

Memo

To: Iowa State University Department of Computer Science  
From: John Doe  
Subject: COM S 227 Proposal

:  
Date: February 25, 2017  
CC: Brian De Wall

---

## Introduction

The Iowa State University Department of Computer Science (ISU) has a problem with the current learning resources available for the Introduction to Object-Oriented Programming (COM S 227) course. Improvements to these learning resources are necessary to remove the negative effects that the current learning resources for COM S 227 create for ISU. These improvements will also allow ISU to become a top educator in programming. This document explains the plan to improve these learning resources by covering:

- The current problem
- Objectives of the project
- The proposed solution
- Methods to achieve the solution
- Benefits of this project

## Current Situation

Nearly half of the students who registered for the Introduction to Object-Oriented Programming course (COM S 227) at ISU either dropped the course within the first week of class or earned a D or F in the course, as stated by Drees in her article, *How to avoid failing the most-failed classes at Iowa State* [1]. This indicates that there is a problem with the learning resources available in this course. Other research in [2], [3], [4], [5], and [6] support a similar conclusion about problems with current learning resources used to learn programming.

When I was enrolled in COM S 227, the out-of-class learning resources were a textbook, sample code from class, homework assignments, and labs. The textbook and sample code are effective for developing a shallow understanding of concepts, but since students are not programming themselves when using these resources, some have difficulty applying the concepts. In 2015, Yang et al. noted in their journal article, *A Two-Tier Test-Based Approach to Improving Students' Computer-Programming Skills in a Web-Based Learning Environment*, that learning from textbooks and sample code can be helpful, but those resources should only be supplemental for topics in which students struggle [3]. Sample code and the textbook in COM S 227 are instead used as the main learning resources. Additionally, the journal article from [3] states that continuous practice in programming is necessary for students to retain the information. Since the homework and labs in COM S 227 do not cover all the topics from the course, continuous practice is difficult for students to

achieve in all topics. This leads to students being confused on how to apply the concepts not covered in homework or lab.

The MIT programming lecture found on YouTube shows a similar experience to the classroom experience at ISU where the teacher does most of the programming while the students just observe [7]. In addition to the little student interaction in lecture, a report by Konecki et al. stated in their journal article, *Intelligent assistant for helping students to learn programming*, that teachers struggle to find a teaching method that appeals to students' preferred learning styles [5]. For these reasons, it can be a challenge for some students to learn from lecture at all.

## Objectives

The goal of this project is to update the learning resources used in COM S 227. This will benefit both students and ISU. The proposed solution will achieve the following goals:

- Create more effective learning resources by updating the sample code, textbook, homework, labs, and lecture content
- Lower the failure and drop rate for COM S 227
- Increase student understanding of COM S 227 concepts
- Improve the quality of programming education at ISU

## Proposed Solution

There has been research that provides potential solutions to the learning resources problem listed above. Results of the study in [3] found that explaining why solutions are correct improves understanding for students in programming. Finding textbooks and sample code that use this method will allow the students to learn more from these resources. Research in *Learn Programming++: The Design, Implementation and Deployment of an Intelligent Environment for the Teaching and Learning of Computer Programming* by Hunter et al. and technologies shown on Khan Academy and Codecademy use a learning interface that teaches students programming concepts but allows the students to do the programming while learning [8] [9] [10]. The problem with current homework and labs will be resolved once a learning interface like this is utilized because students will obtain coding practice in COM S 227 programming topics as they learn.

Wang et al. showed in their journal article, *Enhancing Students' Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online Peer-Assessment Attempt*, that programming using a graphical user interface (GUI) helps students understand the basics of programming by enabling, "novice learners and young students to realize the basic programming logic without being confused by programming language syntax" [2]. In other research referenced in *Games as A Motivation For Freshman Students To Learn Programming*, Feldgen and Clua showed how relating programming to something familiar, such as a game, aids in the understanding of programming topics [4]. Teachers could utilize methods like visualization of concepts by using Scratch [11] and create connections to familiar topics when teaching to help improve students' understanding in lecture. This allows students to gain multiple forms of exposure to programming, rather

than just seeing the teacher code. Implementing improvements to the current learning resources like the ones listed above will help improve the overall quality of COM S 227.

## Methods

The project will be broken down into multiple steps as illustrated below in Figures 1 and 2.

