Student preferences for consuming supplemental instructional material in CS0/CS1/CS2 courses

Giovanni Vincenti
University of Baltimore
United States
gvincenti@ubalt.edu

Scott Hilberg
Towson University
United States
shilberg@towson.edu

James Braman
Comm. College of Baltimore County
United States
jbraman@ccbcmd.edu

Abstract: The concept of flipping the classroom is slowly gaining tractions at all levels of education. An ever-growing set of resources gives unprecedented access to Information Technology (IT), Computer Science (CS), and Information Systems (IS) students to a significant amount of supplemental material. Videos, interactive demonstrations, and sandboxes allow instructors of programming courses to easily implement a flipped classroom model. This work summarizes the preferences and concerns of students regarding desktop-based and smartphone-based resources for CS0/CS1/CS2 education, giving guidelines for instructors to follow regarding the types of material preferred for each medium.

Introduction

Traditional classroom environments are quickly fading into a blended learning paradigm as students utilize online resources to augment the material discussed during lecture or lab. This phenomenon is occurring in most disciplines (Bretzmann 2013), whether the students are seeking further validation on their understanding of Shakespeare or to review for an exam in geometry. Computer programming is certainly no exception.

We can find many resources that are available online, and that we routinely suggest to our students as supplemental material. Videos offered through The New Boston, Codecademy's practice environment, and generic YouTube help allows novices to get acquainted with key concepts. At times though the material that students utilize is either too generic, too specific, or out of context. Either one of these situations endangers an already struggling student to lose their faith in the teacher, the resources that they are using, and most importantly in themselves.

Since our computing students are constantly seeking supplemental online material to use, it is only natural that instructors should plan on implementing some type of blended learning strategy into their course. Such strategy may range from a simple video that students should watch before coming to class, all the way to a fully flipped classroom environment. The authors already discussed briefly the difficulties involved in flipping the classroom, especially from the point of view of resource allocation (Vincenti & Braman 2013). Creating an online environment tailored around the students is essential, as it would enable instructors to keep track of individual progress.

The drive that motivates this study revolves around the project *line_explorer*, which is a tool that is being developed by the authors to support students by combining video- and text-based explanations, interactive demonstrations, and assessments into one tool. This system is described in more detail in Vincenti et al. (2013). Since the students are the end users of this product, we created a questionnaire where they could share with us the sources of supplemental instruction that they use, and to gather their opinion and wish-list on desktop-based and mobile-based educational material. This article reports the initial findings of this questionnaire. Any reference to desktop-based resource indicates something that the students can access from their laptop or desktop.

Literature Review

While the use of video lectures, animations and games have been well established in the e-learning community to reinforce certain topics, there are still difficulties in conveying programming concepts to novice programming students. The use of multimedia and web based interactivity have been used in other learning initiatives to help novice programmers, and have been increasingly used in a flipped classroom approach. As students are able to use various tools to augment their learning based on their own skills and pace, having the option for a web-based tool is often useful. Students are able to receive instant feedback, tips and a visualization of steps in the process which are all helpful components in learning. In a flipped classroom (or sometimes referred to as an inverted classroom), the students are responsible for reading specific content, watching videos and sometimes hands-on activities as “homework” before the class. In the class meeting for a particular topic, the instructor can focus efforts on content mastery though in-class activities, exercises and collaborations. This idea of flipping the traditional role of lecture during class time and hands-on activity as homework can be seen as a changed role of the teacher where class time can be better utilized. With programming in particular, students can benefit from this approach as much of the learning needs to be conducted from a hands-on approach.

In order to provide students with alternative formats of interactive content to foster this flipped approach and motivate learning programming skills, other approaches have been explored. Alternate formats for learning are also needed to help reinforce skills learned in traditional settings. Many instructors that have experience teaching introductory courses can attest to the difficulty many students face in these courses, especially with certain topics. Added to the content complexity, is the diversity of students and experience levels. Gomes and Mendes (2007) devised a list of reasons novice programming students have difficulties, which include:

- Programming demands a high abstraction level;
- Programming needs a good level of both knowledge and practical problem solving techniques;
- Programming requires a very practical and intensive study, which is quite different from what is required in many other courses (more based in theoretical knowledge, implying extensive reading and some memorization);
- Usually teaching cannot be individualized, due to common class size;
- Programming is mostly dynamic, but usually [taught] using static materials;
- Teachers' methodologies many times do not take into consideration the student's learning styles.
- Different students have different learning styles and can have several preferences in the way they learn;
- Programming languages have a very complex syntax with characteristics defined for professional use and not with pedagogical motivations.

