

Introductory Python Course: Interactive Online Course using DataCamp Lite

Matthew Hancock - [matthancock17@gmail.com](mailto:matthancock17@gmail.com) - Undergraduate in C.S.

Rachel Burk - [Rachel.Burk@stmartin.edu](mailto:Rachel.Burk@stmartin.edu) - Undergraduate in C.S.

Harold Nelson, Ph.D. - [HNelson@stmartin.edu](mailto:HNelson@stmartin.edu) – Advisor and C.S. Instructor

December 2019

***ABSTRACT*** *- Electronic learning is becoming more relevant and valuable with recent technology. Several online platforms yield simple implementation of supplemental learning for students in the classroom. This is especially useful for computer science, where students need constant feedback about the correctness of their code. DataCamp allows for interactive learning by utilizing console exercises that provide instant feedback on the mistakes made. We designed and implemented a course in DataCamp Lite as supplemental learning material for an introductory programming course involving Python. The purpose of this course is to improve student engagement and comprehension of course material.*

**Keywords**: DataCamp Lite, Submission Correctness Tests (SCT), Python, Python for Everybody (PY4E), Special Interest Group on Computer Science Education (SIGCSE)

**Introduction**

Our objective is to create a comprehensive and intuitive DataCamp course that will be used to teach students in an accessible and engaging medium. We believe that by applying our programming skills to create this content, it will benefit future computer science students of Saint Martin’s University. This DataCamp course will be utilized in CSC101 to introduce programming to computer science students.

**Literature Review**

We reviewed pedagogic topics specifically from University of Colorado’s Center for Faculty Development. The process for designing this course will require three stages. The first stage of the course design is to outline each chapter that will be covered in the DataCamp course. The Python for Everybody (PY4E) text has chapters broken into roughly sixteen subsections. These subsections benefit the structure of our course as well as influence learning objectives. The first stage’s primary focus is to write “observable learning objectives” which, the University of Colorado’s Center for Faculty Development (UCCFD) states, “provides students with a clear purpose to focus their learning efforts” (Assessment, 2007). A great distinction provided by the UCCFD is between learning goals and learning objectives. The difference is that “learning goals are what you hope to accomplish in your course: the overall goals that do not necessarily result in products of observable and measurable behavior”. Whereas “learning objectives are brief, clear statements about what students will be able to do when they complete instruction” (Assessment, 2007). This is significant for course design because of how concise learning objectives must be to check that students will complete the material successfully.

**Methodology**

As mentioned, learning objectives need to be observable and measurable. Learning objectives, outlined by UCCD, include the following:

\* Condition - the condition under which the student will perform the described behavior

\* Behavior - a description of a specific, observable behavior

\* Degree - the degree indicates the desired level or degree of acceptable performance

In our PY4E course, for example, the learning objective for chapter one is the following:

\* Condition: “Identify the reserved words in python”

\* Behavior: Students will be able to pick the reserved python words from a list

\* Degree: Students will be able to identify reserved words at a minimum of 80% of the time.

