Project Number: IQP-JB7

IMPROVING INTRODUCTION TO PROGRAMMING

An Interactive Qualifying Project
submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
by

Marilda Bozdo

Leonard J. Fisher

Sai Kiran Vadlamudi

Approved:

Professor Joseph E. Beck (advisor)

1 Abstract

This project attempted to improve student learning in the computer science introductory course CS 1101, Introduction to Program Design, by adding content to the video lectures used to assist in teaching concepts. These video lectures were part of a Gregor Kiczales' online course titled, "Introduction to Systematic Program Design" on Coursera, an online course offering site. The videos were then also put on YouTube to increase accessibility for those who were not signed up for the online course. The flaw with the videos, particularly when seen on YouTube, is that simply having the videos alone doesn't provide students a self-assessment. The Coursera course offered assessment questions which, while not the best, did help. In the absence of these questions, students cannot know what important concepts they did and did not miss and if they should re-watch the video. To mitigate this, a previous IQP group added guiding content for the WPI students. There were many criticisms about the added content. Our goal was to figure out ways to address some of the more important and popular criticisms. We also set out to see if our improved content has any discernible effect on the comprehension of the concepts in the video. After each video, we added a set of improved practice questions along with improved hints to allow the students to sense when they missed concepts or formulate questions to ask peers or the professor to better understand them. To measure if the practice questions were having any effect, short tests were added after the practice questions. Unfortunately we couldn't completely test our hypothesis, that these improved practice questions will improve comprehension of the videos. We did not have success in getting a large amount of students to partake in this study and so could not get conclusive results.

Contents

List of Figures

2 Acknowledgements

We'd like to thank our advisor, Professor Joseph Beck, as well as Professor Glynis Hamel, who graciously allowed us to run our study with her students. We'd also like to thank the previous IQP group who worked on this project two years ago and laid out solid groundwork on which we could build upon. Their study and results are available here:
https://www.wpi.edu/Pubs/E-project/Available/E-project-091614-191721/unrestricted/IQP1101FinalPaperVer3.pdf

3 Executive Summary

Many times we have observed our peers and experienced ourselves not being able to focus on the concepts being taught by a lecturer or simply missing one among all the other concepts. Luckily, we could assess our understanding at a later time and know when we missed a concept either during homework, if we were lucky, or unfortunately during a test. But, nonetheless, we had the chance to assess our understanding, know what concepts we missed and learn them. Now imagine we were learning these concepts through a YouTube video, there are no questions to assess ourselves and, worse, there is no way to know if we just missed a fundamental concept until it is needed, which could lead to unpleasant outcomes and a poor foundation to build on for more advanced programming concepts. Only using the Gregor Kiczales videos that are on YouTube without the other resources normally used in his course is problematic, for the aforementioned reasons. Since the YouTube videos are not interactive and have no comprehension assessment, their usefulness seems to drop compared to when used alongside guiding content. This problem of having no self-assessment or guiding content was the subject of another IQP two years ago whose paper can be found in the link on the acknowledgement section. They created questions for some of the videos which were used by the students in the course and the students left useful constructive criticism about the ASSISTments content in an exit survey conducted at the end.

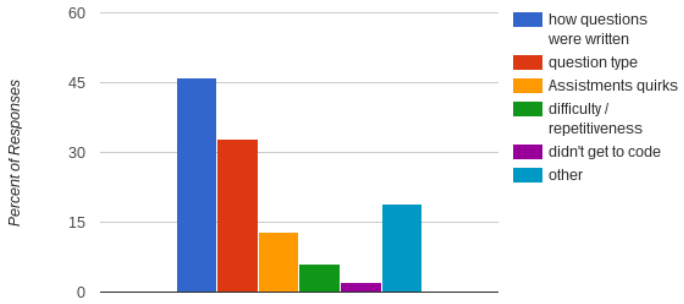


Figure 1: Student Criticisms from Previous Study

One of our main project goals for this IQP was to figure out ways to address some of the major concerns of the participants of the previous IQP. The first step was to mainly get the list of concerns. We achieved this thanks to the an exit survey conducted by the professor who had asked the question "What do you believe to be the biggest flaw of the ASSISTments system? (What annoyed you the most about ASSISTments and the assignments you got in it)". There were a hundred responses left by students on this question and some were valid concerns. Since we had a limited time frame to work on this we first tried to somewhat categorize the concerns and see what we could actually address. For example we could not fix flaws with the ASSISTments system itself and instead

simply had to work around them. Figure 1 shows our categorization of concerns presented by the participants of the previous study.

