Assignment 9: Human Subjects Study  
Assigned: Friday, March 20th  
Due: Tuesday, March 31 or April 7, 11:59pm

Policy

This assignment may be done alone or in pairs. Team members should state that they are collaborating and turn in the same document. Expectations for the scope of the study will be scaled accordingly.

You must turn in either Assignment 9 or Assignment 10 on March 31st. The other one can be turned in on April 7th.

Logistics

All clarifications (if any) to this assignment will be posted on the class discussion board on [Piazza](https://piazza.com). Any revisions will be uploaded to the “assignments” page on the class web page.

Learning Objectives of This Assignment

In this assignment, you will apply what you have learned about human subjects studies of programming language semantics to carry out a study to answer a research question about your programming language design.

Part I: Refine Study Design

Take into account the feedback you got on assignment 7. Use it to improve your study design. Also, given that we are all socially distancing, consider any changes you will have to make so that it is possible to give your study to a friend or classmate over the internet. Create final versions of your study materials based on these criteria.

Part II: Carry Out Your Study

Identify some friends or colleagues with whom you can carry out your study. Schedule an appropriate amount of time and run the study with them, collecting data according to your plan.

We would like you to have the opportunity to publish your work if you get interesting results—you could do this on your own or together with the instructor and other classmates. To take advantage of this opportunity, please read the study design approved by the CMU IRB, which is posted to the course website, and make sure you follow the approach written there. Take special note of the following:

1 Any publication you may wish to do is completely optional and will not affect your performance in the course.
• Use the following approved language in any advertisements: “For a class assignment, I am looking for programmers to participate in a short research study involving some programming tasks. The study will take no more than (three) hours. If you are at least 18 years old (and have at least a year of Java experience), I would appreciate your help.” You may replace the text in the parentheses with any special conditions that reflect the specific needs and plan of your study. For example, while we said these studies would take a maximum of three hours, many of your studies will take much less time, so adjust the maximum appropriately in your own advertisement.

• Be sure to collect a consent form (also posted on the website) from your participants.

• It is unlikely that anything bad will happen during the study, because all of our studies are minimal risk. However, if something does, we’re obligated to report it to the IRB. You can do so directly or I can help you with it.

• All audio-recording must take place somewhere where you won’t accidentally record non-participants (i.e., don’t do it in public). Given that most recording will be done over the internet, this is likely to be OK, but keep in mind your environment and the environment of your subject and make sure you have adequate privacy.

• All records from the research must be kept for 3 years. I can help with that if you want me to, or you can keep it securely yourself.

None of the above restrictions are strictly required if you don’t want the ability to publish your work (though you still need to be careful with audio recording; some states, including PA, require consent from everyone involved). However, they are good practices anyway, so we suggest you follow them regardless.

Part III: Analysis, Reporting, and Discussion

Look at the data you have gathered. Consider the following kinds of questions and analysis techniques, and apply them where applicable:

• If you gave people tasks to carry out with a language (real or hypothetical), did you find that there were barriers for users? If so, describe them.

• If you did a natural programming study, what kinds of approaches were suggested? If you had more than a couple of participants, were some suggested more commonly than others?

• Does your data suggest hypotheses about your language or the domain of programming you are working in? What are they?

• Did some users do the task differently than others? Can you relate how people did the task to characteristics of the people? (e.g. novices did one thing, and experts another)

• Are there numerical metrics of success (e.g. how many subjects succeeded or failed at a task? how many portions did each subject complete? how long did it take? do they like your system?) If so, report them.

• Consider using qualitative analysis methods, if applicable. For example, write down snippets of text that your users wrote, or things they said in a think-aloud study, and “card-sort” them into meaningful categories. What can you learn from those categories?
Report interesting data and trends that you find. But don’t over-interpret your data: it’s fine to say “subject 3 encountered barrier X” but, unless you study many more subjects than we expect you to, it’s unlikely you’ll be able to support a conclusion like “we are confident that developers are more productive when using language construct X than when using construct Y.” Save that one for a future study! In the meantime, you can certainly hypothesize that kind of relationship, without claiming conclusive evidence for its truth.

Discuss the limitations of your study. What might cause the hypotheses you developed to be wrong? What would you have liked to learn but didn’t? Did something not go as well as you liked? Part of scientific maturity is being up front about the imperfections in the science that you do; this can help you (and others) to improve the science the next time the question is studied, and it also gives readers confidence that the things you did report are worth paying attention to.

What changes will you make to your language as a result of your study?

What are the implications, if any, on existing languages (with which you are already familiar)? For example, might you propose changes to JavaScript as a result of what you learned?

What future research questions would you ask, if you had more time?

Write up all of this in a PDF report and turn it in on Canvas.