Intro Block

Study on the Readability of Code Features and Spatial Ability

Thank you for taking our study! Broadly, this study investigates the readability of program functions. We are looking at how the readability of these features interacts with spatial reasoning ability.

Time: 30-35 minutes. As every section is timed, it will **not take longer than 35 minutes**.

Prerequisites: Minimal C++ Programming Experience - EECS 183 or ENGR 101 are enough. **It is ok if you don't recognize all of the code features or can't figure out some of the programs. In those cases, please just answer with your best guess.** Also, please avoid using scratch paper if possible.

Study Details: This study has three parts. They may be in any order:
1. Spatial Reasoning Assessment (<=6 Minutes): Consists of 20 spatial reasoning questions
2. Programming Portion (<=18 Minutes): Consists of six different functions where we ask you for the program's output or return value.
3. Demographics Questionnaire (~5 Minutes)

Risks and Benefits: There are no anticipated risks in this study. Other than the potential for compensation, there are no direct benefits to you for participating in this research study; however, this study may help us understand how code features improve student programmer education and productivity!

Compensation: The rewards of this study are contingent upon the successful completion of the entire study. We do not consider responses that are random or largely left blank to be a successful completion.

Confidentiality: Beyond details necessary for compensation, we will not collect or save any identifying information, and you may stop at any time. Data obtained in this study will be used to compare the effectiveness of different code features only.

Contact Information: If you have any questions about this study, you may contact the investigators: Madeline Endres (endremad@umich.edu) and Amir Kamil (akamil@umich.edu)

By clicking "Next", you certify that you consent to participate in this study. You may retract consent at any time before completing the survey by closing this browser window.

Note: This study has been ruled IRB exempt by the University of Michigan IRB under case number HUM00167300.

Programming Portion Intro

Programming Section: <= 18 Minutes

In this next section, we will show you a series of **6 C++ functions**. For each, we will give you function inputs, and we will ask you to **type either the function's return value or output**, depending on the question.

Each problem has a time limit of 3 minutes. Some problems may be harder than others or may contain programming elements you are not familiar with. Even so, please **answer each problem to the best of your ability**, even if your answer is incomplete or you are unsure of the answer or some of the program's features. If you have finished the question before the 3 minutes are up, feel free to go on to the next question. Once you have moved on from a question, you will not be able to return to the previous one.

Digits Recursive

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(1314)` in the box below:

```
int func(int value) {
    if (value == 0) {
```

```
        return 0;
    } else {
        return (value % 10) + func(value / 10);
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

Digits Tail

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(1314)` in the box below:

```
int func(int value) {
    return func_helper(value, 0);
}

int func_helper(int value, int result) {
    if (value == 0) {
        return result;
    } else {
        return func_helper(value / 10, result + value % 10);
```

```
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes	No
○	○

Digits Iterative

These page timer metrics will not be displayed to the recipient.

First Click: *0 seconds*

Last Click: *0 seconds*

Page Submit: *0 seconds*

Click Count: *0 clicks*

0300

Please write the **return value** of `func(1314)` in the box below:

```
int func(int value) {
    int result = 0;
    while (value > 0) {
        result += value % 10;
        value = value / 10;
    }
    return result;
}
```

Does this function contain elements you are unfamiliar with?

Yes	No
○	○

Exp Recursive

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0300

Please write the **return value** of `func(2, 4)` in the box below:

```
int func(int x, int y) {
    if (y == 0) {
        return 1;
    } else if (y == 1) {
        return x;
    }

    int tmp = func(x, y / 2);
    tmp = tmp * tmp;
    if (y % 2 == 1) {
        return x * tmp;
    } else {
        return tmp;
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes	No
O	O

fiblike Recursive

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(3)` in the box below:

```
int func(int n) {
    if (n <= 1) {
        return n;
    } else {
        return func(n - 1) - 2 * func(n - 2);
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes	No
O	O

fiblike Tail

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*

Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(3)` in the box below:

```
int func(int n) {
    if (n <= 1) {
        return n;
    } else {
        return func_helper(n, 1, 0, 1);
    }
}

int func_helper(int n, int i, int a, int b) {
    if (i == n) {
        return b;
    } else {
        return func_helper(n, i + 1, b, b - 2 * a);
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

fiblike Iterative

These page timer metrics will not be displayed to the recipient.

First Click: *0 seconds*

Last Click: *0 seconds*

Page Submit: *0 seconds*

Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(3)` in the box below:

```
int func(int n) {
    if (n <= 1) {
        return n;
    }
    int a = 0;
    int b = 1;
    for (int i = 1; i < n; ++i) {
        int c = b - 2 * a;
        a = b;
        b = c;
    }
    return b;
}
```

Does this function contain elements you are unfamiliar with?

Yes	No
○	○

gcd Recursive

These page timer metrics will not be displayed to the recipient.

First Click: *0 seconds*

Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(100, 70)` in the box below:

```
int func(int x, int y) {
    if (y == 0) {
        return x;
    } else {
        return func(y, x % y);
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

gcd Iterative

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func(100, 70)` in the box below:

```
int func(int x, int y) {
    while (y != 0) {
        int tmp = x % y;
        x = y;
        y = tmp;
    }
    return x;
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

palindrome Recursive

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func({5, 1, 3, 2, 9, 3, 1, 5}, 8)` in the box below:

```
bool func(int array[], int size) {
    if (size <= 1) {
        return true;
    } else  if (array[0] != array[size - 1]) {
        return false;
    } else {
```

```
        return func(array + 1, size - 2);
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes	No
O	O

palindrome Iterative

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **return value** of `func({5, 1, 3, 2, 9, 3, 1, 5}, 8)` in the box below:

```
bool func(int array[], int size) {
    while (size > 1) {
        if (array[0] != array[size - 1]) {
            return false;
        }
        ++array;
        size -= 2;
    }
    return true;
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

reverse Recursive

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please **write the contents of `array`** after execution of `func({4, 1, 2, 3, 7, 6, 9}, 7)` in the box below:

For example, if `array = {1, 2, 3}` write 1 2 3

```
void func(int array[], int size) {
    if (size > 1) {
        int tmp = array[0];
        array[0] = array[size - 1];
        array[size - 1] = tmp;
        func(array + 1, size - 2);
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

reverse Iterative

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please **write the contents of `array`** after execution of `func({4, 1, 2, 3, 7, 6, 9}, 7)` in the box below:

For example, if `array = [1, 2, 3]` write 1 2 3

```
void func(int array[], int size) {
    while (size > 1) {
        int tmp = array[0];
        array[0] = array[size - 1];
        array[size - 1] = tmp;
        ++array;
        size -= 2;
    }
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

triangle Recursive

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

Please write the **print output** of `func(4)` in the box below:

Use the enter key to separate each line of the output.

```
void func(int value) {
    if (value == 0) {
        return;
    }
    for (int i = 0; i < value; i++) {
        cout << '*';
    }
    cout << endl;
    func(value - 1);
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

triangle Iterative

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*

Page Submit: *0 seconds*
Click Count: *0 clicks*

0300

Please write the **print output** of `func(4)` in the box below:

Use the enter key to separate each line of the output.

```
void func(int value) {
    for (int i = value; i > 0; i--) {
        for (int j = 0; j < i; j++) {
            cout << '*';
        }
        cout << endl;
    }
    return;
}
```

Does this function contain elements you are unfamiliar with?

Yes ○ No ○

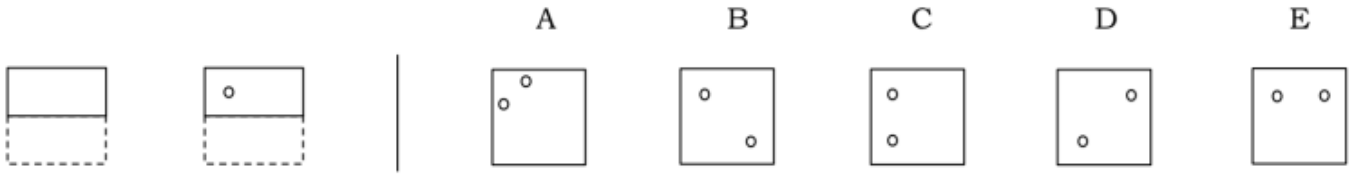
Spatial Reasoning Test

Paper Folding Portion: <= 6 Minutes

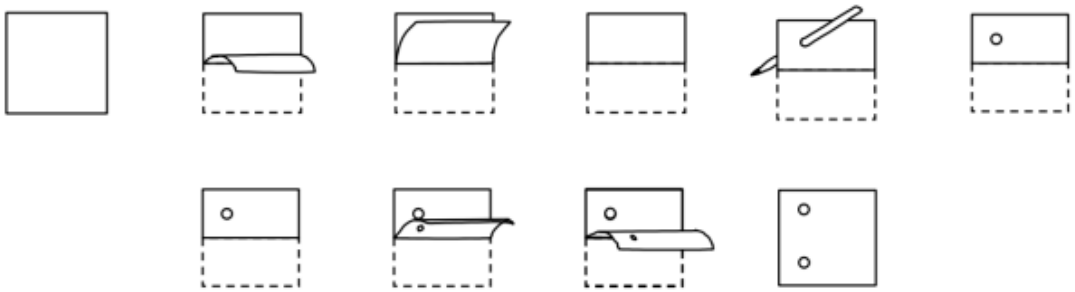
In this test you are to **imagine the folding and unfolding of pieces of paper**. For each problem, there are some figures drawn to the left of a vertical line and there are others drawn to the right of the line. The figures at the left represent a square piece of paper being folded, and the last of these figures has one or two small circles drawn on it to show where the paper has been punched. Each hole is punched through all the thicknesses of paper at that