While some of these issues can be addressed by innovative educational tools and approaches, there are still many difficulties to overcome to reach all students. Using programming environments such as Alice or Scratch to engage students to program through the use of animations in a 3D world have been recommended to convey programming concepts to novices. Some researchers have suggested that learning to program through animations are a helpful way to motivate and engage students on basic concepts, where actions (including mistakes) can be seen in real time (Kelleher & Pausch 2007). However, some studies have suggested that the use of Alice has not shown a significant change in student perception to other tools used in introductory programming courses and that more research is needed (Schultz 2011).

The creation of interactive objects to convey more complex programming concepts have also been explored in a limited content, such as 3D interactive stacks and queues (Braman et al. 2009). Others have proposed other visual representations of algorithms (Ben-Bassat Levy et al. 2003, Malmi et al 2004). These approaches rely on the visualization of concepts to help teach content, but limited research has been conducted on the interactive approach to these visual tools in programming. While it has been suggested that the programming language chosen to teach introductory programming courses are a factor (Goosen 2008), others suggest that the implementation of code, not teaching the concepts of programming is the root of the problem (Lahtinen et al. 2005).

Student Preferences and Concerns

The questionnaire included three main parts. The first part focused on establishing a profile of the respondents. They were asked questions such as their major, previous and current experience with programming. The second section focused on their use and wishes for Desktop-based resources as supplemental instructional material. The third section shifted the focus on resources available through mobile devices.

Profile of the students

The questionnaire was administered to students who were taking CS0, CS1, and CS2 courses at two metropolitan institutions in the Greater Baltimore area during the Spring 2014 semester. 190 people responded to the survey. Our first goal was to identify the type of degree that the students are working on. Table 1 shows a relatively detailed breakdown of degrees.

Table 1: Frequency of academic programs

Academic Program	*Students*
IT	74
CS	46
IS	28
Technology major	148
Mathematics	12
Physical Science	9
Liberal Arts	4
Simulation and Digital Entertainment	4
Social Sciences	3
Undecided	1
Other major	33
Unreported major	9
Total	190

A total of 148 students are enrolled in technology-related majors, predominantly enrolled in Information Technology (IT) degrees. Among the respondents, there were also many Computer Science (CS) majors, and several Information Systems (IS) students. Collectively we will refer to them as Technology majors, giving the group total as well as the breakdown by degree in the rest of the article.

Among the students who answered the survey, several were enrolled in Mathematics and Physical Sciences (such as Chemistry and Molecular Biology). We were expecting to see some overlap with other STEM disciplines. We were surprised to see the amount of students who were pursuing non-STEM majors enrolled in the programming courses, which we collectively refer to as Liberal Arts and Social Sciences. Majors in these groups include Psychology, Sociology, and Mass Communications. We will refer to this group of students as 'Other majors'. Nine students did not report their major.

Among the different disciplines, there were also students working on Simulation and Digital Entertainment (SDE) degrees. We are not going to consider those students as part of the Technology group because this specialization is not included in the list of STEM programs (USDOE 2014). The closest field present in this list is "Modeling, Virtual Environments and Simulation", which does not explicitly gear its curriculum to entertainment and gaming.