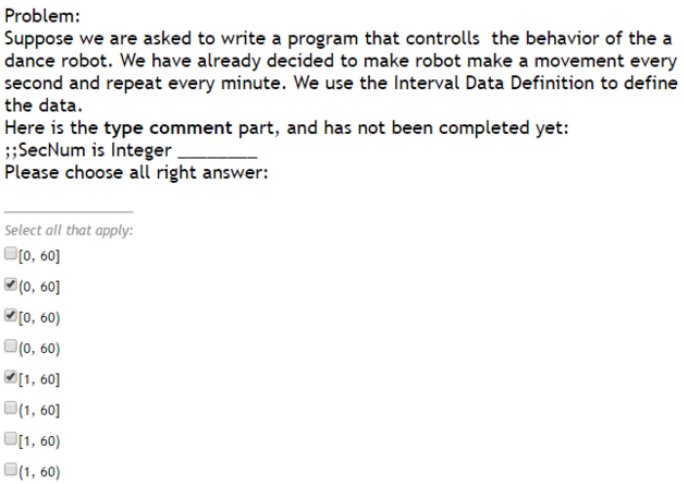


Figure 2: A "check all that apply" question

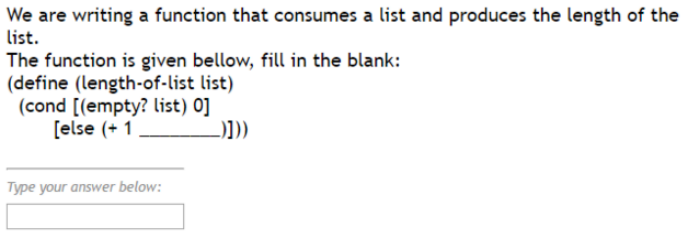
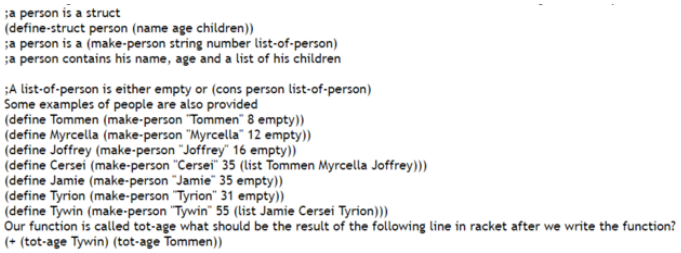


Figure 3: A "fill in the blank" question

The previous study used Check all that apply and entering the answer in a text box as the two main types of recording the answers to questions. Check all that apply questions, shown in Figure 2, are the ones where all choices which correctly answer the question need to be selected. The other questions needed to be answered by typing in the answer, shown in Figure 3, this led to issues with the multiple possibilities of answers with spaces, parenthesis etc. About 30% of the participants, as seen in Figure 1 under question type, had qualms about the check all that apply and answering in a text box along with the absence of a consistent format throughout the questions. Since fill in the blank responses have to be exact matches and several questions have multiple correct answers, this resulted in many participants marked wrong even though their answer was correct. For instance, to test the understanding of basic operators in DrRacket, one could ask to write an expression that adds 3 and 4 together. It is not hard to notice that both expressions "(+3 4)" and "(+4 3)" would be correct

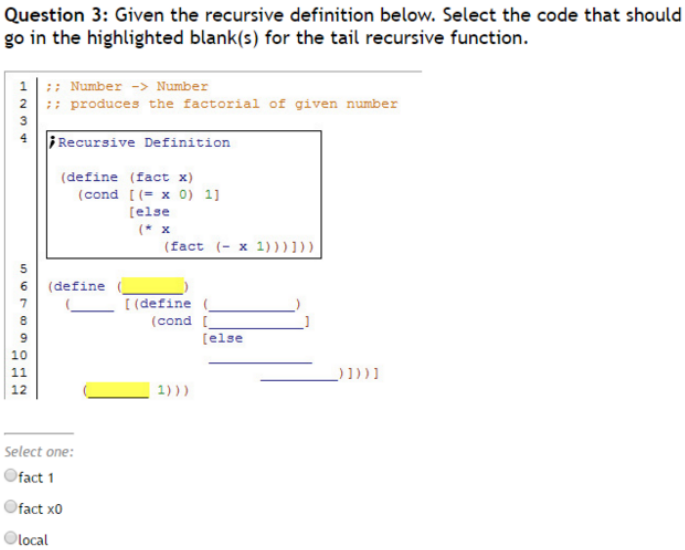
answers to this question, as well as all the expressions that could be created from these by adjusting the number of spaces. Unfortunately, there is no way in ASSISTments to mark as correct more than one of these answers. Therefore, we decided the best option is to limit the possible answer choices and since check all that apply was tried and it failed we chose multiple choice questions as the main answer entry form. Multiple choice questions may sound like easy, not worthwhile questions at first, but it is a choice we had to take to overcome the text entry grading limitations put forward by ASSISTments. Another recurring criticism from about 45% of the participants of the previous study was the way questions were written.



```
;a person is a struct
(define-struct person (name age children))
;a person is a (make-person string number list-of-person)
;a person contains his name, age and a list of his children

;A list-of-person is either empty or (cons person list-of-person)
Some examples of people are also provided
(define Tommen (make-person "Tommen" 8 empty))
(define Myrcella (make-person "Myrcella" 12 empty))
(define Joffrey (make-person "Joffrey" 16 empty))
(define Cersei (make-person "Cersei" 35 (list Tommen Myrcella Joffrey)))
(define Jamie (make-person "Jamie" 35 empty))
(define Tyrion (make-person "Tyrion" 31 empty))
(define Tywin (make-person "Tywin" 55 (list Jamie Cersei Tyrion)))
Our function is called tot-age what should be the result of the following line in racket after we write the function?
(+ (tot-age Tywin) (tot-age Tommen))
```

