3-22-2019

College Student Perceptions of MyProgrammingLab and BlueJ in an Introductory Computing Course

Christine Wania
The College at Brockport, State University of New York, cwania@brockport.edu

COLLEGE STUDENT PERCEPTIONS OF MYPROGRAMMINGLAB AND BLUEJ IN AN INTRODUCTORY COMPUTING COURSE

Christine E. Wania
The College at Brockport, State University of New York
cwania@brockport.edu

ABSTRACT

Students in introductory computing courses face various challenges. Many learning systems are available to support teaching and learning in introductory computing courses. Empirical work examining the use of such learning systems is available, but limited. In this research, we gathered student perceptions of two learning systems MyProgrammingLab and BlueJ. Understanding student perceptions of learning systems and their impact on learning to program is valuable information for both instructors and students. In this analysis, we gathered student perceptions of MyProgrammingLab and BlueJ in three surveys towards the end of a 15-week semester. Although students encountered problems in MyProgrammingLab and BlueJ, more than three quarters of the students perceived MyProgrammingLab and BlueJ to be useful in helping develop their programming skills. Many students agreed that using MyProgrammingLab and BlueJ helped them better understand the course materials.

Keywords

Computer science and information systems education, BlueJ, educational tools, engaged learning, Java, MyProgrammingLab, online learning systems, programming education, smart learning tools

INTRODUCTION

Students in introductory computing courses face many challenges including, but not limited to, learning new problem solving approaches and the syntax and semantics of a programming language (Lahitnen, 2005; Norman and Adams, 2015; Piteira and Costa, 2013; Robins Rountree, and Rountree, 2003). It is no surprise that programming can be difficult for novice programmers. Introductory programming courses have many known issues, such as, high drop rates, low grades, high student stress, and unfortunately, sometimes academic dishonesty. Much has been written about the issues in introductory programming courses (Barr and Trytten, 2016; Ni, Bishop and Donaldson, 2013; Olsson and Mozelius, 2016; Piteira and Costa, 2013). Many learning systems are available to support the teaching and learning of programming. Several systems are free, such as BlueJ, Coderunner, Codingbat, Problets, and PythonTutor, while others such as MyProgrammingLab (Pearson) and Codelab by Turing's Craft charge a fee. There are many potential advantages and some disadvantages of such systems. Empirical work in this area is limited, likely due to the constraints in collecting comparative data in an educational setting. Understanding student perceptions in regards to the assistance of MyProgrammingLab and BlueJ in learning to program is valuable information for both instructors and students. Both MyProgrammingLab and BlueJ are programs designed to support students as they learn to program. The Computing Sciences Department at our institution selected MyProgrammingLab and BlueJ for use in all the introductory programming courses. After the adoption of these tools, students and instructors raised several concerns, specifically about MyProgrammingLab. The instructors, while aware of the potential advantages of MyProgrammingLab, questioned the use, due to some of the known disadvantages and based on early student feedback. Students expressed concerns with the cost of MyProgrammingLab, the wording of some exercises and their interpretations of the feedback. This research attempts to further our understanding of student perceptions of MyProgrammingLab and BlueJ. In this research, we collected student feedback in three surveys during weeks 10 through 15 of a semester. The next section includes a brief summary of related work.

RELATED WORK

Learning to program is challenging. Many systems and tools are available to support learning in this area including BlueJ, Coderunner, Codingbat, Problets, PythonTutor, MyProgrammingLab (Pearson) and Codelab by Turing's Craft. There are many advantages and some disadvantages of such systems. Some of the benefits discussed in the literature include engaged learning (Barr and Trytten, 2016; Prince, 2004), timely feedback (Char, 2016), and, in some studies, improvement in student performance (Allen, Vahid, Downey, and Edgcomb, 2018; Austin, Hortsmann and Vu, 2018; Norman and Adams, 2015). Other studies showed little or no improvement in student performance (Ni et al., 2013). Student perceptions of such systems are varied (Benotti, Aloi, Bulgarelli and Gomez, 2018; Hagan and Markham, 2000; Kumar, 2005; Ni et al., 2013; Williams, Bialac, Liu, 2006). Disadvantages discussed in the literature include cost, ease of plagiarism, lack of flexibility, unhelpful feedback, and ambiguous exercises (Barr and Trytten, 2016; Brusilovsky et al., 2014; Ni et al., 2013; Olsson and Mozelius, 2016). Student

perceptions of BlueJ indicate that they have difficulty installing the software and that compiler error messages are not very helpful (Hagan and Markham, 2000). Student perceptions of MyProgrammingLab indicate the feedback received in MyProgrammingLab is not always helpful (Olsson and Mozelius, 2016). There is little direct evidence of the impact of MyProgrammingLab on student success (Norman and Adams, 2015). However, there is some empirical work suggesting students and faculty perceive that the learning systems are useful (Benotti et al., 2018; Hagan and Markham, 2000; Olsson and Mozelius, 2016).