Table 2: Number of past programming courses

Academic Program	*0*	*1*	*2*	*3*	*4+*
IT	11	18	13	17	12
CS	11	21	7	4	1
IS	6	8	6	7	1
Technology major	28	47	26	28	14
Other major	22	6	3	1	1
Unreported major	0	2	2	1	1
Total (182)	50	55	31	30	16

Table 3: Experience with programming on a scale from 1 (Not experienced) to 5 (Very experienced)

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	12	22	31	8	1	2.51
CS	4	11	16	13	2	2.96
IS	2	4	15	6	1	3
Technology major	18	37	62	27	4	2.74
Other major	10	5	11	5	2	2.52
Unreported major	1	3	2	3	0	2.78
Total (190)	29	45	75	35	6	2.71

We then collected information on their programming experience. Since we are working with students in CS0, CS1, and CS2 we expected a wide difference of experience. First of all, the majority of students were not on their first programming course, as shown in Table 2. Most of them took one course before the one in which they were asked to complete the survey. We also had several students who completed two or three courses. A few completed more than four, and 8 did not answer.

In the next few questions we asked the students to quantify their answer on a Likert scale ranging from 1 to 5. First, we wanted to look at their self-confidence with programming by asking about their experience, comfort level, and attitude towards this activity. We asked the students to rate their experience, ranging from not experienced at all to very experienced. Among the results, reported in Table 3, we could clearly see that IT students regard themselves as the least experienced among the technology majors. Interestingly enough, the mean for the responses is very close to the experience level reported by students in other majors. It is worth reminding the reader that this last category includes Mathematics and SDE majors.

Next we asked the students about their comfort level with programming, and the results are reported in Table 4. Also in this case we can see the mean reported comfort level for IT majors is the lowest among all the technology majors. When we expand this comparison, we can see that IT majors also report less confidence than students of other majors enrolled in programming courses.

Table 4: Comfort level with programming on a scale from 1 (Not comfortable) to 5 (Very comfortable)

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	11	20	29	11	3	2.66
CS	0	8	16	12	10	3.52
IS	1	6	12	5	4	3.18
Technology major	12	34	57	28	17	3.03
Other major	4	7	14	4	4	2.91
Unreported major	1	2	3	1	2	3.11
Total (190)	17	43	74	33	23	3.01

Table 5: Attitude towards programming on a scale from 1 (I hate it) to 5 (I love it)

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	12	19	24	8	11	2.82
CS	0	4	6	18	18	4.09
IS	4	4	8	8	4	3.14
Technology major	16	27	38	34	33	3.28
Other major	5	4	11	9	4	3.09
Unreported major	2	0	3	2	2	3.22
Total (190)	23	31	52	45	39	3.24

We then went on to assess their attitude towards programming, reported in Table 5. Also in this case IT students reported the least favorable feelings towards programming among technology majors. As expected, CS majors reported the highest mean. Interestingly, IT majors reported the least favorable feelings towards programming also compared to students of other majors.

Next, we wanted to assess the students' attitude towards the helpfulness of past and present programming courses. For this question, the students were asked to rate their experience with such courses. The results are reported in Table 6. In this case, IT students reported an attitude that matched the one of IS students, and well below the attitude of CS majors. This question was not answered by 4 students.

Table 6: Attitude towards programming courses on a scale from 1 (They did not help at all) to 5 (They helped very much)

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	2	13	23	20	1	3.45
CS	1	3	9	15	18	4.00
IS	2	3	8	9	5	3.44
Technology major	5	19	40	44	38	3.62
Other major	2	4	9	8	8	3.52
Unreported major	0	4	3	1	1	2.89
Total (186)	7	27	52	53	47	3.57

Table 7: Students who have used Internet in the past to consume supplemental instruction for programming

Academic Program	*Yes*	*No*
IT	69	5
CS	39	7
IS	20	8
Technology major	128	20
Other major	16	17
Unreported major	8	1
Total (190)	152	38

As a last set of questions to establish a high-level profile of students taking programming courses, we asked them whether or not they used the Internet to learn how to program. The results are reported in Table 7. Looking at only technology majors, we can see that a large amount of students have used Internet to learn how to program. Interestingly, students who are pursuing degrees other than IT/CS/IS report a slightly larger percentage of them not using the Internet.

Desktop-based resources

The next portion of the questionnaire focused on the utilization of Desktop-based material as supplemental instruction for a course. Our main goal was to profile the types of resources that students utilize and their feedback on them, so that we can optimize our efforts in creating modules that match their expectations from online material.