Figure 4: One of the older questions



Question 3: Given the recursive definition below. Select the code that should go in the highlighted blank(s) for the tail recursive function.

```
1  ;; Number -> Number
2  ;; produces the factorial of given number
3
4  ;Recursive Definition
   (define (fact x)
     (cond [(= x 0) 1]
           [else
            (* x
               (fact (- x 1)))]))
5
6  (define (_______)
7    (_____ [(define (__________)
8              (cond [__________]
9                    [else
10                    ___________
11                          _________)]))]
12   (_______ 1)))
```

Select one:

- fact 1
- fact x0
- local

Figure 5: A newer question with updated formatting

The questions written by the previous IQP were wordy and didn't separate the actual question, the information needed to answer the question and the instructions on how to answer the questions. An example of one of these poorly

formatted questions can be seen in Figure 4. The information needed is the same font, size, style as the actual question, second to last line, and so hard to separate the parts of the question visually causing potential confusion. To address this concern, we formatted the information using DrRacket, the program the students use to write actual code in the course. This makes the information visually familiar, since they use this format when doing homework assignments, labs etc. An example of the improved question format can be seen in Figure 5 where the question is at the beginning, with all the necessary information and code formatted clearly as they would see in the course and in their homework assignments.

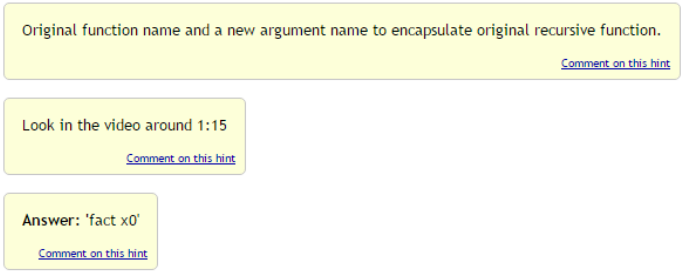


Figure 6: Hints for the question in Figure 5

Another improvement to the questions, which we thought would help understand the concepts better, was providing valuable hints. The previous IQP group considered the fact that students would not be able to continue without answering correctly. They therefore gave the answer to those who were stuck in the form of a hint. They did not get a chance, probably due to time constraints, to give more hints for each question before the answer. Our group believes hints are valuable to guide students to the solution improving comprehension of the content. So, we added more hints to our questions to help students better understand the answer and the concepts behind it. Some hints contain the time in the video where the question and its solution are explained. Some hints are based on the concepts. The last hints for each question contain the answer. An example of this can be seen in Figure 6. We planned to provide 3–4 hints per question, including the answer, which seemed to be a reasonable amount of help without providing the answer.

We ran an experimental study to test if these changes would affect the comprehension of the concepts taught in the video. The participants of this study were the students enrolled in Introduction to Program Design course offered in C-Term of 2016. The students are randomly assigned into two groups, control and experimental. The control group simply watch the video and take a comprehension assessment test providing a baseline of comprehension provided by the video alone. The experimental group has a set of practice questions which provide a way for the participants to self-assess and see what concepts they understand and don't, and provides a way to improve their comprehen-

sion on those concepts they missed in the video. They then have to take the same set of comprehension assessment questions as the control group to see if the experimental group has improved comprehension after the set of practice questions.