MyProgrammingLab and BlueJ

This section contains a brief overview of MyProgrammingLab and BlueJ. MyProgrammingLab is a Pearson product that uses Turing's Craft Codelab. MyProgrammingLab is designed to help students learn programming fundamentals. MyProgrammingLab is available for use with or without a particular textbook. At this time, registration/access to MyProgrammingLab is $44.95 or it can be bundled with the electronic course textbook selected by our department for $96.95. The selection of another textbook changes the cost. MyProgrammingLab provides quite a bit of flexibility for instructors. Instructors can select from the available programing exercises or add additional exercises. Instructors can select a view date, a due date, a solution date and a freeze date for all exercises. In addition to the selections already mentioned, MyProgrammingLab allows instructors to configure a maximum number of attempts for programming exercises. Instructors can view student solutions and their overall progress. MyProgrammingLab allows for rather extensive modification. Students can view the assigned programming exercises according to the view dates assigned by the instructor. Instructors may choose to expose all exercises at the beginning of the semester or gradually reveal exercises as the course progresses. Students may view and submit programming exercises according to the configurations specified by the instructor. After students submit solutions, they receive immediate feedback. If a solution submitted by a student is correct, the student receives immediate feedback indicating the response is correct. If a solution submitted by a student is incorrect, the student receives immediate feedback that attempts to identify the error(s). For example, feedback may identify syntax errors and when appropriate the feedback includes links to additional information that may assist the student. Almost all correct solutions are accepted and marked correct. Some atypical solutions may be marked incorrect and some extraneous characters may cause a solution to be marked incorrect. Instructors can view student attempts and perform an override to mark attempts correct, if necessary.

BlueJ is a free Java development environment. BlueJ is based on the Blue system (Kölling, 1999). BlueJ uses a standard compiler and virtual machine. BlueJ is designed for beginner programmers. It was designed with several goals: to focus on an object-oriented approach, to allow interaction with single objects, and to simplify the user interface in order to allow users to focus on the principles of programming (Kölling, Quig, Patterson, and Rosenberg, 2003; Kölling, 2008). Although the creators of BlueJ designed it to assist with an object-oriented approach, it is used by many without an objects first approach, mainly due to the simplified interface (Kölling, et al. 2003; Kölling, 2008). In BlueJ, students see a UML class diagram for a particular project. This allows for a unique and simple interaction with classes, objects, and methods. BlueJ also color codes related portions of a program to assist students. The simplified editor interface allows students to focus on the principles of programming. Reducing the complexity in the interface of the development environment is of great benefit to students who may already be overwhelmed with learning to program.

BACKGROUND

The students described in this work were enrolled in a four credit Introduction to Computing course at a comprehensive liberal arts college. The Computing Sciences Department has two ABET accredited programs: Computer Science and Computer Information System. The course was recently revised due to updates in ABET requirements and updates in the state university system requirements. The course was previously a breadth first approach to computing. The course is now an introductory programming course. The fall 2018 semester was the first semester teaching the newly revised course. The department selected and required the use of both MyProgrammingLab, BlueJ and a common textbook in each section of this course to maintain consistency. Although this course is taught with a procedural-first approach, and not with an objects-first approach, the department selected BlueJ due to the simplified interface. The department had also used MyProgrammingLab several years ago and decided to once again to use MyProgrammingLab. This course is offered in both fall and spring semesters each academic year. Typical enrollment is between 25 and 35 students and consists of students who are required to take the course as a prerequisite to the Computer Science and Information Systems programs, students who are required to take the course for another major, and students who choose to take the course as an elective.