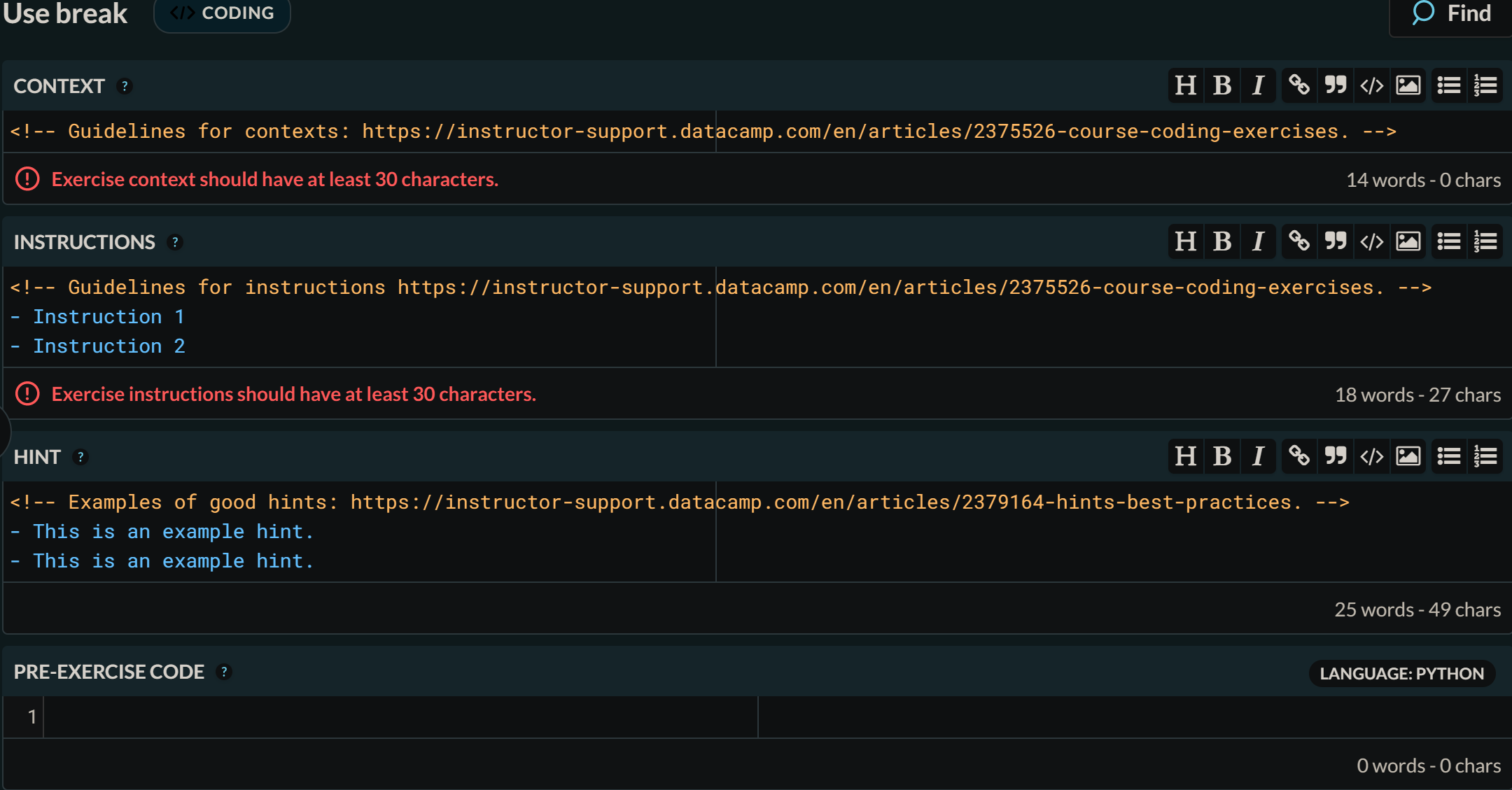
The second stage of the course design will be to create interactive activities and assessments to guide learning. DataCamp offers the ability to utilize multiple choice and console practices as both activities and assessments. Per “Bloom’s pyramid, which illustrates that thinking occurs at different levels of complexity,” multiple choice and console practices use remembering, understanding, applying, and creating processing levels (Assessment, 2007). These processing levels benefit the student because students will be tested in the same style as they are practicing. In computer science, practicing coding skill is essential to correct coding performance. Utilizing many types of process levels to achieve a learning objective may be most effective. For example, multiple-choice questions are great for an activity that requires remembering and memorizing.

Our approach to this online course is to provide a robust and constructive method to reinforce what the students need to learn. The layout and overall content of the DataCamp course is meant to be supplemental and to be a way for students to get hands on experience coding in Python with interactive elements that give useful feedback. The submission correctness tests for each module are robust and straight-forward. Students have plenty of guidance and have enough information to complete the course provided by hints and feedback that lets the student know where they went wrong.