Due to misunderstandings between our IQP group and the instructor of the course we were running the study in, we got extremely poor participation rate. Therefore we could not get conclusive results either showing or not showing that our set of practice questions provided significant benefits on top of the video lectures.

4 Introduction

New ways of teaching are always in development. In the 1960s, America noticed that other countries had children who were doing better in mathematics, so American grade schools adopted the "New Math." The goal of that change was to move the focus from memorizing computation tables towards learning more conceptual aspects of mathematics, such as basic set theory. Today, the Internet is changing how we do almost everything, and education is not an exception.

It has become very easy for people to make videos and put them on the Internet. Some teachers record their lectures so that their students can watch lectures from their homes. Recently, many teachers have started to create Massive Open Online Courses, or MOOCs, which are classes available for anyone with a computer. Instead of going to an expensive school to learn various topics, students can look online for MOOCs on almost any topic. Unfortunately, just as not all teachers are good at explaining concepts to their classes, not every MOOC is good at teaching.

Most MOOCs are a combination of videos and questions to teach the subject matter. Videos vary in length, since online courses don't need to have a fixed length for each class. Rather than a traditional class in which everyone meets at a certain time on certain days, an instructor can prepare several videos and let the students view them at their own pace. Exams usually cover the material recently covered, and they serve to help students find out what they already know and what they should go back are try to better understand. At least, that's the way it should work.

In practice, writing good questions is really hard. In order for MOOCs to be open to massive amounts of students, a computer needs to know if answers are correct or incorrect. An easy way to do this is to have every question be multiple choice. Unfortunately, with multiple choice questions, it's important that the wrong answers be believable. Sometimes, answers are written in a way that a smart person who knows none of the subject matter can still identify the correct answer. When the questions can be answered without knowing the subject matter, they no longer serve to illustrate to students the topics that aren't understood.

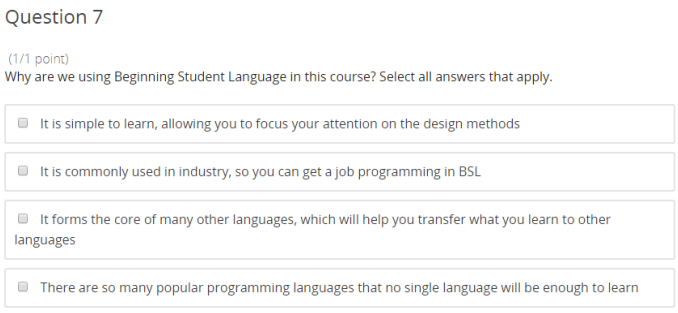


Figure 7: An easily answered question

Figure 7 is a question from the Introduction to Program Design MOOC, that was initially on Coursera, and is currently being offered on edX. Even without watching any of the videos, answering it is quite easy. A language called "Beginning Student Language" is going to be easy to learn, but it won't be common in industry. It may or may not "form the core" of other languages, whatever that means, but it sounds like a good reason to pick a language. Lastly, if someone has heard of Java and C and knows that those are both common languages, they can guess that there is not a single language that is sufficient to learn. While option 3 is not obviously correct, the student is given multiple attempts, and someone who doesn't know how to program and has never seen the videos of the course before can very likely answer this in two tries.

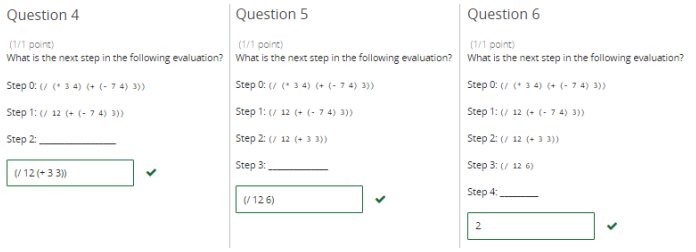


Figure 8: A series of questions from the MOOC

There is another unfortunate feature of edX, where nothing prevents students from reading later questions and answering questions out-of-order. In the second set of questions, questions 5 and 6 show the answers to questions 4 and 5, and all three are presented together before any answers need to be filled in. See figure 8. Both of these are big problems with some MOOCs. In order to adapt the videos for a course at WPI, the problems need to be rewritten.

At WPI, CS 1101, Introduction to Program Design, is a course that many students take. In C term of 2016, 131 students took the class, of which 65 were either Computer Science, Robotics, or Game Design majors. Many of the professors for the course have started to integrate the use of a MOOC into class. The class is currently taught with video lectures assigned as homework, and short quizzes each day that class is scheduled to meet. After the quizzes, students can ask for more help understanding certain topics that were introduced in the videos.

