. Turner, C.F. et al.: Security in Computer Literacy- A Model for Design, Dissemination, and Assessment. Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11. p. 15 ACM Press, New York, New York, USA (2011).
18. Zhang, D.: Interactive Multimedia-Based E-Learning: A Study of Effectiveness. (2010).
19. Curriculum Resources - Teaching Tools for Educators, http://niccs.us-cert.gov/education/curriculum-resources.
20. Handbook of Online Learning. SAGE Publications (2010).