The course consists of a 2-hour lab session and 2.5 hours of class instruction per week. The course is a procedural-first approach with Java. An instructor who had taught the course or a similar programming course more than ten times taught the course. The assigned textbook for the course is titled *Introduction to Java Programming and Data Structures* by Liang (2017). The

instructor for this course posted weekly summaries of each module on Blackboard. The summaries were a concise overview of the material. During class sessions, the instructor discussed the material, demonstrated the material and facilitated in class activities. Although the class sessions are in a traditional classroom, not in a lab setting, the sessions are very much engaged learning also. Programs or portions of programs, written in lab sessions and class sessions as demonstrations by the instructor were written in BlueJ. There were six programming homework assignments, two quizzes, two exams and a final exam in this course. In this course, homework assignments were completed on paper and in BlueJ.

The instructor developed the course material and carefully selected the exercises in MyProgrammingLab. Each week lab exercises related to the current course module were assigned in MyProgrammingLab and in BlueJ. Two hundred eighty-eight exercises were assigned in MyProgrammingLab. Most weeks there were additional lab exercises completed by students on paper and in other online systems or websites. Lab exercises also included correcting incorrect problems from homework assignments, quizzes and exams. Each MyProgrammingLab exercise requires that the student write some Java code that might include an expression, a method, a selection statement or repetition statement. There are also several exercises related to course vocabulary and data representation. The exercises completed at the beginning of the semester had very short answers, whereas the exercises completed in the second half of the semester including several lines of code. The instructor was present in lab. There was also a lab assistant, who was a senior level student, present in lab sessions to help answer student questions. In the first lab session of the semester MyProgrammingLab and BlueJ were demonstrated by the instructor and subsequently used by the students. Several warm up exercises were assigned in both MyProgrammingLab and BlueJ. At the beginning of lab sessions, the instructor briefly reviewed the relevant course material and provided an opportunity for students to ask any questions about the material. The students were able to view the MyProgrammingLab exercises assigned for each week at the beginning of the weekly lab session. Students were given the entire lab session to complete the assigned lab exercises. While many students completed all or most of the exercises during the allotted lab time, others did not. Therefore, because the goal is to allow students to practice and learn, students were given until the end of that week to complete the lab exercises. In the first few weeks of the semester, students had several questions and complaints about MyProgrammingLab. The questions and complaints related mostly to the wording of some of the exercises and the feedback provided by MyProgrammingLab for incorrect submissions.

METHOD

Students completed three surveys during lab sessions in the tenth, eleventh and fifteenth weeks of a semester. The surveys gathered information about the students' major, previous programming experience, previous experience with the selected systems, and perceptions on several items including: ease of use of the systems, assistance in learning programming concepts, assistance in correcting errors, and usefulness of feedback, among other things. The survey questions were derived from a variety of sources (Austin et al., 2018; Hagan and Markham, 2000), with some questions relating to perceived ease of use (Davis et al. 1989). Many questions contained Likert-type responses (strongly disagree, disagree, neither disagree nor agree, agree and strongly agree). Questions on the three surveys were very similar, in essence, gathering perceptions on the same items for each system. For example, question eight on the MyProgrammingLab survey stated, "I enjoy completing the exercises in MyProgrammingLab" and question eight on the BlueJ survey stated, "I enjoy writing programs in BlueJ." The first survey focused on MyProgrammingLab, the second survey focused on BlueJ and the last survey focused on both systems and the course as a whole. Results are described in the following section.

RESULTS

Of the twenty-seven students enrolled in the course, twenty-five students completed the MyProgrammingLab survey, twenty-five students completed the BlueJ survey, and twenty-two students completed the end of the semester survey. Of the twenty-five students who completed both the MyProgrammingLab survey and the BlueJ survey, twenty-three were computer science majors, two were computer information systems majors, and three of the students had more than one major. Nineteen of the students were freshman (76%), one student was a sophomore (4%), and five students were juniors (20%). Of the twenty-five students, twenty-four were male students. Approximately half of the students, thirteen of twenty-five, had some previous programming experience. None of the students had used MyProgrammingLab prior to this course and only one student had used BlueJ prior to this course. All students enrolled in the course purchased access to MyProgrammingLab. Two students purchased access after the second week of the semester due to financial constraints. All students enrolled in the course attempted and completed some of the assigned exercises in MyProgrammingLab. Some students attempted and completed all of the assigned exercises. Seven students correctly completed all of the MyProgrammingLab exercises. Many students completed the exercises correctly with one to a few attempts. The highest number of attempts for an exercise was 33 attempts by one student. The average number of attempts per exercise was 2.3 attempts.