Unfortunately, the MOOC that is being used, while it has very nice videos, does not have good questions. The MOOC was originally hosted on the website www.coursera.org, and has since moved to www.edx.org. Both websites compile large number of courses so that students don't need to do as much searching to find MOOCs for subjects they want to learn. This MOOC also has its videos hosted on YouTube, and that is what CS 1101 uses (in the academic terms we are analyzing in this document).

Last year, a team of students worked to create questions to go along with the videos. Their research led to ideas about how questions should be written

and how many questions should accompany each video. Their content was given to students taking CS 1101, and feedback was mostly positive. However, aside from a survey at the end of the class, their study had no way to measure success of the content.

For this project, we are looking at the feedback provided to the previous team, and we are improving the questions based on that feedback. We also created post-tests after several videos to see if students who had intermittent questions did better on the post-test than students who watched the videos without added questions. We were hoping to see a significant increase in scores for students who had questions. Unfortunately, not enough students participated in our study thus resulting in a lack of data for conclusive results, and our only results come from a handful of students who gave us feedback on the questions.

5 Problems With Original Content

The previous IQP group worked in making the transition between videos and the material offered in class, exams and homework, smoother. They used a software called ASSISTments, which was developed at WPI in collaboration with Carnegie Mellon, and created a number of questions, to help the students practice the material introduced in the videos.They made a number of decisions about the format and the content of the questions which, in their opinion, would benefit students the most.

First and most importantly, this IQP group created questions of a unique nature. These questions had to have a sensible length and complexity, so that students would find them both interesting and helpful in retaining more information. Also, the answers to these questions had to be supported by ASSISTments software. For instance, the questions couldn't be very intense in coding, as "Fill in the blanks" questions in ASSISTments accept very specific answers, including the exact spacing in the code, so, asking the students to fill in a large piece of code would result in problems. However, the importance of the "Fill in the blanks" questions cannot be neglected as they allow students to practice writing code, which is a very useful skill in learning programming. Therefore, the previous IQP group created questions which would have a short and distinct answer, so that students could have the opportunity to practice coding in little-chunks. Also, they came up with a number of conceptual questions testing the understanding of difficult concepts in each video.

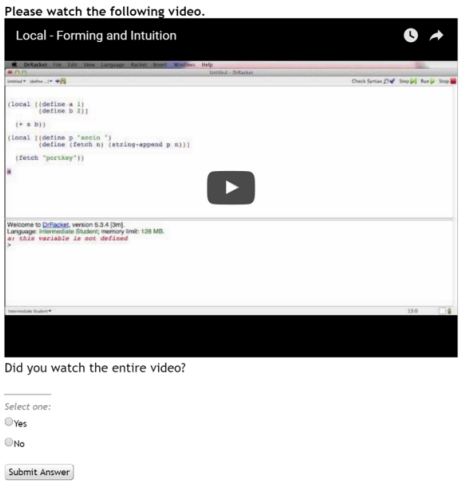


Figure 9: A video embedded in an Assistments question

The previous IQP experimented with many different ways of creating the content and came up with some final design strategies. In terms of dividing the content they reasoned it is better to make a set of questions for each video rather than one set for all videos pertaining to a single concept. This would make it

easier for different professors to use the content, even if they wanted to exclude certain videos or present them in a different order. They decided to limit the number of questions for each video to ten, as to not discourage the students with the length of the assignment. In the interest of the students, they embedded the video in the problem set rather than simply providing the link to help preserve focus. An example of the embedded video can be seen in Figure 9. The content was all on a single page allowing the previous questions and answers to be visible, so the students could build on their knowledge. Every problem set would start with the question "Did you watch the video?" with the correct answer set as "Yes". This alerted the students that videos must be watched before attempting the questions and that a video is supposed to appear, and if it didn't they should inform the professor. Furthermore, it was decided that longer videos should be partitioned and questions should be created for each partition, so that students would not lose interest. Lastly, the questions were typed in Microsoft Word or similar editors, to avoid spelling and grammar mistakes. The last question was designed to ask students to rate their understanding of the concepts in a scale from 1 to 10, providing the professor with valuable information on which topics to provide more guidance on during class.

