

User Study: Programming Understanding from Similar Code

By

Anush Ramsurat Ganapathi Agraharam Sivasankar

ag3630@columbia.edu

Abstract

The aim of the user study conducted is primarily threefold:

- To accurately judge, based on a number of parameters, whether showing similar code helps in code comprehension.
- To investigate separately, a number of cases involving dynamic code, static code, the effect of options on accuracy of responses, and so on.
- To distribute the user survey, collect data, responses and feedback from the user study and draw conclusions.

First, we wanted to investigate whether showing statically similar code fragments in different languages help developers better understand code – and what consequences would follow from this.

- One of the hypotheses we'd like to test is whether showing people code in two different programming languages, say, Java and Python, will help them better understand code.
- This is under the assumption that the person is familiar with exactly one of the two shown programming languages – will it help him/her understand the code written in the other language?
- We test this by showing both statically similar looking code in the two languages (same constructs, loops) versus non-statically similar (dynamically similar) code.

Examples of such code in Java and Python:

Java:

```
private bool hasArrayTwoCandidates (int A[], int arr_size, int sum)
{
    int l, r;
    /* Sort the elements */
    quicksort(A, 0, arr_size-1);
    /* Now look for the two candidates in the sorted
       array*/
    l = 0;
    r = arr_size-1;
    while (l < r)
    {
        if(A[l] + A[r] == sum)
            return 1;
        else if(A[l] + A[r] < sum)
            l++;
        else // A[i] + A[j] > sum
            r--;
    }
    return 0;
}
```

Python:

```
def hasArrayTwoCandidates(A, arr_size,sum):
    # sort the array
    quicksort(A,0,arr_size-1)

    l = 0
    r = arr_size-1
    # traverse the array for the two elements
    while l<r:
        if (A[l] + A[r] == sum):
            return 1

        elif (A[l] + A[r] < sum):
            l += 1

        else:
            r -= 1

    return 0
```

(Code source: [geeksforgeeks.org](https://www.geeksforgeeks.org))[1] – Image 1

This idea of testing was later abandoned owing to lack of tools to procure and curate similar code in different languages. However, it remains a very interesting proposition that can be tested out.

Next, we wanted to see if showing dynamically similar code fragments in the same language help developers better understand code.

- We'd like to test whether showing people two fragments of dynamically similar code in the same language will help them identify what one fragment does.
- The argument we have is: Developers might have previously seen/been familiar with one version of implementation of an algorithm, but may not have come across

another. Hence, shown one version, they should be able to correlate and recognize the other version.

- This is tested by showing dynamically similar code fragments in the same language (Java) and observing the impact it has on the person being tested.

```
private void swap(int x, int y)
{
    int temp;

    /* Swap the elements */

    temp=x;
    x=y;
    y=temp;

    cout<<x<<y;
}
```

```
private void swap (int x, int y)
{
    //Swap the elements
    x=x+y;
    y=x-y;
    x=x-y;

    cout<<x<<y;
}
```

(Self-written code) – Image 2

The next proposition to be tested was: Will showing statically similar code fragments in the same language help developers better understand code?

- We'd like to test whether showing people statically similar code in the same programming language will help them better understand/identify code.
- Shown code with minor static adjustments in the same language, will people be able to recognize what the code does?
- We test this by showing both statically similar looking code in the two languages (same constructs, loops) versus non-statically similar (dynamically similar) code.

```
private void swap(int x, int y)
{
    int temp;

    /* Swap the elements */

    temp=x;
    x=y;
    y=temp;

    cout<<x<<y;
}
```

```
private void swap (int a, int b)
{
    //Swap the elements
    int med;
    med=a;
    a=b;
    b=med;
    cout<<a<<b;
}
```

(Self-written) – Image 3

Next, we check if showing dynamically similar code fragments in the different languages help developers better understand code. This case too is not actually tested because of the unavailability of suitable tools to procure, curate and compare similar code in different languages.

- We'd like to test whether showing people dynamically similar code in the same programming language will help them better understand/identify code.
- Shown code in two different languages that look nothing alike (different approach, variables, constructs, etc), will it help?
- We test this by showing both dynamically similar code in the two languages – C++ and Python (different constructs, loops).

An example of this:

<p>Java:</p> <pre>private int getMissingNo (int a[], int n) { int i, total; total = (n+1)*(n+2)/2; for (i = 0; i < n; i++) total -= a[i]; return total }</pre>	<p>Python:</p> <pre>def missingno(A, m): t = (n+1)*(n+2)/2 while i<n: t = t- A[i] return t</pre>
--	---

(Source: geeksforgeeks.org) [1] – Image 4

We look at what happens when we show code that is NOT similar to the code fragment that is under consideration, and the person being tested is informed.

- We'd like to test whether showing people two fragments of code that are NOT similar to each other – and informing the person being tested about this fact – has any effect on the person's ability to understand/recognize what the original code fragment does.
- We want to monitor whether showing code fragments not similar to the original code helps or hurts them. Will being informed about the fact that dissimilar code is shown have any effect?
- An interesting thought experiment with a follow-up: what if the person being tested is not informed of the fact that the code being shown alongside is dissimilar?

Java:

```
private int getMissingNo (int a[], int n)
{
    int i, total;
    total = (n+1)*(n+2)/2;

    for ( i = 0; i < n; i++)
        total -= a[i];

    return total
}
```

Python:

```
def missingno(x, y):

    temp=x
    x=y
    y=temp

    return x,y
```

Fragment 1: Java function to get missing number from 1 to n

Fragment 2: Python function to swap two numbers

(Source: geeksforgeeks.org) [1] – Image 5

Follow up experiment: What happens when we show code that is NOT similar to the code fragment that is under consideration, and the person being tested is NOT informed?