Student perceptions of MyProgrammingLab and BlueJ varied. More than three quarters of the students agreed that both MyProgrammingLab and BlueJ were useful in helping them develop their programming skills, twenty-one students (84%) and twenty-one students (84%) respectively. More than three quarters of the students, twenty-one students (84%) felt as though the MyProgrammingLab exercises helped them understand the course material and sixteen students (64%) felt as though the BlueJ exercises and assignments helped them understand the course material. Most students felt as though the MyProgrammingLab exercises helped them complete their homework assignments, eighteen students (72%) agreed, five students (20%) neither agreed nor disagreed, and two students (8%) disagreed. More students, fourteen students (56%), agreed that writing programs in BlueJ helped them study for quizzes and exams, compared to eight students (32%) for MyProgrammingLab. Many students were unsure or disagreed. About half of the students, thirteen students (52%), agreed that it was easy to use MyProgrammingLab. However, nine students (36%) were unsure and three students (12%) did not feel as though it was easy to use MyProgrammingLab. More than three-quarters of the students, twenty students (80%) agreed that it was easy to use BlueJ. One student was unsure, and four students did not feel as though it was easy to use BlueJ. Many students agreed that both MyProgrammingLab and BlueJ helped them identify and correct their errors, fifteen students (60%) and eighteen students (72%) respectively, as seen in table 1. Several students found the feedback in MyProgrammingLab and BlueJ to be helpful, sixteen students (64%) and fourteen students (56%), correspondingly. As seen in table 1, more students felt that BlueJ helped them identify and correct their errors compared to MyProgrammingLab.

	Disagree	**Neither**	**Agree**
MyProgrammingLab helps me identify and correct my errors	16%	24%	60%
BlueJ helps me identify and correct my errors	8%	16%	72%
The feedback I receive in MyProgrammingLab is helpful	12%	24%	64%
The feedback I receive in BlueJ is helpful	4%	40%	56%

Table 1. Perceptions of Feedback Received in MyProgrammingLab and BlueJ

Eleven students (44%) enjoyed completing the exercises in MyProgrammingLab and six (24%) students did not enjoy completing the exercises in MyProgrammingLab. More students enjoyed writing their programs in BlueJ, compared to completing the exercises in MyProgrammingLab. Sixteen students (64%) enjoyed writing programs in BlueJ and only two students indicated that they did not enjoy writing programs in BlueJ. However, eighteen students (72%) indicated that they felt as though the MyProgrammingLab exercises were helping them learn to write Java programs. Only one student disagreed, and six students (24%) were not sure. Students were also asked if they would like to use MyProgrammingLab and BlueJ in other programming courses. About half of the students expressed interest in using MyProgrammingLab and BlueJ in other programming courses, as seen in table 2.

	Disagree	**Neither**	**Agree**
I would like to use MyProgrammingLab in other programming courses	20%	32%	48%
I would like to use BlueJ in other programming courses	20%	28%	52%

Table 2. Preferences of Future Use of MyProgrammingLab and BlueJ

At the end of the MyProgrammingLab survey students were also asked to provide additional written comments about MyProgrammingLab. Ten students did not write any comments. Student comments about MyProgrammingLab ranged from the 'instant feedback is useful' to 'wants a very precise answer which is annoying but understandable.' Seven students commented on how 'picky' MyProgrammingLab can be with requiring exact spacing, naming of variables, and in some cases particular solutions to problems. Five students commented that overall they thought MyProgrammingLab was useful and helpful in learning to program. Four students mentioned that the feedback from MyProgrammingLab was not always useful in helping them identify their mistakes. Two students explicitly stated that they did not feel as though MyProgrammingLab was helpful at all. One student expressed that they felt many of the questions were poorly worded and one student commented that they did not like the balance of questions from one topic to the next. At the end of the BlueJ survey students were also asked to provide additional written comments about BlueJ. Thirteen students did not write any comments. Five students commented that they found BlueJ useful. For example 'Overall pretty useful and easy to navigate' and 'I think it's useful.' Three students commented on the usefulness of the color coding in BlueJ, for example: 'The colors in BlueJ really help me understand the

anatomy of a program.' Two students mentioned that the compiler errors were not helpful and one student expressed that the compiler errors were helpful. Two students mentioned that they wanted to use an IDE 'used in industry.' One student was 'not a fan of the UI' and one student mentioned that BlueJ crashes on their personal machine, but works fine on the college machines.