The first question inquired whether students would use Desktop-based resources to supplement their learning experience. The results are reported in Table 8. The data does not clash with what was presented in Table 7, since the previous data indicates that some students may not have utilized such resource already. The fact that nearly all students would utilize Desktop-based resources supports our project, which would develop modular and tailored material supporting typical classroom instruction.

Table 8: Students who would use Internet in the future to consume supplemental instruction for programming

Academic Program	*Yes*	*No*
IT	73	1
CS	46	0
IS	28	0
Technology major	147	1
Other major	31	2
Unreported major	9	0
Total (190)	187	3

The next set of open-ended questions focused on the reasons for using (or not using), the desirable features, and issues or concerns with using Internet based materials to help learn programming. Students indicated they would use online materials primarily to supplement course materials. Availability, accessibility and convenience were the top reasons provided. Students pointed to the number and variety of examples available online and their ability to use these materials on demand and repetitively as needed. They also suggested that online materials were current and frequently updated.

Students were asked, again in an open-ended question, what online resources they were already using to help learn programming. Results are reported in Table 9 and Table 10. Of the 190 students completing the questionnaire, 163 reported using online resources, 16 indicated they have not used online resources and 11 students did not respond. These results conflict somewhat with responses reported in Table 7 where 38 students indicated they had not used the Internet in the past to supplement instruction for programming. Further investigation found that some students indicated they had not previously used the Internet to supplement learning but indicated they had used online course materials such as PowerPoint slides and lecture notes suggesting they viewed online course materials differently from external online resources. A few students simply responded inconsistently indicating they had not previously used the Internet to supplement learning and then also responded that they used online resources such as YouTube and Google.

Table 9: Online resources students are already using to help learn

Academic Program	*YouTube Videos*	*Online Forums*	*Google*	*Programming Websites*
IT	28	23	12	20
CS	15	12	10	9
IS	14	7	5	1
Technology major	57	42	27	30
Other major	11	5	8	5
Unreported major	2	1	2	1
Total (186)	70	48	37	36

YouTube was the number one online resource used by students to learn programming. They referenced the volume, variety, and availability of programming videos on YouTube. The second most popular resource for students, were online forums. They referenced "coding" and "programming" forums without providing specific forum names. Interestingly, Google was the third most popular response for online resources, suggesting students did not make a distinction between the search engine the actual content provider. In nearly equal numbers to Google, students indicated they used websites, such as StackOverflow, Codecademy, and W3Schools that specifically provide programming instruction, tutorials, and questions & answers. Finally, a few of students reported accessing online documentation from vendors such as Microsoft and Oracle.

Table 10: Students who have not used online resources or did not report using online resources

Academic Program	*Have not used online resources*	*Did not report using online resources*
IT	73	1
CS	46	0
IS	28	0
Technology major	147	1
Other major	31	2
Unreported major	9	0
Total (190)	187	3

Students prefer online instructional videos, tutorials and tools to test or validate code. The validity and reliability of online programming materials and the lack of personal interaction and feedback from others topped the list of student concerns. They also felt that online materials may emphasize "copying code" rather than learning programming concepts.

Mobile-based resources

The last part of the questionnaire focused on the potential use of mobile applications to learn or review concepts of programming. This portion of the questionnaire revolved around what students would like an educational app to feature, and what concerns them about this particular delivery system. We also added a few questions related to our own initial concerns.

Our first question investigates whether the students would be interested in supplemental educational material available on mobile devices. The results, reported in Table 11, show that technology majors are divided over the possible use of such apps. There is a slight preference about not using such resources for students who are classified as other majors in this study. Four students did not respond to this question.

Table 11: Students who would use mobile devices to consume supplemental instruction for programming

Academic Program	*Yes*	*No*
IT	37	36
CS	23	22
IS	14	14
Technology major	74	72
Other major	13	19
Unreported major	4	4
Total (186)	91	95

The next set of open-ended questions focused on the reasons for using (or not using), the desired features and issues/concerns with using mobile devices to help learn programming. Students were split on using mobile devices to supplement learning to program. Many indicated they always carry their phone and therefore mobile devices would be a good way to access online programming materials. A nearly equal number of students objected to using a mobile device because of the small screen size and the lack of full size keyboard. In addition, a number of students reported wanting to learn on the same platform that they were actually programming on (typically a laptop or desktop computer). Those who were supportive of using mobile devices to help learn programming, primarily expressed interest in viewing programming videos. While many students seemed supportive of using mobile devices to supplement learning, the vast majority expressed concerns about their actual use. Bandwidth, small screen size and lack of full keyboard were the predominate concerns.