- We'd like to test whether showing people two fragments of code that are NOT similar to each other – and NOT informing the person being tested about this fact – has any effect on the person's ability to understand/recognize what the original code fragment does.
- We want to monitor whether showing code fragments not similar to the original code helps or hurts them. Will not being informed about the fact that dissimilar code is shown have any effect? Will the person assume that the code is similar and proceed to predict output accordingly?

Does the number of options presented affect accuracy of responses?

- We alter the number of options presented – with a single one of those being correct – and check if this somehow affects the accuracy of responses.
- Obviously, the greater the number of options presented, the lesser the probability of guessing the right answer.
- Also, more options implies more alternatives to consider and hence a higher chance of getting it wrong, generally, but this must be validated.

```

#include<iostream>
using namespace std;
class Test
{
    int value;
public:
    Test(int v = 0) { value = v; }
    int getValue() { return value; }
};

int main()
{
    const Test t;
    cout << t.getValue();
    return 0;
}

```

Option Set 1: (a) Compiler error (b) 0 (c) NULL (d) Seg Fault

Option Set 2: (a) Compiler error (b) 0

The correct answer is more likely to be picked in Set 2 (to be validated)

(Source: geeksforgeeks.org) [1] – Image 6

Do questions having multiple correct answers affect the outcome/response?

- We alter the options and make them multiple correct answer type questions.
- This presents a greater challenge to the candidate as the task now involves selecting not one but all correct answers.
- This should have a big effect – needs validation.

Does making the user enter textual responses via a textbox affect the response?

- Responses are collected via text box instead of using option selection.
- The answer is hence subjectively evaluated by a human being instead of objectively (predetermined).
- Subjective answers also prevent the possibility of guesswork and we can evaluate the candidate's understanding to a greater extent.
- We evaluate how this compares to questions where answers are input by option selection (objective type).

Our sources of code include:

- Simple level Google Code Jam problems
- Open-source repositories of code
- Competitive programming contest submissions that are open sourced
- Self-authored code
- Popular technical interview preparation websites

Generation of code clones is done as follows:

- Use Deckard/ other open source online tools
- Cluster code clones into categories as per their origin and nature – i.e, which code clone generator created them, whether they are statically similar or dynamically similar, etc.
- These code clusters are then used for further analysis.

Prior experience of test-takers:

- We need to take into account the prior experience of the test takers.
- A lot of our testing methodology is based on assumptions about the test taker's experience with the programming languages that we use in our user surveys – Java, C++, Python.
- Some of the code similarity detection questions may not result in totally valid results because a lot depends on their own prior knowledge and experience.

Effect of showing Minified Code

- Minification: It is the process wherein code is compressed by removing unnecessary/redundant characters from code to simplify and shorten the code length to enable faster processing/transmission over networks. [4]
- Minification often involves small changes – this may confuse the person under consideration and may only help code comprehension minimally if at all. [4]
- Minification is primarily applicable to code written in Javascript and web-development languages. [4]

Example of original HTML Code:

```
<html>
  <head>
    <style>
      #newStuff { font-family: Times New Roman }
      #newStuff { font-size: 70% }
    </style>
  </head>

  <body>
    <!-- start newStuff -->
    <div id="newStuff">
      <p>Original Code</p>
    </div>
    <!-- end newStuff -->
  </body>
</html>
```

Corresponding Minified Code:

```
<html><head><style># newStuff {font-family:Times New Roman;font-size:70%}</style></head><body><div id=" newStuff"><p>Original Code</p></div></body></html>
```

[Source: <https://www.maxcdn.com/one/visual-glossary/minification/>][2]

Types of Code Similarity – Static:

- Text-based similarity: The code fragments are identical except for minor differences in whitespace, layout, code comments, etc. It may be an exact copy-paste or with slight modifications that do not alter the logic of flow of the program.
- Token-based similarity: Changing parameter names, changing parameter names to expressions, renaming identifiers, etc.
- Syntax-based: Deletion within a line, insertion within a line, modification of a line, reordering declarations, lines and statements

- Semantics-based: Same functionality, but different implementation. Eg. Replacing while with for, etc.

[Source: http://www.cs.usask.ca/~croy/papers/2009/RCK_SCP_Clones.pdf][3]

Method of Administration – Questionnaire

- Make an online questionnaire and distribute to people via link – many questionnaire platforms such as Google Forms, etc.
- Questionnaire contains all mentioned cases and varieties of questions – including multiple choice multiple correct, text box, etc.
- There is no manual overhead in distributing questionnaires, and we can use online tools for data analysis and drawing inferences.
- Hence, effectively covers all scenarios to be tested.

Incorporating feedback about code similarity assessment

- A suggestion: Before and/or after we administer the questionnaire, we can ask a small, select group of people familiar with programming in Java and Python to rate our questions.
- We can have a scale of 1 to 5 where they can indicate whether they thought our test cases were alright, for example, whether two instances of dynamically similar code we claimed in our test were, in fact, really valid.
- They can rate our questions not only based on the similarity of code fragments in them but also on how the answer format is calibrated, quality of code, etc.
- This helps improve our user study for the next iteration, and is an effective method of spotting weaknesses and gaps in our thought process when creating the user study.
- These ratings are included as part of our data gathering and analysis, along with survey responses.
- They reflect on the quality of code we use and the effectiveness of our questionnaire.

Sample questions to ask:

- Consider the following code fragments. How similar would you say they were, on a scale of 1 to 5?
- How would you rate the correctness of the code on a scale of 5?
- How would you say the options influenced your answers?

- Do you prefer answering via textbox or by selecting from a list of options?
- Would you say it is more effective being shown code in the same language or in a different language? What about being shown static and dynamic code?

