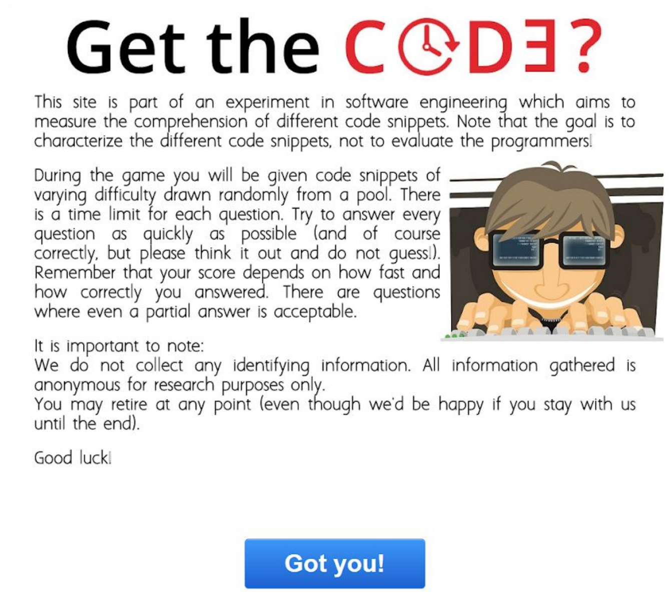
Detailed description of the experiment flow

The first screen includes an explanation about the experiment:



Get the C⏱D∃?

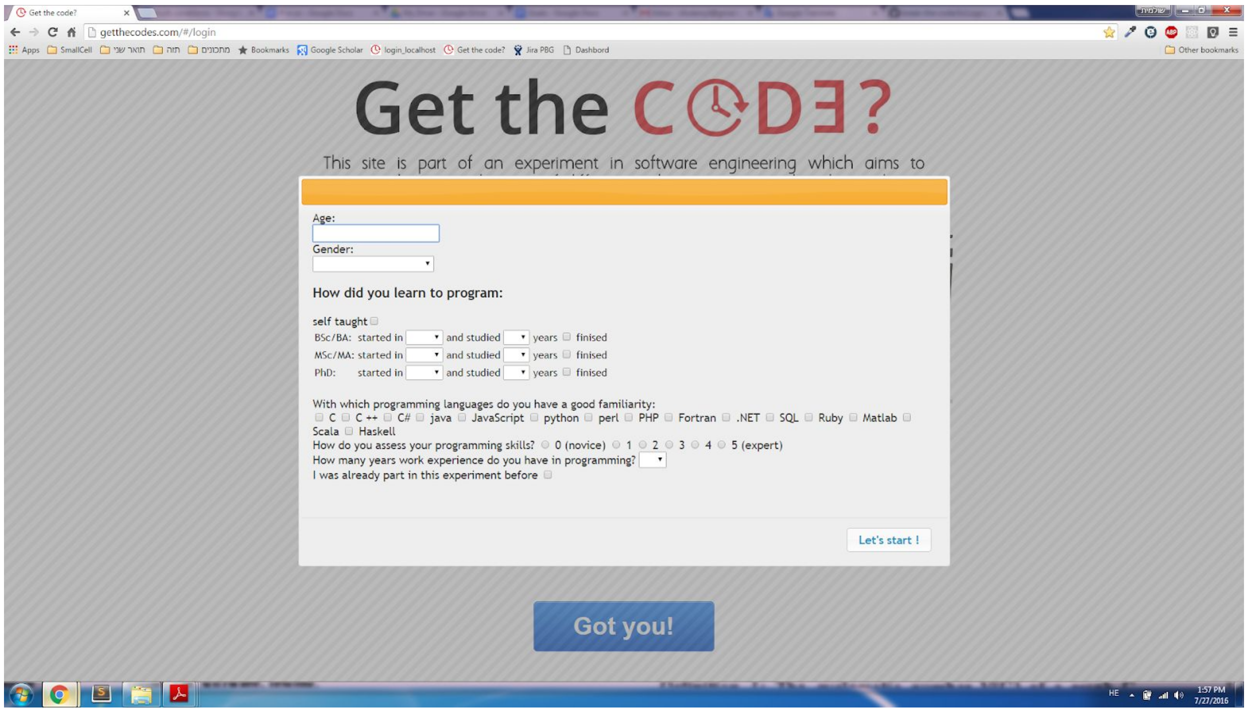
This site is part of an experiment in software engineering which aims to measure the comprehension of different code snippets. Note that the goal is to characterize the different code snippets, not to evaluate the programmers!

During the game you will be given code snippets of varying difficulty drawn randomly from a pool. There is a time limit for each question. Try to answer every question as quickly as possible (and of course correctly, but please think it out and do not guess!). Remember that your score depends on how fast and how correctly you answered. There are questions where even a partial answer is acceptable.