point. One of the five figures on the right of the vertical line shows where the holes will be when the paper is completely unfolded. You are to decide which one of these figures is correct and select the corresponding button (choices A-E).

Now try the sample problem below. (In this problem only one hole was punched in the folded paper).



The **correct answer to the sample problem above is C**. The figures below show how the paper was folded and why C is the correct answer.



In these problems, all of the folds that are made are shown in the figures at the left of the line, and the paper is not turned or moved in any way except to make the folds shown in the figures. Remember, the answer is the figure that shows the positions of the holes when the paper is completely unfolded.

Some of the problems on this sheet are more difficult than others. **If you are unable to do one of the problems, simply skip over it and go on to the next one.**

You will have three minutes for each of the two parts of this test. Each part has ten questions.

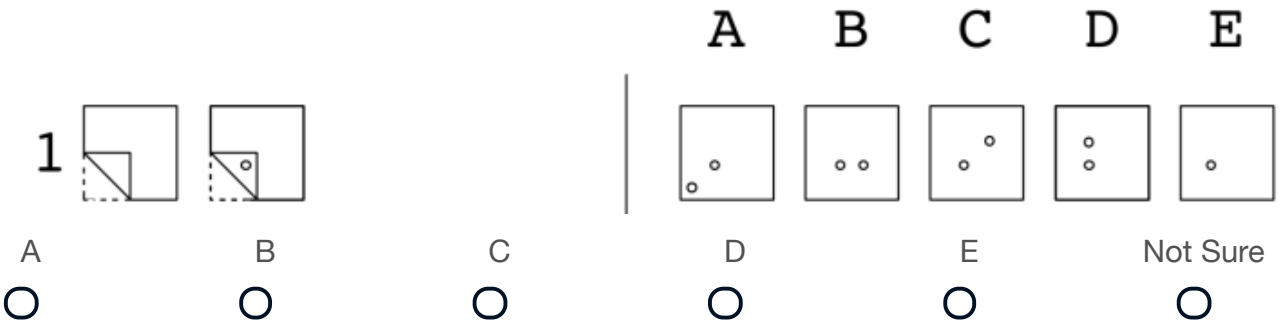
Please do not use scratch paper for this section.
When you are ready to start, press the button below.

These page timer metrics will not be displayed to the recipient.

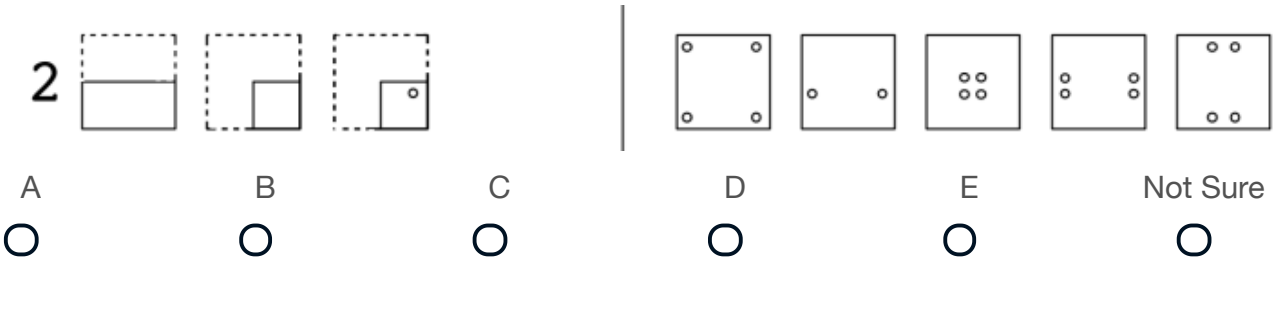
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0300