RESULTS OF THE SURVEY

Conducted preliminary survey – with 38 respondents. It involved waiting for some time to get responses

Survey was conducted using Survey Monkey

Survey consisted of questions that were intended to collect responder information and results.

This included demographical information and answers to questions on code similarity involving Java

I had a carefully controlled questions in the preliminary survey in order to keep it less time-consuming and hence have a greater chance of gathering more responses.

In addition to this, questions on Java code similarity evaluation were asked.

Questions asked in the survey have been discussed next.

Question: What is your gender?

This question is intended to survey the diversity of the respondents. There is a perceived problem, often, of diversity in tech.

I sent survey to almost as many female survey candidates as male as I wanted to make sure we had a diverse set of people answering the question.

However, more male candidates responded to the survey as compared to female. Out of 38 responses in total, 26 were male and 12 were female.

However, more male candidates responded to the survey as compared to female. Out of 38 responses in total, 26 were male and 12 were female.

The survey was sent to an equal number of male and female candidates.

The respondents were asked their age.

The rationale between including this question: Most developers tend to be young. People in tech generally transition into management beyond the age of 30, and hence I wanted to have a large representative sample of respondents from many ages: 18 to 40 (as coding proficiency varies with age and experience).

The sample of people who responded were mostly in the 20-25 range, but there were a few responses from people over the age of 30, too.

Out of 38 responses, 25 people indicated that they were in the age range 20-25.

5 people were in the range 25-30.

3 people were below 20.

5 people were above 30.

Question on Location – Country

- I asked a question to get the location (country) of the respondents
- Some of the respondents lived in a different country than the US
- Included were college students and software engineers living in India, Singapore, etc
- If the person responding lived in the US, a follow-up question on state was asked.
- 27 respondents were in the US.
- 5 respondents were in India.
- 2 were in Singapore and Australia.
- 3 were in Europe.

Question on Location – State

- Most core tech companies tend to be in California, so I wanted diverse opinions.
- I made sure to send the survey to people living and working in other states such as New York, Texas, Oregon, Illinois and Connecticut.
- This was done in order to avoid getting one-dimensional responses that may reflect only the opinions of people in a particular cultural and work environment.
- 18 respondents lived and worked in California
- 4 respondents were in NY
- 2 were in Ohio
- 1 were in Oregon

- 1 were in Connecticut
- 1 were in Chicago

Question on Industry

People were asked to indicate the industry in which their company operates.

Wanted to diversify responses away from Tech – Survey sent to people working in the big banks in NYC, at semiconductor companies, at insurance companies (AIG and the like).

People in these industries often use different coding practices than those in core technology companies in the west coast. This must be taken into account.

The industry also matters because respondents in different industries have different amounts of coding experience.

For example, tech companies require software engineers to do more coding work than banks or other businesses where the work may sometimes even be focused on just testing and documentation.

19 were in core software – Google, Microsoft, Facebook, etc.

3 were in Cloud and Storage

3 were in pure IT

9 were in banking/finance industry

2 were in Government agencies

2 were in the Service Sector

Question on Company Location

This question is intended for people who work remotely or as contactors – these people need to be included as an edge case too.

In this case, their location may be different from where their company is located.

The company location is most often in California or one of the big states, so we needed to include companies based out of states that aren't as big too like Intel (Portland, Oregon).

Question on Highest Level of Education

- Needed to include people at different age ranges and levels of education – Bachelor’s degree, Master’s degree and Doctorate.
- Most people who responded has a Bachelor’s degree. A small amount had Master’s degrees or were in the process of getting them, and the number of PhDs was very less – goes to show how employment in software engineering is more about skills obtained than the formal education level attained.
- 22 were Bachelor’s degree students/holders
- 12 were Master’s degree students or were in the process of getting one
- 4 were PhDs

Question on Major chosen in Education

- People in the software industry are mostly Computer Science majors with a solid background in coding, algorithms and data structures.
- However, there tend to be people from Electronics, Mechanical or other backgrounds who have different coding styles and perceptions based on their own education and coursework.
- Electronics students are more used to programming on microcontrollers or FPGAs, which need a different outlook and set of skills oriented toward a low-level language than a normal computer science major who has been programming in Java or high-level languages.
- 28 people majored in CS
- 9 people majored in Electronics
- 1 person majored in Economics

Question on Job Role

- Within a particular company, people can be in different job roles.
- People in different roles have different responsibilities, different perceptions of code, different coding styles, different levels of experience with code and vastly different responses to this survey.
- A software tester’s only job is to make sure the code performs well and find bugs in the code. A software engineer’s job is to produce the code. A product manager’s role is to ideate and facilitate smooth operation of the product team.

- Even within a particular category or job role, we don't know how many people are familiar to an intermediate level with Java, which is what we have used for our user study.
- Some software engineers use Java, many have now shifted to Python and other languages with higher abstraction. Others may be front-end developers, system developers, Linux/Mac OS administrators and so on.
- People in senior roles such as Engineering Managers and Software Architects transition out of day to day coding and espouse leadership and other responsibilities.

Results for this question:

- Software Engineers, Senior/Junior, Students – 26
- Product Managers – 5
- Software QA Testers – 4
- Engineering Managers - 2

Question on Number of Years of Employment as a Software Professional

- Again, to directly accommodate for different years of experience in the industry.
- This is an important question to ask as we need people who are freshers, as well as those with a lot of experience to take the survey to achieve some uniform results.
- 17 people – students/ less than one year experience
- 11 people – 2 to 4 years
- 10 people – Greater than 4 years

Question on Proficiency in Programming Languages

- Different people have experience with different programming languages.
- We need people taking the survey to be proficient in Java, because of the nature of the questions and their code being in Java.
- Hence, the survey was sent out to people who I knew were Java developers, but I wanted them to rate themselves on their knowledge of Java, among and against, other languages.