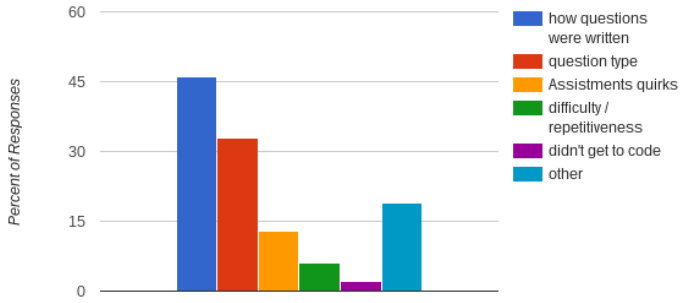


Figure 10: Student Criticisms from Previous Study

We believe that the previous IQP group did a good job in creating questions which would increase student comprehension along with retention of the content. However, according to the students' feedback, which can be seen in Figure 10, there were a number of ways the content could be improved.

Although typing in Microsoft Word helped with grammar and spelling mistakes, the visual look of the code is very different from how the questions look when typed in the software used for the course, DrRacket. We believe that the formatting was the reason about 45% of the students wrote "how questions were written" as a concern. The questions from the previous IQP appeared wordy sometimes and not very well organized, as can be seen in Figure 11. The actual question is not clearly visible at first glance, the information needed to answer does not appear as it would if it was written in DrRacket.

Another major concern with the previous IQP content, as expressed by the student participants, was the overuse of the check all that apply questions.

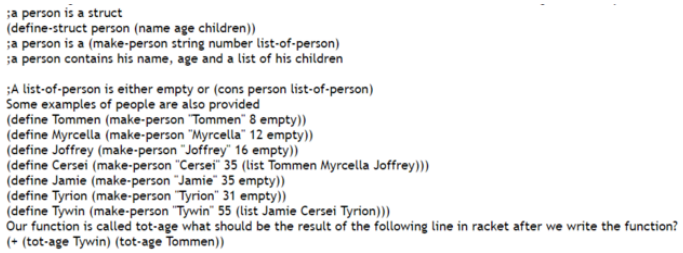


Figure 11: A difficult-to-read question

These questions would seem repetitive and when you missed one choice, it's hard to see which one you missed or got wrong. ASSISTments also has an issue with fill-in-the-blank questions since each possible answer needs to be entered. This led to the frustration of many students since a possible correct solution is not marked as correct, students could get a correct solution marked as wrong. About 30% of the students had concerns about the check all that apply and the fill in the blank questions for the aforementioned reasons.

Although ASSISTments supports it, there were no explanatory hints designed to help students have a better comprehension of the questions and build their own path towards the solution. The previous IQP content provided no hints for the questions and instead simply gave the answer to the students who asked for a hint. This was to overcome an issue with ASSISTments where the student would not be allowed to continue without giving the correct answer. Giving the answer also does have some significance since it is similar to looking at the back of a textbook for the answer and working out the steps from the answer. A couple of the students wished for more hints before the answer and we also believed it is important to have more hints to improve comprehension of the concepts. When a student is stuck, guiding them to the answer and having them learn along the way is much better for those who did not understand or missed the concepts in the videos.

We agree with the previous IQP group that it is necessary to create a question that will alert students that there is a video that they are supposed to see. However, simply asking them every time if they watched the video does not allow us to gather any information from the answer of the question, due to it simply being a yes or no answer. We thought that it would be more beneficial to figure out a way to ask a similar question which would both collect useful information and still inform students that there is supposed to be a video which needs to be watched. As for the last question of the video, we thought that if we asked students an open answer question on how they felt about the content and their feedback, we would gather much more data than from simply looking at a rating of their understanding.

Last but not least, there was no assessment of the questions themselves so that the IQP group could understand if and when the questions designed helped the students understand the content more. It is essential for the IQP group to

understand the value of the questions they are designing and improve them if needed to make possible that more information is retained from the students.

6 Modified Content

Since we couldn't create questions for all the sections covered in the video lectures we discussed which of the sections are most important with the two professors teaching the course and concluded that it would be best to create the improved questions for How to Design Functions and Self-Reference.

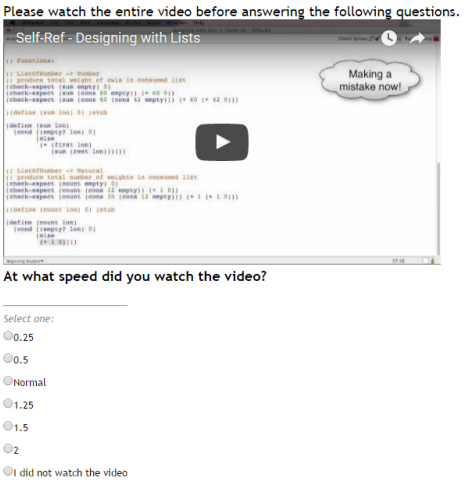


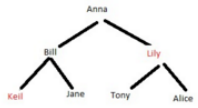
Figure 12: A video embedded in an Assistments question

One of the modifications, as we discussed above, is changing the question about watching the video. It is still important to have the question there to alert students that there is an embedded video there but simply asking a yes or no question seemed wasteful. We altered this question to instead ask 'At what speed did you watch the video?', since YouTube allows the videos to be played faster or slower than intended. We believed this would show any correlation between how fast a student watched the video and how well they understood the concepts. An example of this can be seen in Figure 12 and the original format can be seen in Figure 9.