The third survey included questions about MyProgrammingLab and BlueJ, in addition to questions about the textbook, study techniques, class sessions, lab sessions and the course overall. Twenty-two students completed this survey at the end of the semester. Students responded to statements with Likert-type responses (strongly disagree, disagree, neither disagree nor agree, agree and strongly agree). Most students agreed that completing the MyProgrammingLab exercises helped them understand the course material (17 agreed, 4 neither and 1 disagreed). Almost all students agreed that writing programs in BlueJ helped them better understand the course material (21 agreed and 1 neither). In regards to the course textbook, students had mixed feelings. Only six students agreed that the textbook helped them understand the course material. Nine students were unsure and seven students did not feel as though the course textbook helped them understand the course material. Many students agreed that the module notes posted by the instructor on Blackboard were helpful (16 agreed and 6 neither). Many students felt that studying alone helped them understand the course material (12 agreed, 4 neither and 6 disagreed), but more students felt that studying and/or talking with others helped them understand the course material (13 agreed, 7 neither and 2 disagreed). Many students agreed that information they found on the Internet helped them better understand the course material (18 agreed and 4 disagreed). Almost all students agreed that the exercises completed in the lab sessions helped them better understand the course material (19 agree, 1 neither and 2 disagreed). Most students agreed that the class lectures helped them understand the course material (18 agree and 4 neither). Most students agreed that the program examples reviewed and demonstrated in class lectures helped them understand the course material (19 agreed and 3 neither). Most students agreed that completing exercises on paper in class sessions helped them better understand the course material (19 agreed and 3 neither). Almost all students agreed that the homework assignments helped them better understand the course material (21 agreed and 1 neither). Almost all students agreed that correcting their incorrect answers from homework assignments, quizzes and exams helped them better understand the course material (20 agreed and 2 neither). A brief discussion of the results is included in the following section.

DISCUSSION

These findings contribute to an important area of computing education research. The findings are similar to others (Benotti et al., 2018; Hagan and Markham, 2000; Ni et al., 2013; Williams et al., 2006) in that students surveyed in this work also at times struggled with the wording of the MyProgrammingLab exercises and unhelpful feedback, but did perceive that MyProgrammingLab and BlueJ were helpful in learning to program. More than three quarters of the students in this study agreed that both MyProgrammingLab and BlueJ were useful in helping them develop their programming skills. More than half of the students agreed that both MyProgrammingLab and BlueJ helped them identify and correct their errors. Approximately half of the students felt it was easy to use both MyProgrammingLab and BlueJ. One student reported having installation and performance issues using BlueJ on their own computer. However, this was not an issue experienced in the college labs. Perhaps providing a dedicated lab or support to students using their own computers would alleviate the installation issues reported in Hagan and Markham (2000). To our surprise, considering the student complaints in the beginning of the semester, almost half of the students enjoyed completing the MyProgrammingLab exercises and more than half of the students enjoyed writing programs in BlueJ. The advantages of timely feedback and engaged learning, facilitated by these learning systems, are quite possibly greater than the disadvantages of these learning systems. More work is needed in this area.

CONCLUSION

There are many learning systems to support teaching and learning in introductory computing courses. Advantages and disadvantages of such systems are documented in the literature, but direct evidence of advantages is limited. In this work, we gathered student perceptions of two systems: MyProgrammingLab and BlueJ. Student perceptions varied on particular items. However, many students perceived that using MyProgrammingLab and BlueJ helped them better understand the course material. More than half of the students found the feedback received in MyProgrammingLab and BlueJ to be helpful in identifying and correcting their errors. Despite the cost concerns and frustrations that some students encountered when using MyProgrammingLab, most students perceived that MyProgrammingLab was useful in helping develop their programming skills. Approximately half of the students in the course expressed interest in using MyProgrammingLab and BlueJ in future courses.

LIMITATIONS AND FUTURE WORK

We acknowledge that there are many ways to examine perceptions of learning systems and the impact of learning systems. In this study, we gathered feedback from students enrolled in one course. This is a limited sample and only one of the many ways to examine this issue. Based on the results of this study we intend to refine the surveys and subsequently gather additional

student feedback. We plan to further analyze MyProgrammingLab exercises in terms of attempts and completions. We also plan to examine student grades in relationship to the feedback collected in these surveys and the attempt and completion rates.