Average rating for:

- Java – 8.6/10

- C++ - 8.8/10
- Python – 8.1/10
- Scala – 6.3/10

Question on Number of Years of Programming Experience

- This is to gauge metrically how many years of experience people have in specific coding languages.
- The number of years of experience matters because people who are fresh out of school may have different coding abilities than industry veterans or those who've been in the workforce for some time.
- Also a good metric for checking if both of these demographics have the same perception of similar code.

Average Number of Years of Experience:

- C++ - Average: 7.3
- Java – 7.4
- Python – 3.1
- Scala – 1.2

Question on Use of Stack Overflow and other Platforms

- Respondents are asked about their use of crowd-sourced code platforms such as Stack Overflow.
- The rationale behind this is that forums such as these often have code similar to what people are looking for, in order to use or to refer to for ideas on what to present in their own code.
- Stackoverflow, Github, and other open-source repositories are invaluable resources for people who wish to learn to code and even for professionals who work at software companies.
- Almost all respondents indicated that they used Stackoverflow and Github as reference sources
- Some indicated that they used company internal wikis and popular textbooks on coding languages as a reference.
- Shows how these platforms, which show people similar code, are a huge aid in the code understanding and writing process.

Question on whether seeing similar code will actually help

- This question is asked as it is the premise of our survey and we want to see how people think of seeing similar code to code they want to understand, and whether or not this will help them with their comprehension.
- If so, how does it help, and does it help more for statically similar or dynamically similar code?
- 35 people responded positively
- A large fraction (>70%) indicated that seeing statically similar code was more beneficial than seeing dynamically similar code
- Dynamically similar code was preferred in cases where people could not understand the approach being used and wished to see a different approach to the same coding question

Question on involvement in competitive programming

- Participation in competitive programming and on popular websites such as Topcoder and SPOJ often helps participants identify easily and understand a piece of code, given similar code.
- In fact, the Google Code Jam, from where some of our sample Java code is taken, itself is a competitive programming contest.
- Dynamically similar as well as statically similar code both run in these contests. Contestants sometimes have radically different methods of approaching the same problem, leading to solutions of different complexity (dynamic similarity of code), and in other cases, have similar code with different constructs, loops and variables (static similarity).
- 29 participants reported having participated in at least one of these contests/ being active in at least one of these websites
- All 29 acknowledged that having competitive programming experience comes handy in identifying what other people's code does, without any documentation.

Question on prior experience in using text editors, and choice of text editor.

- Some text editors have a different way of indenting code, which often makes code look statically similar.
- Example of this: Codeblocks, Sublime Text
- Many programmers prefer to use Vim, which doesn't have this problem.
- Hence, a question on this was included too.

Calibration of the survey

Great user surveys are always easy to answer, take less time and elicit a lot of participation.

My goal in making this user survey was to make it easy to answer the questions posed in a limited amount of time, so as to not turn off respondents.

Most questions were either multiple choice, textbox or had grid-like options.

Java Questions

- Questions to gauge code similarity (static and dynamic) were posed in Java to the survey takers.
- Survey takers were adept at using and understanding Java, so this was not much of a problem

Testing for Static Similarity Identification: Static Similarity – Swapping Two Numbers – Code Fragment 1

```
class Numexchange{
    public static void main(String args[])
    {
        int p, q, intermediate;
        System.out.println("Enter p and q");
        Scanner in = new Scanner(System.in);

        p = in.nextInt();
        q = in.nextInt();

        System.out.println("The original numbers are\np = "+p+"\nq = "+q);

        intermediate = p;
        p = q;
        q= intermediate;

        System.out.println("The swapped numbers are\np = "+p+"\nq = "+q);
    }
}
```

(Code Source: <http://www.programmingsimplified.com/java/source-code/java-program-swap-numbers>)

Static Similarity – Swapping Two Numbers – Code Fragment 2 – Image 7

[Self-authored]

```

class Numexchange
{
public static void main(String args[])
{
int a, b, dummy;
System.out.println("Enter a and b");
Scanner in = new Scanner(System.in);

a = in.nextInt();
b = in.nextInt();

System.out.println("Before Swapping\na = "+a+"\nb = "+b);

dummy = a;
a = b;
b = dummy;

System.out.println("After Swapping\na = "+a+"\nb = "+b);
}
}

```

(Modified Version of previous code – statically similar)

[I wrote this code] – Image 8

- 37 out of 38 people identified what the code fragment did upon showing the 2nd code fragment.
- Most people reported the change in variable name as a help, but stated that the question was trivial by itself.

Inferences:

- Showing a statically similar piece of code does help, particularly when people who have less experience with coding are used to certain variable names and constructs being used for specific instances and in particular functions.
- If the code itself is easy to identify, then there is little effect in showing statically similar code.
- If it is a complex Google Code Jam style problem, even then it is not useful to show statically similar code.
- There exists a Goldilocks zone of difficulty where showing code has an effect

Dynamic Similarity – Swapping Two Numbers – Code Fragment 1

```
class SwapNumbers
{
    public static void main(String args[])
    {
        int p, q, intermediate;
        System.out.println("Enter p and q");
        Scanner in = new Scanner(System.in);

        p = in.nextInt();
        q = in.nextInt();

        System.out.println("Before Swapping\np = "+p+"\nq = "+q);

        intermediate = p;
        p = q;
        q= intermediate;

        System.out.println("After Swapping\np = "+p+"\nq= "+q);
    }
}
```

[Self-written – Simple code to swap] – Image 9

Dynamic Similarity – Swapping Two Numbers – Code Fragment 2

```
class SwapNumbers
{
    public static void main(String args[])
    {
        int p, q;
        System.out.println("Enter p and q");
        Scanner in = new Scanner(System.in);
        p = in.nextInt();
        q = in.nextInt();

        System.out.println("Before Swapping\np = "+p+"\nq = "+q);

        p = p + q;
        q= p - q;
        p = p - q;
        System.out.println("After Swapping\np = "+p+"\nq = "+q);
    }
}
```

[Self-written] – Image 10

These are two different algorithms. The first uses an intermediate variable called 'intermediate' to swap the two variables, whereas the second algorithm does not use any intermediate variables. These use totally different approaches.