1. Setup Scratch
2. Determine Useful Programming Connections
3. Update Textbook and Sample Code
4. Research Requirements for Software Interface
5. Develop and Test Software
6. Improve Software
7. Update Lesson Plans

ID	Task Name	Duration	15-Apr	22-Apr	29-Apr	6-May	13-May	20-May	27-May
1	Setup Scratch	1 Day							
2	Determine Useful Programming Connections	2 Weeks							
3	Update Textbook and Sample Code	3 Weeks							
4	Research Requirements for Software Interface	2 Weeks							
5	Develop and Test Software	5 Weeks							
6	Improve Software	2 Weeks							
7	Update Lesson Plans	3 Weeks							

**Figure 1:** The project timeline for weeks one through seven.

ID	Task Name	Duration	3-Jun	10-Jun	17-Jun	24-Jun	1-Jul	8-Jul	15-Jul	22-Jul	29-Jul
1	Setup Scratch	1 Day									
2	Determine Useful Programming Connections	2 Weeks									
3	Update Textbook and Sample Code	3 Weeks									
4	Research Requirements for Software Interface	2 Weeks									
5	Develop and Test Software	5 Weeks									
6	Improve Software	2 Weeks									
7	Update Lesson Plans	2 Weeks									

**Figures 2:** The project timeline for weeks eight through sixteen.

### Cost Sources:

- According to Chronicle Data, the average annual salary of an Iowa State professor is around \$123,000 [12]. We can also assume that professors work around 2,000 hours per year. This creates an approximate hourly wage of \$61.5.
- According to Glassdoor, the average annual salary of a software developer in education and schools is around \$68,000 [13]. We can also assume that software developers work around 2,000 hours per year. This creates an approximate hourly wage of \$34.

- According to Glassdoor, the average hourly wage of an undergraduate TA at Iowa State University is around \$10 [14].

### 1. *Setup Scratch – (1 Day ~ 3 Hours)*

Scratch accounts must be setup for all professors and teaching assistants of COM S 227 along with all students who are enrolled in the course [11]. The teacher and student accounts will be placed into a class group for each section of COM S 227 [11]. Scratch will facilitate visual learning of programming concepts. This task will need to be completed every semester COM S 227 is offered. These accounts and groups will be used in the *Update Lesson Plans* task below.

- Facilities and Equipment: A room and a computer with online access are necessary. These resources are available to teaching assistants through campus computer labs.
- Personnel: One teaching assistant is necessary.
- Cost: 3 hours at \$10 per hour = \$30

### 2. *Determine Useful Programming Connections – (2 Weeks ~ 65 Hours)*

Ten teaching assistants will meet to determine connections, like analogies, between COM S 227 course topics and other familiar topics. Having teaching assistants come up with helpful metaphors allows students to create their own metaphors more easily. Creating these connections is valuable for student understanding, and when teaching assistants do the initial work it is more cost-effective than using professors. At least one connection will be determined for each programming topic. These connections will be used in the *Update Lesson Plans* task below.

- Facilities and Equipment: A meeting location will be necessary. These locations may include professor offices, private, bookable campus rooms, or public meeting areas. An online notetaking medium like Google Docs can be utilized for taking meeting notes on which connections to make for each topic.
- Personnel: Ten teaching assistants are recommended.
- Cost: 65 hours at \$10 per hour = \$650

### 3. *Update Textbook and Sample Code – (3 Weeks ~ 90 Hours)*

The textbook and sample code used in the class will be updated to explain why answers are correct and why the code accomplishes what it does. Doing so will help students learn more effectively from these resources. Professors will work towards either writing new material or finding and using materials that explain why the answers and code are correct. They will also work towards updating the comments in the current class code to explain why it behaves the way it does. These updated resources will be used in the *Update Lesson Plans* task below.

- Facilities and Equipment: A location will be necessary for the work. The professor's office will be used for this location. Updates to the sample code will be posted on the current COM S 227 website; therefore, having access to change the website is necessary. Updates to the textbook will either be distributed as a hard copy through the ISU bookstore if a new textbook is found or online if the written materials are chosen to be rewritten.
- Personnel: One professor will be necessary.
- Cost: 90 hours at \$61.5 per hour = \$5,535

#### 4. *Research Requirements for Software Interface – (2 Weeks ~ 75 Hours)*

Professors, teaching assistants, and developers will brainstorm a list of requirements and features for the online interactive learning interface that students will use to learn programming. This software interface will help students get practice in all programming topics. The requirements that will be determined include what topics will be covered, specific problems or ways to learn the material through programming, designing projects for the students to practice after they learn the material, and the general look and feel of the application. These requirements will be used in the *Develop Software* task below.