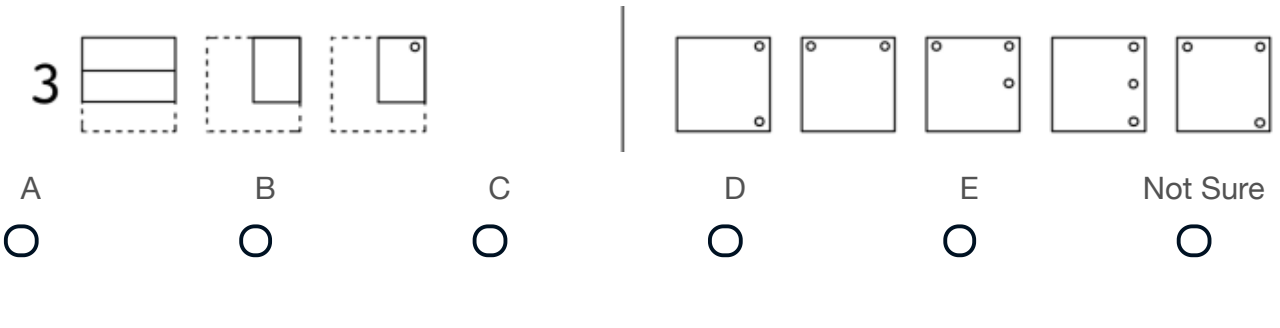
For each of the following, choose the paper on the right that results from the folding operations on the left:



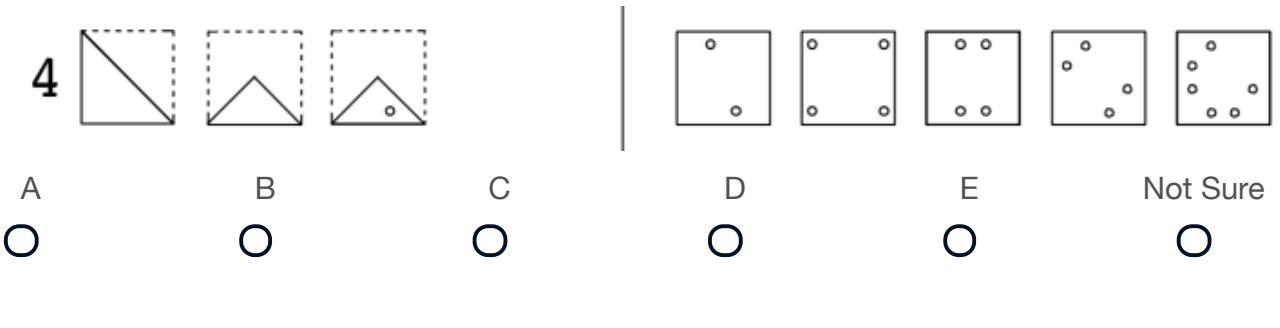
A B C D E Not Sure



A B C D E Not Sure

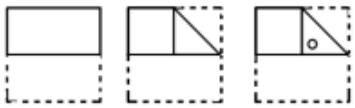


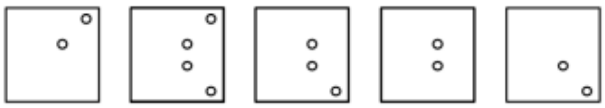
A B C D E Not Sure



A B C D E Not Sure

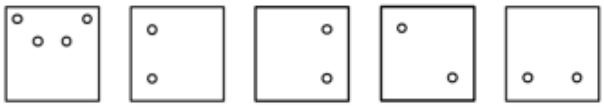
5





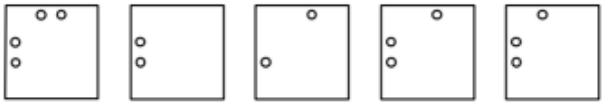
A B C D E Not Sure

6



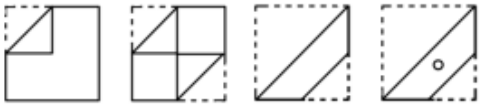
A B C D E Not Sure

7



A B C D E Not Sure

8

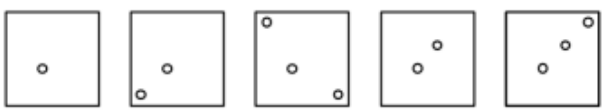


A B C D E Not Sure

9

A B C D E Not Sure

10



A B C D E Not Sure

Thank you for completing the first half of the paper folding questions!

Please click the button below to go to the second half.

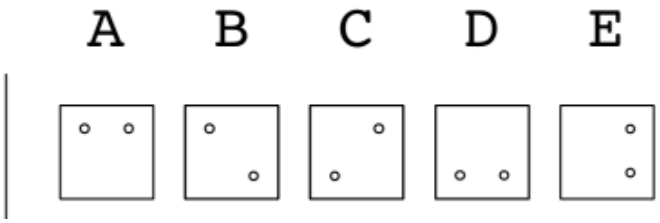
The second half will also contain ten questions and take 3 minutes.