At last, we wanted to understand how our own doubts regarding media-rich and interactive content can appeal to programming students. We used these last four questions to assess the students' view on such limitations of mobile devices. We did not explicitly specify whether the mobile application would be running on a tablet or a cellular phone at this time. We asked our students to rate their concern for each of the following issues on a scale of 1 (great concern) to 5 (no concern at all). The results are reported in Table 12 through Table 15.

The first and ever-present issue is in reference to small screens, often found in smartphones. The results, reported in Table 12, show that the size of the screen is a great concert for students. CS students are most concerned, but also IT and IS students see this aspect of mobile devices as a potential limitation. Students working on other majors seem slightly less concerned, but certainly not at ease with material delivered on such small size screens. A total of 182 students answered this question.

Table 12: Concerns regarding screen size

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	27	20	9	5	11	2.35
CS	14	14	7	3	5	2.33
IS	13	4	5	3	2	2.15
Technology major	54	38	21	11	18	2.3
Other major	12	3	9	5	3	2.5
Unreported major	2	2	1	1	2	2.87
Total (182)	68	43	31	17	23	2.36

Table 13: Concerns regarding listening to voice explanations

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	7	14	18	20	13	3.25
CS	7	6	9	10	11	3.28
IS	6	6	9	6	1	2.64
Technology major	20	26	36	36	25	3.14
Other major	4	10	5	7	6	3.03
Unreported major	0	1	1	1	5	4.25
Total (183)	24	37	42	44	36	3.17

Many online resources rely on video and audio features, which are easily available on mobile devices but not necessarily easy to consume effectively. When we asked the students about possible difficulties listening to audio description of the material, 183 answered as reported in Table 13.

Overall the students are less concerned about this type of delivery. It is worth noting that IS students are more concerned than all other students. If we look at the overall distribution, students are spread out relatively evenly with a focus on the ratings of 3 and 4. This distribution is consistent with technology students. We notice an inverse trend when we ask students about concerns of video material, reported in Table 14.

Table 14: Concerns regarding watching video explanations

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	13	13	20	16	10	2.96
CS	9	10	8	9	7	2.88
IS	7	7	8	2	4	2.61
Technology major	29	30	36	27	21	2.87
Other major	5	11	2	8	6	2.97
Unreported major	1	2	2	1	2	3.12
Total (183)	35	43	40	36	29	2.9

Table 15: Concerns regarding interaction with the material

Academic Program	*1*	*2*	*3*	*4*	*5*	*Mean*
IT	30	28	11	2	1	1.83
CS	21	5	11	3	3	2.12
IS	13	8	5	2	0	1.86
Technology major	64	41	27	7	4	1.92
Other major	12	9	7	2	2	2.16
Unreported major	3	2	1	2	0	2.25
Total (183)	79	52	35	11	6	1.98

Also in this case IS students are most concerned among the technology students. Overall students are slightly more concerned about watching videos (2.9) compared to listening to explanations (3.17). Looking at the distribution, we can notice nearly a mirror image of the overall distribution, where the majority of students chose ratings of 2 and 3.

We also asked the students how concerned they were about interacting with the material, when an input would be required. Table 15 reports the responses. We can immediately notice that this aspect of mobile applications gives students the most concern for technology majors (1.92). We can also see that among the technology students, those working on CS degrees are slightly less concerned (2.12) and IT students are the most concerned (1.83).

Discussion of the Results

This article shows a very brief overview of the results, which carry a wealth of information within the short answer portions of the survey. From this superficial analysis of the results we can notice a few interesting trends. First of all, results of the questionnaire show that IT students are not inclined to enjoy programming. This is distinct from their ability of carrying out such activity. Overall, there seems to be a disconnect between what they wish to do and how they wish to carry it out. As programming and scripting are often essential components of their future careers, we would expect a better attitude towards this aspect of IT.