The first swap is often familiar to many and hence I intentionally posed the second code fragment as the question, and showed the first code sample as the similar code fragment.

This had a profound effect. Surprisingly many people did not identify what the second sample did, but upon showing the first code sample, could instantly identify.

29 out of 38 people correctly identified it without any help.

Out of the remaining 8, upon showing the similar code fragment, all 8 were able to identify that it was a function to swap two values.

- In this case, an exhibition of dynamic code similarity has a huge effect on code comprehension.
- People, especially college students, are trained on particular ways of approaching problems, such as swapping two values stored in variables.
- Hence, showing them the most familiar and widely-used approach sure helped them identify what the less familiar code was doing.
- This principle can be applied to a lot of places including as a replacement to lengthy documentation – to avoid time consuming work and help programmers understand code more intuitively.

We decided to try out the dissimilar code experiment: people were shown dissimilar code and we checked if it helped or hurt.

Case: Dissimilar code fragment shown – Code Fragment 1

```

private boolean hasArrayTwoCandidates(int[] A, int arr_size, int sum)
{
    int l;
    int r;
    /* Sort the elements */
    quickSort(A, 0, arr_size-1);
    /* Now look for the two candidates in the sorted
    array*/
    l = 0;
    r = arr_size-1;
    while (l < r)
    {
        if (A[l] + A[r] == sum)
        {
            return 1;
        }
        else if (A[l] + A[r] < sum)
        {
            l++;
        }
        else // A[i] + A[j] > sum
        {
            r--;
        }
    }
    return 0;
}

```

[Source: geeksforgeeks.org] [1] – Image 11

Case: Dissimilar code fragment shown – Code Fragment 2

```

private int getMissingNo(int[] a, int n)
{
    int i;
    int total;
    total = (n + 1) * (n + 2) / 2;
    for (i = 0; i < n; i++)
    {
        total -= a[i];
    }
    return total;
}

```

[Code Source: geeksforgeeks.org][1] – Image 12

- Respondents were divided into 2 groups of 19 people each.
- When the first group of respondents were told the code fragments were dissimilar, 18 out of 19 still reported studying the dissimilar code fragments to find out what could be different in the original fragment.
- When respondents were not told they were dissimilar, all 19 reported studying the code fragment, believing it to be a useful hint to discover what the original fragment did.

Conclusions:

- This indicates that it would be useful to show dissimilar code fragments as opposed to showing nothing, because it helps people look for differences in the coding approach and deduce what the original fragment does by thinking in reverse.
- Contrary to what is believed, participants reported that it does not hurt or negatively impact them to see a dissimilar code fragment IF they are informed about it being dissimilar.

Multiple choice single correct – number of options and variation

Given this piece of Java code, respondents were asked to determine what it did.

```
• private int function(int[] arr1, int[] arr2, int m, int n)
• {
•   int i = 0;
•   int j = 0;
•   while (i < m && j < n)
•   {
•       if (arr1[i] < arr2[j])
•       {
•           i++;
•       }
•       else if (arr2[j] < arr1[i])
•       {
•           j++;
•       }
•       else // if arr1[i] == arr2[j]
•       {
•           System.out.printf(" %d ", arr2[j++]);
•           i++;
•       }
•   }
• }
```

[Code Source: [geeksforgeeks.org](https://www.geeksforgeeks.org/)][1] – Image 13

Multiple choice single/multi correct answer – variation

- Respondents needed to state what the function did – they were given 4 options: prints the intersection of two arrays, prints union of two arrays, merges two arrays and compares elements in two arrays.
- Respondents were shown dynamically similar code, and separated into 3 sets with the number of options varying for each set.
- There was not much variation in the choice between the sets. One set chose the correct answer (intersection of two arrays) more frequently than the other, despite having more number of options (thereby making it less likely).

- Respondents needed to state what the function did – they were given 4 options: prints the intersection of two arrays, prints union of two arrays, merges two arrays and compares elements in two arrays.
- Respondents were shown dynamically similar code, and separated into 3 sets with the number of options varying for each set.
- There was not much variation in the choice between the sets. One set chose the correct answer (intersection of two arrays) more frequently than the other, despite having more number of options (thereby making it less likely).

Textbox response – Variation

- Code for array reversal was shown, with dynamically similar code being shown too.
- The first method involves reversal method using block reversal technique, and the second involves block swap method. The code is a bit long.
- Both techniques produce same result.

- Textbox was given for response.
- Only 27 people correctly identified what the code was doing and wrote down an expected, precise description.
- Two persons did not submit anything (blank response).
- Nine people responded incorrectly.
- In the feedback, it was stated that having a textbox response posed a harder challenge than a multiple choice response because multiple choice could be figured out by elimination.

Minified code – Effect

- Survey takers were shown minified HTML code as part of this exercise and asked to rate its usefulness to understand the original code.
- 35 out of 38 respondents chose “Little or no use” as their response.
- Minified code is code that removes unnecessary whitespaces and other variable declarations etc. to make the code shorter on the whole and enable for easier transmission across networks.

The example of HTML Code used has been stated previously in this paper.

As expected, this provided little to no aid to the survey participants.