- Facilities and Equipment: A larger, dedicated room will be necessary for the development of the requirements for the online learning interface. ISU will provide this room. Access to whiteboards, sticky notes, and online project management tools like Trello may be necessary.
- Personnel: Two professors, five teaching assistants, and three developers will be necessary.
- Cost:
  - Professors: 25 hours at \$61.5 per hour = \$1,537.5
  - Teaching assistants: 25 hours at \$10 per hour = \$250
  - Developers: 25 hours at \$34 = \$850
  - Total Cost = \$2,637.5

#### 5. *Develop and Test Software – (5 Weeks ~ 600 Hours)*

A group of three developers (one for front-end, one for back-end, and one for connectivity) will develop the software based on the requirements generated in task 4 above. The development process will work in one week increments with demos at the end of each week to ensure the project remains on the correct path.

- Facilities and Equipment: A larger, dedicated room will be necessary for the development of the learning interface. ISU will provide this room. Developer equipment will also be required. ISU currently possesses and will provide these resources. Some potential examples include the following:
  - A computer with development software preinstalled
  - Connection to Iowa State University's network
  - Servers for running the application
  - Databases to store information about the application
  - Online software project management tools like Trello and Gitlab
  - Tables, chairs, and whiteboards
- Personnel: Three developers will be necessary.
- Cost: 600 hours at \$34 per hour = \$20,400
- Note: An alternative approach can be utilized by using a pre-existing interface that accomplishes a similar goal. The cost and details of this implementation is dependent on which option is chosen.

#### 6. *Improve Software – (2 weeks ~ 100 Hours)*

At the end of the development process, there will be about a one-week testing period where the developers, professors, teaching assistants, and a random sample of students will test out the software and give their feedback. If the feedback indicates further improvements need to be made to the software, the development team will work to make improvements on those

areas. Once those improvements have been made, a final test will be done by the developers, professors, teaching assistants, and students to ensure the software is as expected. The final software will be used in the *Update Lesson Plans* task below.

- Facilities and Equipment: A larger, dedicated room will be necessary for the improvement of the learning interface. ISU will provide this room. Access to whiteboards, sticky notes, and online project managements tools like Trello may be necessary.
- Personnel: One professor, five teaching assistants, three developers, and ten students are necessary.
- Cost:
  - Professors: 5 hours at \$61.5 per hour = \$307.5
  - Teaching Assistants: 25 hours at \$10 per hour = \$250
  - Developers: 50 hours at \$34 = \$1700
  - Students 20 hours at \$0 = \$0
  - Total Cost = \$2,057.5

#### 7. *Update Lesson Plans – (2 Weeks ~ 100 Hours)*

Updates to the lesson plans for COM S 227 will be conducted by professors and teaching assistants of COM S 227. These changes will update the lecture content to include the use of Scratch and drawing connections between the COM S 227 content and other well-known topics. The out-of-the-classroom updates will include using the updated textbook and sample code as the new reference material, and the use of the new programming learning interface software. The software interface's learning modules will replace homework as the student will work through new COM S 227 topics as they are doing the programming. The projects within the software interface will replace the labs because these modules will allow the students to apply their knowledge to real-world applications.

- Facilities and Equipment: A meeting location will be necessary. These areas may include professor offices, private bookable campus rooms, or public meeting areas. A medium for creating the new lesson plans will be necessary such as Microsoft Word or the course website.
- Personnel: All professors of COM S 227 will be necessary.
- Cost: 100 hours at \$61.5 per hour = \$6,150

Total Time = About 16 weeks

Total Cost = About \$35,423

Please note that the cost estimate is calculated by the amount of time professors, TAs, and developers spend on this project. These individuals are already paid employees of Iowa State University. Please also note that all the other resources necessary for this project are already owned by and will be provided by Iowa State University.

### **Benefits**

The improvements to the learning resources for COM S 227 will provide the following benefits to ISU:

- More prestigious programming education

Improving the COM S 227 course learning resources will allow students to have a stronger programming foundation. Students will reflect well on ISU due to their high learning achievements. This will help attract high-quality students to ISU as these students will recognize ISU as a top educator in programming.

- Increased retention and enrollment rates  
Since the improvements will increase the effectiveness of the COM S 227 course, fewer students will fail the course and ISU will thus retain more students. Because of the increased quality of education, more students will also want to attend ISU, increasing the enrollment rates.
- Increased understanding and knowledge of COM S 227 topics for students  
Student understanding of COM S 227 topics will increase. This will help them in their future classes and careers, which will result in students better reflecting on ISU.
- Learning resource improvements that can be applied to other courses  
These improvements to the learning resources can not only improve COM S 227, but they can also be applied to other COM S courses as well. Making changes to other COM S courses will be much quicker and more economical since the process has already been completed once. Making these changes to more COM S courses will create an even more effective computer science education at ISU.