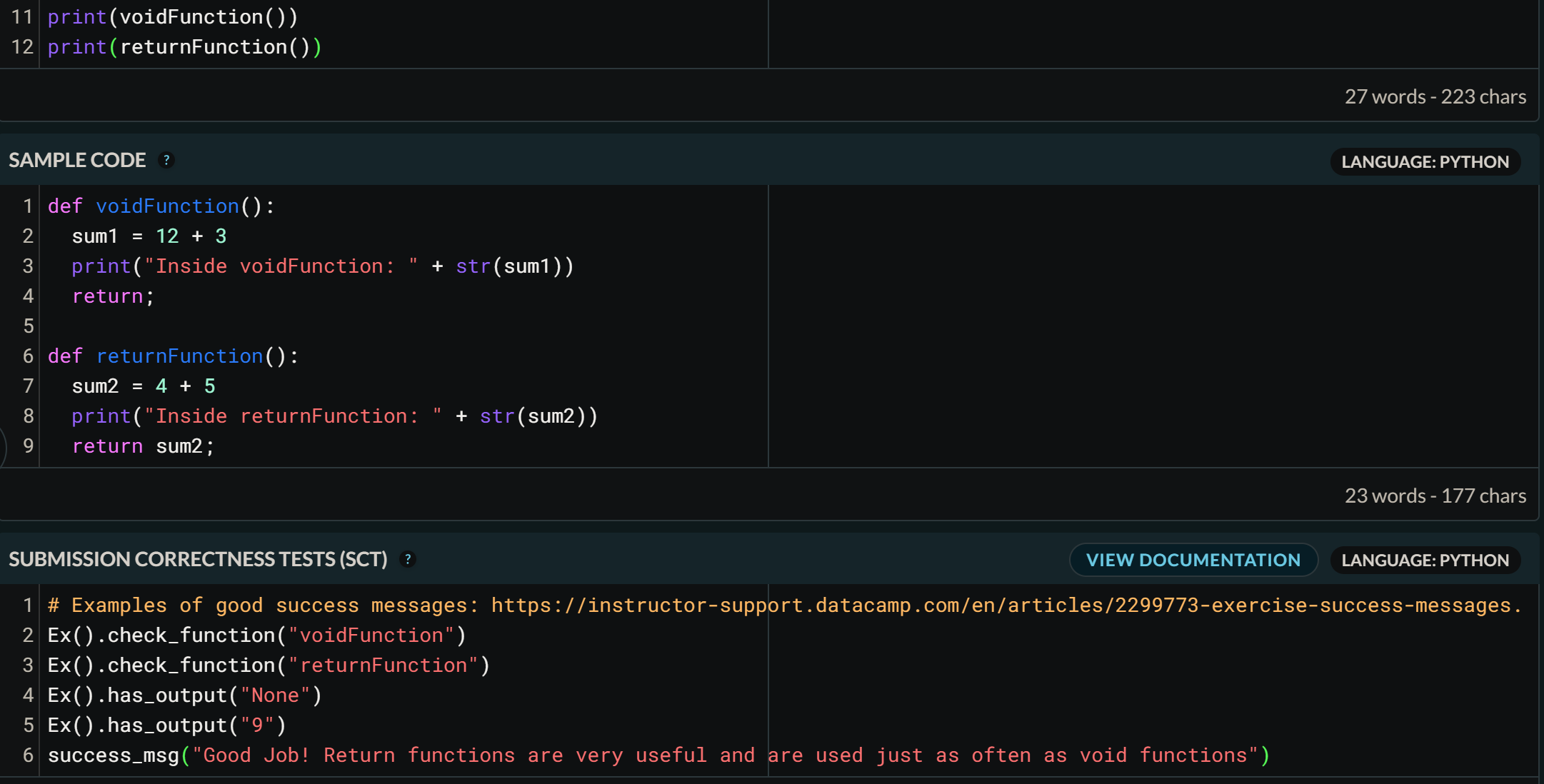
**Implementation**

The implementation will take place through GitHub and DataCamp. This will require us to understand the python commands that DataCamp recognizes and structure our activities to reflect that. Exporting our course to GitHub allows us to document and implement the material as well as allowing future input on the course. This is especially important because there is a great likelihood that Python for Everybody through DataCamp will need to be worked on by a second senior team. The way that we will handle this transition is to comment our content well, and to leave detailed instructions on the direction we were looking to head for the content.

*Figure 1: DataCamp Light Editor*



In Figure 1 we can see the layout for the DataCamp Light Editor. When we are developing exercises for students, we must design the exercise so that the instructions, hints, and sample code are useful and give the student enough context without giving away the answer. The most difficult part about this process is creating the Submission Correctness Test (SCT) code. Part of making a good course is creating robust and exact SCT code. Within the SCT, we must make sure that the student could not submit answers or code that skips or bypasses the exercise. SCT code ensures that students follow the directions, submit correct answers, and provides useful feedback when they make a mistake. We can see this process implemented in our course in Figure 2 below.

*Figure 2: DataCamp Light SCT Example*

**Experimental Results and Analysis**

We were able to create a well developed DataCamp Lite course for a teaching python to introductory programmers. Four chapters have been completed and tested by two peers who are have no prior programming experience. This course was overall successful, and we were able to create enough content to provide a useful addition to the classroom.

Our submission to Special Interest Group on Computer Science Education (SIGCSE) 2020 competition was not accepted. This was mostly due to lack of experimental results and a light literature review. We had three reviews, detailed below, that gave helpful feedback and we used this feedback to refine our paper to the best of our ability.