It is important to note:
We do not collect any identifying information. All information gathered is anonymous for research purposes only.
You may retire at any point (even though we'd be happy if you stay with us until the end).

Good luck!

The second screen - a short demographic questionnaire that focuses on the experimenter's programming experience:



getthecodes.com/#/login
Get the C⏱D∃?
This site is part of an experiment in software engineering which aims to

Age:

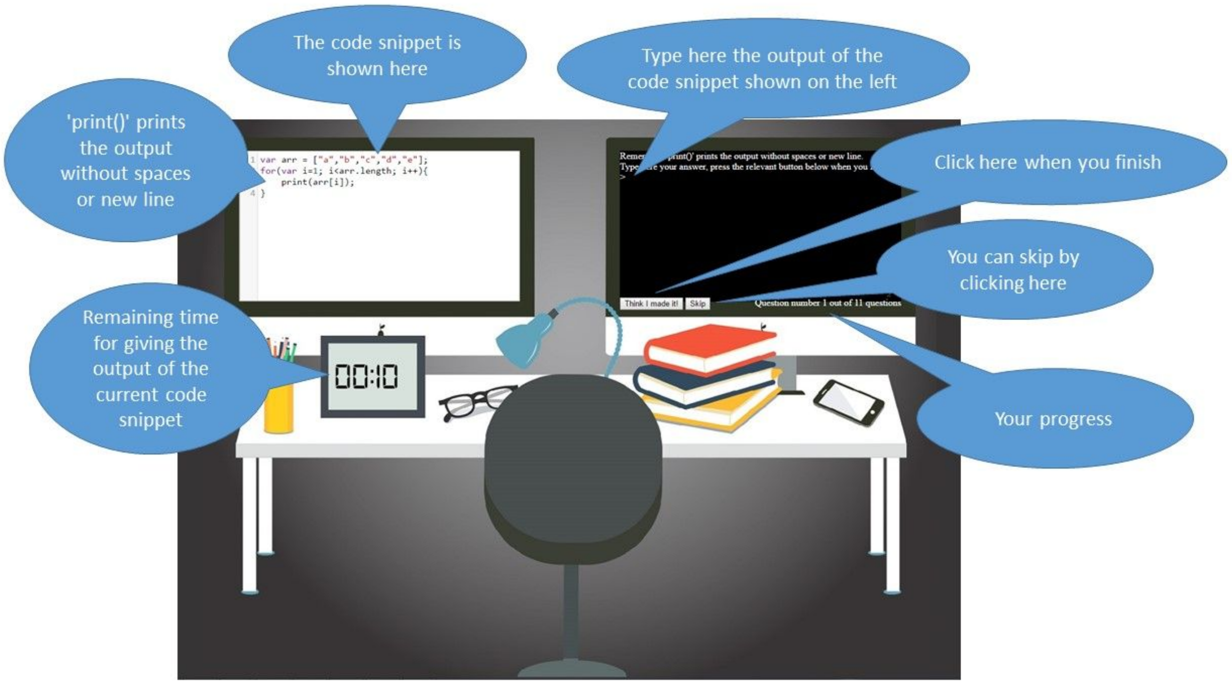
Gender:

How did you learn to program:

self taught ☐
BSc/BA: started in ▾ and studied ▾ years ☐ finised
MSc/MA: started in ▾ and studied ▾ years ☐ finised
PhD: started in ▾ and studied ▾ years ☐ finised

With which programming languages do you have a good familiarity:
☐ C ☐ C ++ ☐ C# ☐ java ☐ JavaScript ☐ python ☐ perl ☐ PHP ☐ Fortran ☐ .NET ☐ SQL ☐ Ruby ☐ Matlab ☐ Scala ☐ Haskell
How do you assess your programming skills? ○ 0 (novice) ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 (expert)
How many years work experience do you have in programming? ▾
I was already part in this experiment before ☐