## Conclusion

These improvements to the COM S 227 learning resources will help students by giving them a better understanding of the programming concepts covered in this course. It will also help instructors be recognized as high-quality educators in the programming field, and it will help Iowa State University Department of Computer Science as they will be recognized as a top educator in programming; thus, attracting high-achieving students to the University. For more information about this proposed change to the learning resources in COM S 227 at Iowa State University, please contact Matthew Orth at [mmorth@iastate.edu](mailto:mmorth@iastate.edu). I will be eager to assist.

JD

## References

- [1] E. Drees, "How to avoid failing the most-failed classes at Iowa State", *Iowa State Daily*, 2013. [Online]. Available: [http://www.iowastatedaily.com/news/article\\_4ea93afe-7a4d-11e2-a34d-001a4bcf887a.html](http://www.iowastatedaily.com/news/article_4ea93afe-7a4d-11e2-a34d-001a4bcf887a.html). [Accessed: 10- Jan- 2018].
- [2] X. Wang, G. Hwang, Z. Liang and H. Wang, "Enhancing Students' Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online Peer-Assessment Attempt", *Educational Technology & Society*, vol. 20, no. 4, pp. 58-68, 2017.
- [3] T. Yang, G. Hwang, S. Yang and G. Hwang, "A Two-Tier Test-based Approach to Improving Students' Computer-Programming Skills in a Web-Based Learning Environment", *Education Technology & Society*, vol. 18, no. 1, pp. 198-210, 2015.
- [4] M. Feldgen and O. Clua, "Games As A Motivation For Freshman Students To Learn Programming", in *ASEE/IEEE Frontiers in Education Conference*, Savannah, GA, 2014, pp. S1H-11 - S1H-16.
- [5] M. Konecki, N. Kadoic and R. Piltaver, "Intelligent assistant for helping students to learn programming", in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2015, pp. 924-928.
- [6] Z. Szalayné Tahy and Z. Czirkos, "'Why Can't I Learn Programming?' The Learning and Teaching Environment of Programming", *Informatics in Schools: Improvement of Informatics Knowledge and Perception*, vol. 9973, pp. 199-204, 2016.
- [7] MIT OpenCourseWare, *6.0001 Introduction to Computer Science and Programming in Python. Fall 2016*. 2017.
- [8] G. Hunter, D. Livingstone, P. Neve and G. Alsop, "Learn Programming++: The Design, Implementation and Deployment of an Intelligent Environment for the Teaching and Learning of Computer Programming", *2013 9th International Conference on Intelligent Environments*, pp. 129-136, 2013.
- [9] "Learning programming on Khan Academy", *Khan Academy*, 2018. [Online]. Available: <https://www.khanacademy.org/computing/computer-programming/programming/intro-to-programming/a/learning-programming-on-khan-academy>. [Accessed: 10- Jan- 2018].
- [10] "We're Here To Help", *Codecademy*, 2018. [Online]. Available: <https://help.codecademy.com/hc/en-us>. [Accessed: 10- Jan- 2018].
- [11] "Scratch - Educators", *Scratch*. [Online]. Available: <https://scratch.mit.edu/educators/>. [Accessed: 15- Jan- 2018].



- [12] “Iowa State University Faculty Salaries – Chronicle Data”, *Chronicle Data*, 2018. [Online]. Available: <https://data.chronicle.com/153603/Iowa-State-University/faculty-salaries/>. [Accessed: 16- Feb- 2018].
- [13] “Salary: Developer | Glassdoor.co.uk”, *Glassdoor.com*, 2018. [Online]. Available: [https://www.glassdoor.com/Salaries/developer-salary-SRCH\\_KO0,9.htm](https://www.glassdoor.com/Salaries/developer-salary-SRCH_KO0,9.htm). [Accessed: 16- Feb- 2018].
- [14] “Iowa State University Undergraduate Teaching Assistant Hourly Page | Glassdoor.co.uk”, *Glassdoor.com*, 2018. [Online]. Available: [https://www.glassdoor.com/Hourly-Pay/Iowa-State-University-Undergraduate-Teaching-Assistant-Hourly-Pay-E23963\\_D\\_KO22,54.htm](https://www.glassdoor.com/Hourly-Pay/Iowa-State-University-Undergraduate-Teaching-Assistant-Hourly-Pay-E23963_D_KO22,54.htm) [Accessed: 14- Feb- 2018].