4 of the participants reported that this was a negative and actually hurt their chances of being able to identify what the code did, because of the lack of whitespace and indentation which guides them along the natural flow of the program and helps them visualize constructs and programming elements in the HTML code better, so as to aid understanding.

Text based static similarity

- This was of no help at all, and was rated similar to minified code in terms of usefulness.
- Respondents found that this provided neither any different logical approach nor any lever from where they could see familiar code.
- Hence, this must not be included in future surveys.

Example code for text-based static similarity – 1st sample:

```
private void reverse (int[] A, int beg, int end)
{
    int temp;
    while (beg < end)
    {
        temp = arr[beg];
        arr[beg] = arr[end];
        arr[end] = temp;
        beg++;
        end--;
    }
}
```

(Image 14)

Example code for text-based static similarity – 2nd sample:

```
private void functionrev (int[] B, int first, int last)
{
    int temp;
    while (first < last)
    {
        temp = arr[first];
        arr[first] = arr[last];
        arr[last] = temp;
        first++;
        last--;
    }
}
```

[1] – Image 15

Syntactic static similarity

- Reordering helped somewhat, as did removal of unnecessary lines and declarations.

- 12 out of 38 respondents believed syntactically similar code helped them understand given code better.
- The code they were shown could be easily rearranged.

Token based static similarity

- Token based similarity did not help a lot, according to 28 of 38 respondents.
- This is because mere changes in variable/parameter names and renaming identifiers does not help in understanding the original code or offer any extra degree of familiarity to the survey taker.
- There is little in the way of non-dynamic differences that actually contribute to a different perspective on the part of the user, in most of these cases.

Semantics based static similarity

- This involves code having the same functionality, but different implementations.
- This often applies to code with loops. For loops are replaced with while loops, and so on.
- 26 respondents indicated they prefer static code that is semantically similar, as opposed to other forms of statically similar code offered to aid comprehension.

Feedback – Rating and Comments on Code Quality

- Some developers mentioned that the code was restricted to Java and inter-language code similarity comparisons must be made to establish our results on those, but I explained that the tools we have don't allow us to do this.
- Code quality was fine according to responders but some mentioned that we needed to ask more than one question on the same category to eliminate false positive results.
- The code quality of the survey was rated as 8.1 on 10 on average by the 38 survey responders.

Feedback – Rating and Comments on Answer Option Formats

- Answer option formats were rated 8 on 10 by test takers.
- Multiple choice questions were preferred by 30 out of 38 as opposed to textbox response or other formats of response, presumably because of the option to eliminate options to get to the correct answer.
- The number of answer options was also mentioned – people asked for lesser number of options.

REVISED TECHNICAL INPUT TO THE SURVEY

The code samples that we wrote and used are all deemed to be similar in nature based on our manual assessment of them. We also used certain tools to compare and analyse code samples and determine them to be statically or dynamically similar in nature. As always, brevity is key in an survey as survey respondents do not wish to spend more time answering questions, so we restricted the number of code samples used to 6 for dynamic (3 pairs) and 4 for static (2 pairs). The dynamic code samples were compared using the tools in the lab and the static code samples were generated using the tool SourcererCC.

The respondents to the survey were shown each of these 3 pairs of dynamically similar code. The code samples collected above were run and tested in the lab and were determined to be dynamically similar as per the tools in the lab.

The respondents were asked to identify what each of the code fragments did when executed, and show the corresponding pair element code to see if it helped them identify the original code fragment.

Here are the dynamic code samples used.

Sample 1: Google Code Jam Problem "Counting Sheep"

Problem Statement :

<https://code.google.com/codejam/contest/6254486/dashboard#s=p0&a=0>

- Since Google Code Jam solutions are open sourced and free for reuse, I took two working solutions to this problem from the solutions submitted and posted on the solutions collection, and administered them to the survey respondents stating they were dynamically similar and produced the same output.
- These 2 solutions were authored by 'Upgrade' and 'OMGTallMonster' respectively.
- <https://www.go-hero.net/jam/16/name/upgrade>
- <https://www.go-hero.net/jam/16/name/OMGTallMonster>

Respondents were asked to identify what the code did (i.e., a correct response would come close to what the problem statement indicates).

- Only 4 out of 27 respondents correctly identified what the code did.
- The high error rate was anticipated given the complexity of Google Code Jam problems, but even then the surprisingly low number of 4 shows that survey questions should be of lesser complexity as most people are not exposed to competitive programming and hence would not find it apt to recognize what a competitive programming solution does.
- Due to the highly complex nature of the code and multiple functions distributed throughout the program, it is quite difficult to find out the nature of the program and what it does as opposed to the other questions on the survey which are single-function and easy to understand.

```
Pair2:
//Method 1
public static boolean isSpaceChar(int c) {
    return c == ' ' || c == '\n' || c == '\r' || c == '\t' || c
== -1;
}

//Method 2
public boolean isSpaceChar(int c) {
    if (filter != null)
        return filter.isSpaceChar(c);
    return isWhitespace(c);
}

public static boolean isWhitespace(int c) {
    return c == ' ' || c == '\n' || c == '\r' || c == '\t' || c ==
-1;
}
```

[Code Sample provided by Fang-Hsiang Su]

- This is a dynamically similar code samples verified for similarity in the lab. It checks if there is any sort of space/tab/whitespace in the text.
- 25 out of 27 respondents got the output correct. Most respondents suggested that showing the similar code did not help as the code was easy to identify and the second code, although dynamically similar in nature, is largely unhelpful.

```
Pair3:
//Method 1
public long getSum(long[] n, int L, int R) {
    long sum = 0;
    if (R >= 0) {
```

```

        sum = n[R];
    }
    if (L > 0) {
        sum -= n[L - 1];
    }
    return sum;
}

//Method 2
public static long sum(int a, int b) {
    if(a > b)
    {
        return 0;
    }

    return array[b + 1] - array[a];
}

```

- This code fragment is a bit tricky as it doesn't return completely same output but is still dynamically similar.
- 12 out of 27 respondents correctly identified what the code sample does when shown the additional code sample. These samples were also compared using the tools in the lab.