These page timer metrics will not be displayed to the recipient.
First Click: *0 seconds*
Last Click: *0 seconds*
Page Submit: *0 seconds*
Click Count: *0 clicks*

0 3 0 0

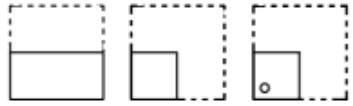
For each of the following, choose the paper on the right that results from the folding operations on the left:

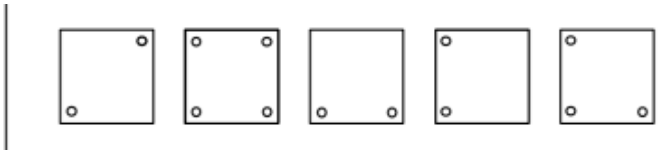
11



A B C D E Not Sure

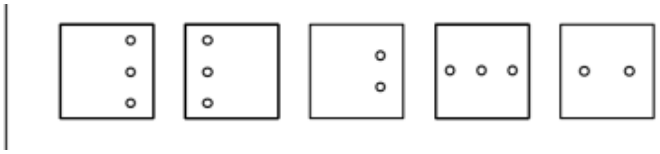
12





A B C D E Not Sure

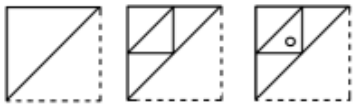
13



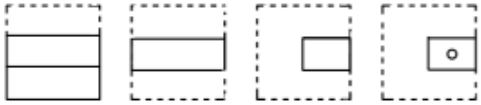
A B C D E Not Sure

14 

A B C D E Not Sure

15 

A B C D E Not Sure

16 

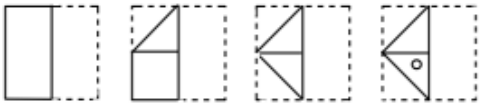
A B C D E Not Sure

17 

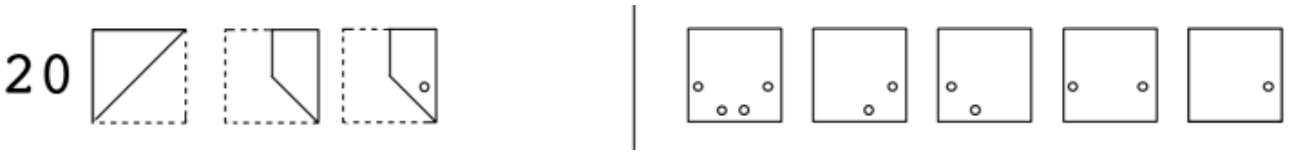
A B C D E Not Sure

18 

A B C D E Not Sure

19 

A B C D E Not Sure



A B C D E Not Sure

Demographics Results

Demographics (Part 1/2)

Thank you again for taking the study! Next, we will ask for some general demographics information. All information from this section will be kept confidential and not stored with any identifying information. This is the last section before the end of the study.

Which gender do you most identify with?

I identify my ethnicity as (select all that apply):

- [ ] Asian
- [ ] Black/African
- [ ] Caucasian
- [ ] Hispanic/Latinx
- [ ] Native American
- [ ] Pacific Islander
- [ ] Other
- [ ] Prefer not to answer

How many years of programming experience do you have?

Please select which EECS courses you are currently enrolled in or have completed (select all that apply):

- [ ] ENGIN 101
- [ ] EECS 183
- [ ] EECS 280
- [ ] EECS 281
- [ ] EECS 370
- [ ] One upper level EECS elective
- [ ] Two or more upper level EECS electives

Did you use scratch paper while completing the programming portion?

- ○ Yes
- ○ No

Socioeconomic Block

Demographics (Part 2/2)

The following are questions related to socioeconomic status. As with before, all information will be kept confidential and not stored with any identifying information.

Does your family own a car, van, or truck?

- ○ No
- ○ Yes, one
- ○ Two or more

How many times did you or your family travel away on holiday during the past 12 months?

- ○ Never
- ○ Once
- ○ Twice
- ○ Three or more times

Growing up, did you have a bedroom to yourself?

- Yes
- No

How many computers (including laptops and tablets, not including game consoles and smartphones) does your family own?

- None
- One
- Two
- More than two

How well off do you think your family is?

- Extremely well
- Very well
- Moderately well
- Slightly well
- Not well at all

What is your family's zipcode?

Final Block

Survey Complete

Thank you for completing our study on programming style features!

To be considered for compensation, please enter either your uniquename or the access code from the email we sent you in the box below. All compensation notifications will be sent by November 30th.