**Review 1**Online learning is a relevant and valuable tool in providing students with instant feedback. The authors designed and implemented a course which uses Data Camp Lite as supplemental learning material for an introductory programming course involving Python. The purpose of this course is to improve student engagement and to teach students in an accessible and engaging medium.  
----------- Problems and Motivations ----------- SCORE: 4 (Very good)  
The problem is to design and implement an interactive approach to teaching Python programming effectively, using DataCamp. The authors are very motivated to benefit future students at their own University and beyond.  
----------- Background and Related Work ----------- SCORE: 3 (Fair)  
The authors reviewed the procedures for designing a course on their campus and discovered that it requires three stages, but they only specifically mention the first two:  1)outline each chapter and write observable objectives; 2)create interactive activities and assessments to guide learning. (These design stages are sometimes referred to as: 1) Desired Results; 2) Assessment Evidence and 3) Learning Plan.) The activities in these stages would be developed in accordance with Bloom’s Taxonomy to support different levels of thinking.  
A significant portion of the paper was devoted to explaining the difference between learning goals and learning objectives and few if any activities were mentioned.  
----------- Approach and Uniqueness ----------- SCORE: 3 (Fair)  
Implementing an interactive Python course using DataCamp would be engaging and reinforce what students learned. Such a course would also possibly be unique. Unfortunately, it does not seem that the actual design of this course has yet been begun.  
----------- Results and Contribution ----------- SCORE: 2 (Poor)  
Since this course is still in the early design stage, there are no results to report.  
----------- Strengths and Weaknesses -----------  
The idea of an interactive Python course is a good one and should be pursued. A few typos need to be corrected as do a few sentences. The third design stage should be included, and the actual course design needs to be developed.  
----------- Overall evaluation -----------  
As previously mentioned, the design of this course has just barely been begun. Much more work and details would need to be included to make it presentable.  
  
**Review 2**  
The authors designed and implemented a course in DataCamp Lite as supplemental learning material for an introductory programming course involving Python.  
----------- Problems and Motivations ----------- SCORE: 4 (Very good)  
The problem is well stated  
----------- Background and Related Work ----------- SCORE: 4 (Very Good)  
Related work is reasonably well described  
----------- Approach and Uniqueness ----------- SCORE: 4 (Very good)  
The approach is relevant  
----------- Results and Contribution ----------- SCORE: 3 (Fair)  
A reasonably good contribution to the field  
----------- Strengths and Weaknesses -----------  
The authors should have utilized all space available to them.  
----------- Overall evaluation -----------  
This will make a good addition to the SIGCSE program.

**Review 3**  
The paper describes a course designed and implemented in DataCamp Lite to support of an introductory programming in Python. The material is supplemental, and the goal is improving student engagement and comprehension. The paper is mostly for instructors. The main point is to reinforce what students need to learn, focusing in interactivity and instant feedback.  
----------- Problems and Motivations ----------- SCORE: 3 (Fair)  
To engage students is a hard task. At the same time to provide instant feedback and to be able to point out students’ mistakes are important in programming classes.  In this context the paper is relevant, however the problem motivation described in the paper is really brief and not clear about the relevance of the research.  
----------- Background and Related Work ----------- SCORE: 1 (Very poor)  
The related work is almost inexistent. The paper has two references only, which doesn't gives the reviewers a clear picture of how many papers about the state of art the authors read.  
----------- Approach and Uniqueness ----------- SCORE: 2 (Poor)  
Giving the lack of related work it is hard to measure the contributions of the work.  
----------- Results and Contribution ----------- SCORE: 2 (Poor)  
The idea is interesting  but the proposal was not tested yet. The research is new, and the results are yet to be acquired. As mentioned before, the authors are not clear about their contribution.  
----------- Strengths and Weaknesses -----------  
The context of the work is relevant and the concept behind the tool is interesting.  
The weakness are the fact they don't tested their tool and the lack of details how they are going to achieve the goals they described, such as engage students and improve their learning.  
----------- Overall evaluation -----------  
The work it is in early stages and in this reviewer's opinions needs several improvements.  
The contribution must be clear since the beginning. The related work needs serious improvement, given that the number of references is small.  The lack of related works doesn't allow the reviewer to see a critical analysis of the state-of-art. They don't provide any details of how they intend to achieve their goals.  The short paper version, sometimes, sacrifices some details but it is not the case. They still had to room to provide more information. Again, the theme is relevant, but the paper needs to address these issues carefully.

**Conclusions & Future Work**

Electronic learning can be a fantastic tool for teaching and reinforcing material for students. In computer science specifically, having immediate feedback for syntax errors in programming for interpreted languages, like Python, is most valuable. This allows instructors to better teach their students and give better interactive and helpful practice.

Implementing the DataCamp course into the introductory programming class and seeing how students perform both with and without this material will determine how successful this application is. If students show a deeper understanding and perform better in their exams with this DataCamp course, then future applications of electronic learning will be developed to improve student understanding in the classroom. The course requires a substantial time commitment to create robust and comprehensive exercises, which should be kept in mind while outlining future content. We planned this project out well and managed our time proportionately. In order to complete the entire course for classroom use will require future development. This course will be tested by Dr. Harold Nelson in his future introductory python classes to determine whether the final product will be beneficial to success and learning of the students.

**References**

“Assessment & Instructional Alignment.” Introduction, 2007, www.ucdenver.edu/faculty\_staff/faculty/center-for-faculty-development/Documents/tutorials/Assessment/index.htm

Severance, Charles Russell. Python for Everybody Exploring Data Using Python 3. Create Space Independent Publishing Platform, 2016. http://do1.dr-chuck.com/pythonlearn/EN\_us/pythonlearn.pdf