For the static code samples, we ran the code samples on a research software package called 'SourcerCC'.

<https://github.com/Mondego/SourcererCC>

- This package is a three step software process to determine code clones: tokenization, indexing and clone detection.
- After configuring and executing commands to run each of these stages, we can detect code clones for input code snippets – I used 2 pairs of statically similar code samples and determined their similarity quotient using the SourcererCC package.

- These code samples were put in groups with different ParentIds to distinguish them and enable for, and they passed the threshold for similarity set within the package and hence were deemed statically similar.

Here are the static code fragments used:

```
//Pair 1:

//Method 1
public static boolean space(int z) {
return z == ' ' || z == '\n' || z == '\r' || z == '\t' || z == -1;
}

//Method 2

public static boolean spacedOut(int d) {
return d == ' ' || d == '\r' || d == '\n' || d == '\t' || d == -1;
}
```

- Here, the variable names and ordering in the return statement have been changed in the two code fragments.
- They have been deemed statically similar by SourcererCC when I fed them into it.
- They are code samples to check if there are any whitespaces/tabs.
- All survey respondents responded with the correct output and 9 survey respondents indicated that showing the similar code helped them understand and clarify their doubts over what the original code was doing.

```
//Pair 2:

//Method 1

static void function1(int a[], int beg, int end)
{
    int temp;
    if (beg >= end)
        return;
    temp = a[beg];
    a[beg] = arr[end];
    a[end] = temp;
    reverseArray(a, beg+1, end-1);
}

//Method 2
```

```
static void function2(int x[], int first, int last)
{
    int inter;
    if (last<=first)
        return;

    inter = x[first];
    x[first] = x[end];
    x[end] = inter;
    revfn(x, first+1, last-1);
}
```

[Source: www.geeksforgeeks.org]

- These are functions to reverse an array.
- The static changes made are in the variable names and the function name.
- SourcererCC identified these code fragments as being similar and above the threshold value needed for them to be deemed statically similar.
- 21 people correctly identified the code as array reversal.
- 16 out of these people indicated that the statically similar code that was shown helped them make their judgement.

References:

- [1] www.geeksforgeeks.org/data-structures
- [2] <https://www.maxcdn.com/one/visual-glossary/minification/>
- [3] http://www.cs.usask.ca/~croy/papers/2009/RCK_SCP_Clones.pdf
- [4] [https://en.wikipedia.org/wiki/Minification_\(programming\)](https://en.wikipedia.org/wiki/Minification_(programming))

Other references have been mentioned through the paper as and when they were used.

Appendices:

Survey material – Certain answers are included from the first part of the survey. The rest of the responses were condensed and included as part of the paper itself as they were mostly yes/no responses or technical question responses that were either correct or wrong.

Demographics

Gender: Male; Age: 19; Country: US; State: California; Industry: Software; Education: Bachelor's; Major: CS; Job Role: Student;

Gender: Male; Age: 19; Country: US; State: California; Industry: Software; Education: Bachelor's; Major: CS; Job Role: Student;

Gender: Male; Age: 19; Country: India; Industry: Software; Education: Bachelor's; Major: Electronics; Job Role: Student;

Gender: Male; Age: 31; Country: US; State: California; Industry: Storage; Education: Bachelor's; Major: Electronics; Job Role: Software Engineer;

Gender: Male; Age: 31; Country: India; Industry: Software; Education: Bachelor's; Major: Electronics; Job Role: Software Engineer;

Gender: Male; Age: 33; Country: US; State: California; Industry: Software; Education: PhD; Major: Electronics; Job Role: Engineering Manager;

Gender: Male; Age: 32; Country: US; State: California; Industry: Software; Education: Bachelor's; Major: CS; Job Role: Product Manager;

Gender: Male; Age: 35; Country: US; State: California; Industry: Storage; Education: Master's; Major: Electronics; Job Role: Product Manager;

Gender: Male; Age: 26; Country: US; State: California; Industry: Software; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 26; Country: US; State: Illinois; Industry: Banking; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 28; Country: US; State: California; Industry: Software; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 29; Country: US; State: Ohio; Industry: Software; Education: PhD; Major: CS; Job Role: Engineering Manager;

Gender: Male; Age: 29; Country: US; State: Ohio; Industry: IT; Education: Bachelor's; Major: Electronics; Job Role: Software Engineer;

Gender: Male; Age: 22; Country: US; State: California; Industry: Software; Education: Master's; Major: CS; Job Role: Product Manager;

Gender: Male; Age: 21; Country: US; State: Connecticut; Industry: Banking; Education: Bachelor's; Major: Electronics; Job Role: Software Engineer;

Gender: Male; Age: 22; Country: US; State: California; Industry: Storage; Education: Bachelor's; Major: CS; Job Role: Product Manager;

Gender: Male; Age: 22; Country: US; State: California; Industry: IT; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 22; Country: US; State: California; Industry: IT; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 25; Country: US; State: NY; Industry: Banking; Education: PhD; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 23; Country: US; State: NY; Industry: Banking; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 23; Country: US; State: NY; Industry: Banking; Education: Master's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 24; Country: US; State: California; Industry: Govt; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 21; Country: US; State: California; Industry: Govt; Education: Bachelor's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 22; Country: US; State: California; Industry: Software; Education: Master's; Major: CS; Job Role: Software Engineer;

Gender: Male; Age: 23; Country: US; State: California; Industry: Service; Education: Bachelor's; Major: CS; Job Role: QA Tester;