REFERENCES

1. Allen, J. M., Vahid, F., Downey, K., and Edgcomb, A. D. (2018) Weekly Programs in a CS1 Class: Experiences with Auto-graded Many-small Programs (MSP), *2018 ASEE Annual Conference and Exposition*.
2. Austin, T., Horstmann, C., and Vu, H. (2018) Explicit short program practice in a programming languages course, *Journal of Computing Sciences in Colleges*, 33, 4, 114-122.
3. Barr, V., and Trytten, D. (2016) Using Turing's Craft Codelab to support CS1 students as they learn to program, *ACM Inroads*, 7, 2, 67-75.
4. Benotti, L., Aloi, F., Bulgarelli, F., and Gomez, M. J. (2018) The Effect of a Web-based Coding Tool with Automatic Feedback on Students' Performance and Perceptions. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ACM, 2-7.
5. BlueJ. https://www.bluej.org/
6. Brusilovsky, P., Edwards, S., Kumar, A., Malmi, L., Benotti, L., Buck, D., ... and Urquiza, J. (2014) Increasing adoption of smart learning content for computer science education, *Proceedings of the Working Group Reports of the 2014 on Innovation and Technology in Computer Science Education Conference*, ACM, 31-57.
7. Char, B. (2016) Automatic feedback systems for code: what can they tell the busy instructor?, *Journal of Computing Sciences in Colleges*, 31, 4, 87-93.
8. Coderunner. https://coderunner.org.nz/
9. CodingBat. https://codingbat.com/java
10. Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. (1989) User acceptance of computer technology: a comparison of two theoretical models, *Management Science*, 35, 8, 982-1003.
11. Hagan, D., and Markham, S. (2000) Teaching Java with the BlueJ environment, *Proceedings of Australasian Society for Computers in Learning in Tertiary Education Conference ASCILITE 2000*.
12. Kölling, M. (1999) Teaching object orientation with the blue environment, *Journal of Object-Oriented Programming*, 12, 2, 14-23.
13. Kölling, M. (2008) Using BlueJ to introduce programming, In *Reflections on the Teaching of Programming*, Springer, 98-115.
14. Kölling, M., Quig, B., Patterson, A., and Rosenberg, J. (2003) The BlueJ system and its pedagogy, *Computer Science Education*, 13, 4, 249-268.
15. Kumar, A. N. (2005). Results from the evaluation of the effectiveness of an online tutor on expression evaluation, *ACM SIGCSE Bulletin*, 37, 1, 216-220.
16. Liang, Y. D. (2017) Introduction to Java Programming and Data Structures, plus MyProgrammingLab, Pearson.
17. Lahtinen, E., Ala-Mutka, K., and Järvinen, H. M. (2005) A study of the difficulties of novice programmers, *ACM SIGCSE Bulletin*, 37, 3, 14-18.
18. Lobb, R., and Harlow, J. (2016) Coderunner: a tool for assessing computer programming skills, *ACM Inroads*, 7, 1, 47-51.
19. MyProgrammingLab, Pearson. https://www.pearsonmylabandmastering.com/northamerica/myprogramminglab/
20. Ni, G., Bishop, D., and Donaldson, A. (2013) Use of Online Tools in Teaching C++ Programming to Freshmen in All Engineering Majors, *5th First Year Engineering Experience Conference*.
21. Norman, V. T., and Adams, J. C. (2015) Improving non-CS major performance in CS1, In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ACM, 558-562.
22. Olsson, M., and Mozelius, P. (2016) On design of online learning environments for programming education, In *Academic Conferences and Publishing International ECEL*, 12-24.
23. Piteira, M., and Costa, C. (2013) Learning computer programming: study of difficulties in learning programming, In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, ACM, 75-80.
24. Prince, M. (2004) Does active learning work? A review of the research. *Journal of engineering education*, 93, 3, 223-231.
25. Problets. http://www.problets.org/
26. Python Tutor. http://pythontutor.com/
27. Robins, A., Rountree, J., and Rountree, N. (2003) Learning and teaching programming: A review and discussion, *Computer science education*, 13, 2, 137-172.
28. Williams, G. C., Bialac, R., and Liu, Y. (2006) Using Online self-assessment in introductory programming classes. *Journal of Computing Sciences in Colleges*, 22, 2, 115-122.