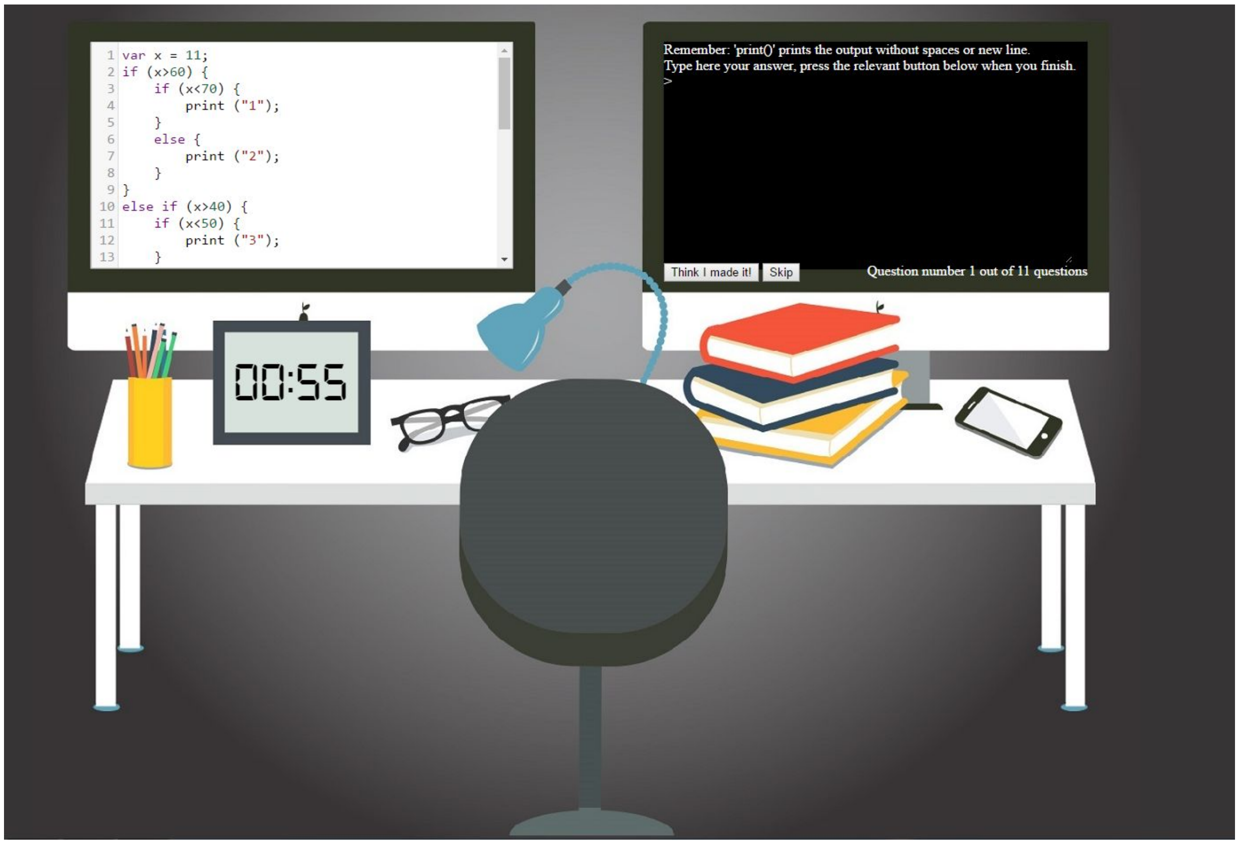
The third screen - detailed explanation about the experiment screen:


The code snippet is shown here
Type here the output of the code snippet shown on the left
'print()' prints the output without spaces or new line
Click here when you finish
You can skip by clicking here
Remaining time for giving the output of the current code snippet
Your progress
00:10
Think I made it!
Skip

```
var arr = ["a","b","c","d","e"];
for(var i=1; i<arr.length; i++){
    print(arr[i]);
}
```

Question number 1 out of 11 questions

The fourth screen - the experiment:



```
var x = 11;
if (x>60) {
    if (x<70) {
        print ("1");
    }
    else {
        print ("2");
    }
}
else if (x>40) {
    if (x<50) {
        print ("3");
    }
```

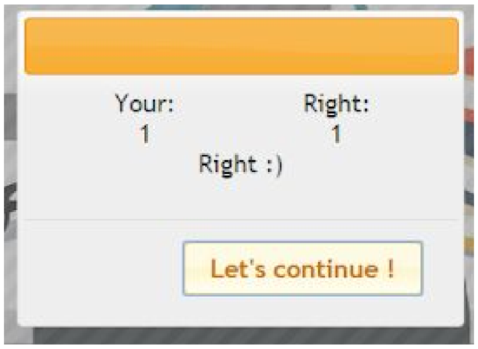
Remember: 'print()' prints the output without spaces or new line.
Type here your answer, press the relevant button below when you finish.
>

Question number 1 out of 11 questions

On this screen, the experimenter gets a code snippet that is presented on the white window to the left. The experimenter should write the output of this code snippet in the black window to the right. From reviews we received from the pilot we conducted, we learned that graphics add a lot to the gamification experience, and creates a sense of fun on the experimenter's side. Aside for the two mentioned windows that appear on this screen, there are four additional important components on it:

- "I think I made it" button and "skip" button: the experimenter has the option to skip a question or press the first button if she or he believes that they solved it. We added the "skip" button after feedbacks we got from the pilot about experimenters testifying they guessed in some of the question where for example the question was too difficult for them or when they got tired of the experiment, and we wanted to try to reduce the guesses.
- The clock that counts back: since time is a major component in measuring the complexity, and we wanted to invoke pressure among the experimenters to answer the questions as fast as they can, we added this clock. In most of the questions the clock counts back 60 seconds. For questions where some experimenters could not answer within a minute, we gave 90 seconds. In the pilot we gave different times to answer different questions, but it caused the experimenters to notice that there are relatively short questions of 30 seconds, and relatively long questions of 60 seconds, and there were more skips in advance on the longer questions because some of the experimenters labeled question that take more time as difficult ones. In addition, we noticed in the pilot that the very existence of a countdown clock put pressure on the experimenters.
- The progress component: in which question in the sequence of questions is the experimenter now. From observations we conducted in the pilot, it was very important for the experimenters to know how many questions were left for the experiment to be over. In the long experiments we conducted, some of the experimenters that leff reported that they felt like they are not progressing, and thought that maybe the program got stuck, or that it is an infinite questionnaire. Hence, we decided to add this component in order to give the experimenters the sense that they are progressing, and solve the sense of uncertainty of how many questions are left.

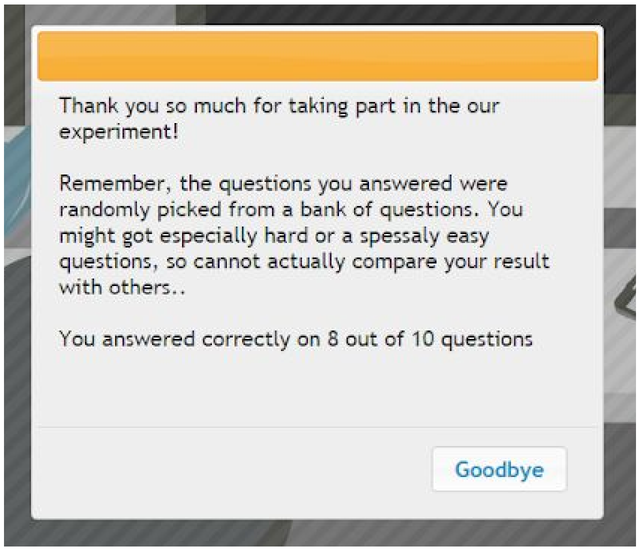
When the time is over or when the experimenter presses one of the buttons, a question summary window is opened. In the window the experimenters can see if they answered correctly. The experimenters can continue to the next question whenever they desire. It was important to us that the experimenters could control the time they have for going from one question to the next one. If at some point they cannot continue, but they might want to proceed afterwards, they have such an option.



It is important to note here that the result of the experimenter, i.e. its answer and the time it took her to answer, are sent to a server after right after each question. In this way, if the experimenter decided to quit the experiment, we can still use the results from what she already did. In addition, it is important to note that the correct answer to the question does not reach the browser at the beginning of the question, but in the end, so experimenters with web skills could not peek in the console to see what the correct answers are.

In the question summary window, we show the experimenter's answer against the correct answer. The questions still appears on the background so the experimenter can understand what was wrong in case she was mistaken. The textual feedback to the experimenter is drawn from some feedbacks entered in advance, like: "Wow", "Nice :)", "!Good!".

When the experimenter finishes the test plan that was prepared for her, the following window is presented:



Thank you so much for taking part in the our experiment!

Remember, the questions you answered were randomly picked from a bank of questions. You might got especially hard or a spessaly easy questions, so cannot actually compare your result with others..

You answered correctly on 8 out of 10 questions

This summary window includes three main elements. First, thanking them for participating in the experiment. The last part presents a summary of the experimenter's results - how many questions she answered correctly. In the pilot there was no summary window like this, since actually it has no significance as we noted in the opening screen - we test performances on the code, and not some experimenter. But then, many experimenters came to ask about their performance. Some of the experimenters considered the result as an integral part of the process, and even more than that: "This is my result!" / "I deserve to know my result, don't I?" / "I took part in this experiment, so I deserve at least to know my score". To some of the experimenters, receiving the results was something that belongs to them. Some of them considered getting their results a feedback they deserve for their participation in the experiment. As a result, we decided to add the line that presents their score.

Adding this line was followed by an interesting phenomenon. We ran the pilot in separated teams. Adding this line started a contest between the experimenters. The experimenters compared their results with each other. In the beginning we enjoyed watching that, and considered that as a kind of success for the implementation of the gamification. However, we then realized that experimenters that got relatively bad score were offended by these comparisons. The thing is that bad score can result because some experimenter was "unlucky" and especially difficult questions were drawn for her, and not because she is less skilled than others. Either way, we decided to try to diminish the comparisons or to make them look superfluous in some way. We decided to deal with this by noting explicitly before presenting the score, that the questions are drawn and it might be that some got especially difficult or especially easy questions, so comparisons are not of any significance here.