The main goal of this work was to gain a better understanding of what types of resources students who are learning how to program utilize. Our second goal was to gain data regarding which different types of supplemental instructional resources the students would utilize and through which technological medium. In doing so, we have painted a clear picture of what students like and do not like. First of all, students prefer consuming material while at a computer system with a larger viewable surface, possibly where they program. Such setting would suggest that most meaningful learning does not happen while working with mobile-based systems since they have to switch the technological context in which they operate.

Another interesting trend that we noticed is that a number of students (22%) did not appear to make a distinction between the search mechanism and content provider when seeking online programming materials. When asked about the online resources they use, these students responded simply "Google" with no specific mention of the actual material or content provider. We would expect such inaccuracy (or at least lack of preference) from students who are not majoring in a technology-related field, but instead this observation was distributed throughout.

Implementing the concept of a flipped classroom involves the use of material that the students will consume while away from class, in order to maximize the time spent with an instructor to work through problems or at least apply the concepts learned. Often we think of mobile-based devices as the perfect way for someone to review material “on the go”, but our results show that students are not too keen on doing anything but watch instructional videos on their mobile phones. When we are working with programming, it is often necessary to practice the skills acquired, as services such as Codecademy allow. The interaction component related to mobile-based resources offers a particularly challenging situation that may or may not be resolved easily with current technologies. Certainly a keyboard or a stylus would simplify such interaction, but the combination of a small screen would seriously limit the ability to convey elaborate code or wide areas dedicated to interactive exercises or demonstrations.

Conclusions

Overall we were very satisfied with the results of this survey. The main concept that prompted this brief survey was to ensure that what we will build into *line_explorer* is in line with what the students like and expect. The students' desire for online materials that allow step-by-step walk through of programming examples supports this project and continued development of *line_explorer*. We will also evaluate the adaptation of certain modules of this system for mobile devices, but at this point we will focus the majority of our efforts to creating material accessible through traditional web browsers.

References

Ben-Bassat Levy, R., Ben-Ari, M., & Uronen, P.A. (2003). The Jeliot 2000 program animation system. *Computers & Education*, 40(1), 1-15.

Braman, J., Vincenti, G., Arboleda Diaz, A.M., & Jinman, A. (2009). Learning Computer Science Fundamentals through Virtual Environments. *Online Communities and Social Computing*, 423-431.

Bretzmann, J. (2013) Flipping 2.0: Practical Strategies for Flipping Your Class. Bretzman Press. New Berlin, Wisconsin.

Gomes, A., & Mendes, A.J. (2007). An environment to improve programming education. *Proceedings of the 2007 international conference on Computer systems and technologies*, Association for Computing Machinery.

Goosen, L. (2008). A brief history of choosing first programming languages. History of Computing and Education. 167-170.

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, 50(7), 58-64.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.M. (2005). A study of the difficulties of novice programmers. ACM SIGCSE Bulletin, 37(3), 14-18.

Malmi, L., Karavirta, V., Korhonen, A., Nikander, J., Seppälä, O., & Silvasti, P. (2004). Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. Informatics in Education-An International Journal, 3(2), 267-288.

Schultz, L.A. (2011). Student Perceptions of Instructional Tools in Programming Logic: A Comparison of Traditional versus Alice Teaching Environments. *Information Systems Education Journal*, 9(1).

U.S. Department of Education. Science, Technology, Engineering and Math: Education for Global Leadership. Retrieved from http://www.ed.gov/stem, on June 2, 2014.

Vincenti, G., & Braman, J. (2013). A few thoughts on the Flipped Classroom. *EAI Endorsed Transactions on e-Learning*, 13(3). doi: 10.4108/el.1.3.e1

Vincenti, G., Braman, J., & Hilberg, S. (2013). Teaching Introductory Programming Through Reusable Learning Objects: A Pilot Study. *Journal of Computing Sciences in Colleges*, 28(3), 38-45.