One of the issues we discussed with the existing content was the type of the questions. The participants of the previous study showed concern with both check all that apply and fill in the blank questions. This left us with multiple choice questions as the main type of questions for our new content. Since the fill in the blank also lead to problems when trying to define all the possibilities of correct answers, we tried to limit their use if needed and tried to not use them at all. Our questions mainly consisted of multiple choice to overcome these limitations and concerns.

Another issue is the way the questions were written, Figure 13 is an example of the original content from the previous IQP. As you can see in the original content, the question is buried in the description with the same font as the information needed to answer the question. The directions to answer the question

Suppose we have a family tree, for example:



Anna is the child of Bill and Lily, Bill is the child of Keil (dead) and Jane, Lily (dead) is child of Tony and Alice

Here is the defination of the data:
(define-struct child (name age isalive parents))
;;child is (make-child String Integer Boolean ListOfchild))
Now a child is planning to have a trip to Hawaii with his/her parents and grandparents, he/she need to know the total number of people.
We need a function that consumes a child and return the number people

Here is the testcase:
What is the expacted output of the testcases blow?
(check-expect (sum--parents Anna) ___)
(check-expect (sum--parents Bill) ___)
(check-expect (sum--parents Tony) ___)
Please give the answer in order and saperate them with only one space

Figure 13: A question from the previous study

Question 5: Given the information below, select the choice which best fits in the highlighted blank.

```
;; Tests
(check-expect (numPoints empty) 0)
(check-expect (numPoints (cons 80 empty)) (+ 80 0))
(check-expect (numPoints (cons 80 (cons 90 empty))) (+80 (+ 90 0)))

(define (numPoints lon)
  (cond [(empty? lon) ____]
        [else
         (____ (first lon)
               (numPoints (rest lon)))]))
```

Select one:

- (first lon)
- 0
- (rest lon)
- 1

Figure 14: The new way to fill in the blank

Question 1: Which of the following has a well-formed self-referential data definition?

```
A. ;; ListOfString is one of:
   ;;  - empty
   ;;  - String

B. ;; ListOfNumber is one of:
   ;;  - empty
   ;;  - (cons Number ListOfNumber)

C. ;; ListOfString is one of:
   ;;  - empty
   ;;  - (cons Number ListOfString)

D. ;; ListOfNumber is one of:
   ;;  - empty
   ;;  - Number
```

Figure 15: Another new question

are also buried in the information and is inconsistent with the answer format for the other questions. Figure 14 and Figure 15 are examples of our new content. In the new content the information needed to answer the question and the answers are formatted as they are shown in the video lectures, and in DrRacket, to make the content clear. Also we used multiple choice answer entry throughout our content to keep the answer entry consistent. Figure 14 also shows how we tried to incorporate some fill in the blank type questions.

The question is stated clearly at the top. The necessary information is a picture of it written in DrRacket. This makes the information familiar as it is presented in the way it is in the course. The possible choices for the answers are also limited to combat the confusion with text entry in ASSISTments.

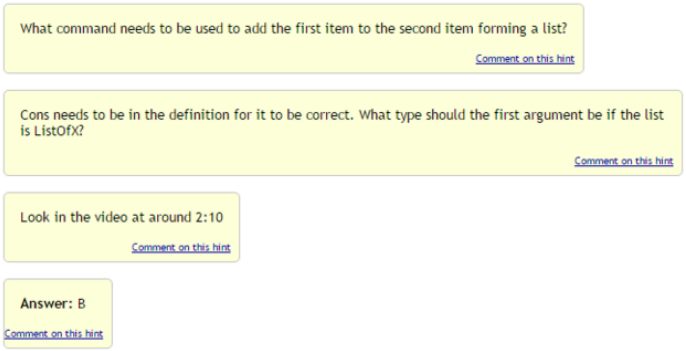


Figure 16: Hints for the question in Figure 15

In terms of hints we added three to four hints for each question. Some directed the student to the time in the video where the concept was taught, others guided the student to the solution by explaining the concepts clearly and concisely and the last hint gave the answer. An example of these hints is shown in Figure 16 which are hints given to the question in Figure 15. The last hint provided the answer, since you cannot proceed to the next question without answering correctly.