Gender: Male; Age: 25; Country: US; State: Oregon; Industry: Service; Education: PhD; Major: Economics; Job Role: Software Engineer;

Gender: Female; Age: 21; Country: Netherlands; Industry: Banking; Education: Bachelor's; Major: CS

Gender: Female; Age: 22; Country: UK; Industry: Software; Education: Bachelor's; Major: CS

Gender: Female; Age: 22; Country: Belgium; Industry: Banking; Education: Master's; Major: CS

Gender: Female; Age: 21; Country: Australia; Industry: Software; Education: Bachelor's; Major: Electronics; Job Role: QA Tester

Gender: Female; Age: 22; Country: Singapore; Industry: Banking; Education: Master's; Major: Electronics

Gender: Female; Age: 24; Country: India; Industry: Software; Education: Master's; Major: CS

Gender: Female; Age: 21; Country: India; Industry: Software; Education: Master's; Major: CS

Gender: Female; Age: 21; Country: India; Industry: Software; Education: Master's; Major: CS

Gender: Female; Age: 22; Country: US; State: NY; Industry: Banking; Education: Master's; Major: CS

Gender: Female; Age: 21; Country: US; State: California; Industry: Software; Education: Bachelor's; Major: CS

Gender: Female; Age: 23; Country: US; State: California; Industry: Software; Education: Master's;
Major: CS; Job Role: Product Manager

Gender: Female; Age: 23; Country: US; State: California; Industry: Software; Education: Master's;
Major: CS; Job Role: QA Tester

Gender: Female; Age: 22; Country: US; State: California; Industry: Software; Education: Master's;
Major: CS; Job Role: QA Tester

Some sample responses to questions on survey improvement feedback

“The survey had good overall demographic questions and technical questions and I think it is a good way to judge how comfortable people are with seeing similar code and doing some experiments on that. However, there should be more questions in the survey on the technical part to fully judge this.” – Columbia University Master’s Student

“Nice work with the survey. I am not too sure about the project you are working on but the survey seems pretty long and comprehensive. For feedback, I would say that you should have a uniform way of giving it to everyone in a single form rather than separating it into demographic and technical surveys” – Software Engineer at Bridgewater Associates

“You should not keep the options for some of them as multiple choice and some as textbox. Try to keep all the answer options in one style.” – Columbia University Master’s Student

“You should ask more than just a single question on some of the categories because there may be some false positives in judging what the code does. Also, code quality was pretty good and simple to understand. Nice job.” – Software Engineer at Google

“Hey nice survey Anush but you should not ask questions with too many options. Reduce the number of options in the answers. Also, just Java questions? I’m a C++ person so in my opinion you should include some C++ questions too for people like me. Thanks.” – Columbia University Master’s Student

“What is the survey for actually? The questions were all based on Java and I code in C++. Can you please add some C/C++ questions and I’ll retake it?” – Software Engineer at Microsoft

“Code quality in the survey is good” – Software Engineer at Intel

“Nice job on the survey – just curious about what kind of research you are working on – I’m assuming this is to test if people can understand code better after being shown similar code.” – Technology Analyst at JP Morgan Chase

“Good work man but I’m not sure why you’re asking so many questions about my background considering this is a technical survey. “ – Software Engineer at Visa

Ratings

Subject	Java	C++	Python	Scala
1	8.6	8	8	6
2	8.8	8	8	5
3	8.4	8	8	5
4	8.6	8	8	6
5	8.7	8	8.6	7
6	8.5	8	8.5	9
7	8.9	8	8.1	8
8	8.3	8.5	7	6
9	8.3	8.5	7	5
10	8.9	8.6	8.5	6
11	8.6	8.7	9	8
12	7.8	9.5	8.3	8
13	7.8	9.5	7.5	7
14	8.8	9.5	8.9	5
15	8.8	9.4	8.1	6.5
16	8.7	9.2	8.3	6
17	9	8.8	7.9	6
18	9.1	8.6	7.8	7
19	8.8	8.8	8.4	5.5
20	8.6	9.1	8.5	6.3
21	8.7	9.2	7.7	6.5
22	8.7	9.2	8	6
23	8.4	9.3	8	5.5
24	8.6	8.8	8.2	8
25	8.5	9	8.1	6
26	8.5	9	8.2	6.1
27	9	8.9	9	6
28	8.4	8.8	7.2	6.5
29	8.6	8.8	7	6
30	8.6	9.1	9.2	5
31	8.1	8.5	8.5	7
32	9	8.6	8.6	7
33	8.7	8.9	7.6	5
34	7.5	9	7.8	6
35	9.5	9.4	8	6
36	8.7	8.5	8.2	6.5
37	8.5	8.8	8	7
38	8.7	9.2	8.1	5

Years of Experience

Subject	Java	C++	Python	Scala
1	6	7	2	1
2	8	8	3	0
3	6	5	3	0
4	6	5	4	0
5	3	4	3	0
6	8	7	4	3
7	5	8	2	1
8	8	6	4	1
9	7	7	2	1
10	5	6	4	0
11	9	8	5	1
12	9	7	2	0
13	8	8	2	0
14	10	7	0	0
15	8	7	3	2
16	3	5	2	2
17	6	7	2	1
18	7	7	4	2
19	8	6	4	1
20	6	8	3	4
21	7	9	3	2
22	8	8	5	1
23	7	7	1	0
24	10	7	2	1
25	10	10	3	2
26	10	8	3	2
27	7	9	5	0
28	5	8	7	2
29	8	6	4	1
30	9	8	4	1
31	5	6	2	1
32	7	7	2	1
33	9	9	0	3
34	10	9	3	2
35	7	8	3	2
36	10	9	3	2
37	7	8	5	2
38	9	9	5	1