7 Methodology

The platform for running our study remained as ASSISTments for several reasons. The original intent of this study was to adapt the Introduction to Systematic Program Design course to ASSISTments and was funded to do so by the Gates foundation. The previous IQP used this platform, providing us examples of creating content using this system. If there were issues we knew who to ask for help, since the system was co-developed by a professor at WPI, thereby avoiding long response time in assistance which could cause problems during the study. The participants of the previous study as well the group running the IQP provided us the positives along with the problems with the system, so we knew when to work around issues.

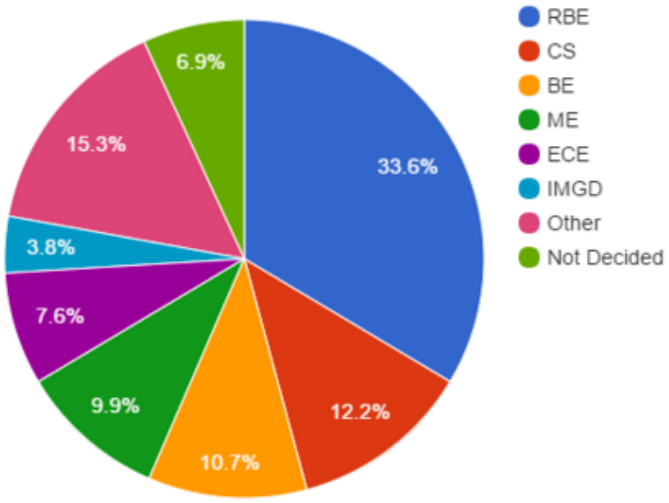


Figure 17: Majors of students who took CS 1101 in C Term 2016

The participants for the study are WPI undergraduate students taking Introduction to Program Design, an introductory computer science course. There are a total of 131 students in the course with a set of 16 majors including those undecided, as seen in Figure 17. Due to this course being taken by students of many majors, the participants of this study seem to be a representative sample of WPI students. To simplify the process for the participants, the ASSISTments accounts were created by the IQP group on their behalf. Similar to many experimental study the students were then randomly assigned into two groups either control or experimental.

The assignments for the control group consisted of simply watching the video lectures and taking a three to four question assessment of their understanding. This provides a baseline of how well the main concepts are being understood by the students by just watching the video lecture. The experimental group first is prompted to watch the video lecture and then have five to six guiding questions

which guide them to the important concepts that are discussed in the video. Then they have to complete the same assessment as the control group.

The assessment score for the control group is the baseline of just watching the videos alone. The assessment score for the experimental group is a quantitative measure of the effectiveness of the added questions. By comparing the scores of the experimental group with the baseline group it is possible to see if the added question are affecting comprehension in a significant manner.

8 Conclusion

Our study did not produce enough data to have conclusive results. The project advisor and the professor instructing CS 1101 were not the same, and this led to some level of miscommunication about goals. Unfortunately, our hypothesis could not be tested with the number of responses we received.

For future work, a group of students could run the study we designed. To them, we have some recommendations. First and most important, do not run the study unless the project advisor and the course instructor are the same professor. It is too easy for misunderstandings to lead to a lack of data and a failed study. Before using the content we created, we recommend finding a way to prevent ASSISTments from marking an answer as "wrong" when a student requests a hint.

If it becomes feasible, one type of question that would be more useful on a future study would be an open response question where students write code to complete part of a program. One thing that many students would like is more opportunities to write code. If ASSISTments had a way to interpret and test Racket code, this could be implemented in ASSISTments. An instructor could write part of a function and leave parts of the functionality missing. This file could then be given to the students to finish and upload at which point the the code would be run to make sure it works as expected.

Since CS 1101 is intended to teach how to design programs beginning with good comments, it would also be nice to test students on their abilities to comment code. An interesting question could ask the student to write a function and have them upload it. Then the uploaded code could be analyzed for proper commenting and scored appropriately. There is not a good way for ASSISTments to grade this, and it seems to be even more difficult to implement than a Racket interpreter due to many variations of student writing.

If a study were to be done over the course of more than a term, it could be interesting to look at how students do in later courses if they had more help in CS 1101. Since learning to program is cumulative, being comfortable with the basics is important towards being able to write more complicated programs. However, this study would need to observe students for far more than a term, and it would probably need to track their grades.