



Instructions and consent form

Welcome to the CS Survey!

- Many questions just ask for your opinion - those questions do not have right or wrong answers.
- Please complete the survey on your own.
- Please do not use outside resources as you complete the survey.
- Please do not discuss the survey with classmates until they complete it.
- You cannot change your answers after you move on to the next page.

Thank you for doing the survey!

You may read the consent form below or at this link:

https://docs.google.com/document/d/1LGNcQgm8zsb1YHxZemOsaag3T_RsCKvUmVt1IXpb3k4/edit

Consent Cover Letter Readability and Intelligibility of Code Examples

The purpose of this research study is to investigate how computer science students think about the style and readability of code. We are doing this study because we want to contribute to the development of instruction for teaching good coding style.

It will take you approximately 1 hour to complete this study. You will be asked questions about code samples. You will also be asked demographic questions (e.g., age, gender) and questions about your experience with computer science (e.g., previous courses taken).

We cannot promise any direct benefit for taking part in this study. However, possible benefits include practice thinking through the style and readability of code. We hope the information we get from this study may help develop a greater understanding of how students learn programming in the future.

Your instructor may award credit or extra credit for completing this survey. Your answers to survey questions will be shared with your instructor, along with your uID, so that credit can be awarded. However, **your instructor will not know if you consented to participate in the research.**

After the data above has been shared with your instructor, I will delete all of your data if you chose not to participate in the research. If you consented to participate in the research, I will keep your data confidential by replacing your uID with an anonymous code before analyzing the data. Only de-identified data using this code will be shared.

If you have any questions complaints or if you feel you have been harmed by this research please contact Dr. Eliane Wiese (School of Computing, University of Utah) at eliane.wiese@utah.edu or at 617-448-4948.

Contact the Institutional Review Board (IRB) if you have questions regarding your rights as a research participant. Also, contact the IRB if you have questions, complaints or concerns which you do not feel you can discuss with the investigator. The University of Utah IRB may be reached by phone at (801) 581-3655 or by e-mail at irb@hsc.utah.edu.

It should take 1 hour to complete the questionnaire. Participation in this study is voluntary. You can choose not to take part. You can choose not to finish the questionnaire or omit any non-CS (computer science) question you prefer not to answer without penalty or loss of benefits.

To give your consent to have your results included in the research project, click "I agree". If you do not want to participate in the research, click "I do not agree." This choice has no impact on you receiving extra credit. Thank you for participating in this study. I appreciate your time.

FOOTER FOR IRB USE ONLY
Version: 112011

«Institution»	«IRB»
«Image:Stamp»	«Approved» «ApprovedDate»
	«Expiration» «ExpirationDate»
	«Number»

Do you agree to consent to the research?

- ☐ I agree
- ☐ I do not agree

Are you 18 or older? (only data from participants who are 18 or older will be included in the research.)

- ☐ Yes, I am 18 or older.
- ☐ No, I am 17 or younger.

Please enter your uID (your instructor needs your uID to award you extra credit - otherwise we won't know it's you).

Writing directions

The next few questions will ask you to write code. Do not use the TAB key in the text boxes. Use spaces instead.

If you need to make the text boxes longer or wider, click and drag the lower right corner.

Writing: input is 7

Write a function that takes an int as input and returns a boolean. The first and last line are provided for you.

- When the input is 7, return true.
- Otherwise, return false.

```
public boolean isSeven(int num) {
```

```
}
```

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

Writing: Fish-ishness

Write a function that takes a String as input and returns a String. The first and last lines are provided for you.

- For input Strings that end in "sh", concatenate the ending "-ishness" and return a message saying how long the new word is.
- For all other input Strings, return a message saying how long the given word is.
- Write the function so that it would **easy for someone else to modify** (e.g., if someone else wanted to look for Strings that end with different letters, or concatenate a different ending)

HINTS:

- `word.endsWith("sh")` will return true if the `String` `word` ends with "sh".
- The messages should be **returned**, not printed.
- You may use `+` to concatenate strings. For example:

```
String word2 = word1 + " there";
```


If `word1` is "hello", `word2` is "hello there".

The messages should follow the format of the examples:

Input	Output
Fish	Your word was Fish. The length of Fish-ishness is 12.
Hat	Your word was Hat. The length of Hat is 3.

```
public String ishness(String word) {
```

```
}
```

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

Writing: reverse array

Write a function that takes an arrays of ints as input and returns an array of ints. The first and last line are provided for you. **Read the hints.**

- The returned array should have the same content as the input array, but in reverse order.
- Do not change the original input array (that is, this function should be **non-destructive**).

Hints:

- Be careful about out-of-bounds exceptions. Remember that the last item in an array is at position `array.length - 1`.
- Initialize a new array with `int[] array = new int[size];`
- Find the length of an array with `array.length` (no parentheses).

Examples:

Input	Output
[1,2,3]	[3,2,1]
[1]	[1]
[]	[]

```
public int[] reverseArray(int[] array) {
```

```
}
```

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

Writing: special number array

Write a function that takes an arrays of ints and an int as input and returns a boolean. The first and last line are provided for you. **Read the hints.**

- Return true if: there are two numbers in the array that multiply to the inputted int OR if an element in the array is the square root of the inputted int.
- Return false otherwise.
- Do not change the original input array (that is, this function should be **non-destructive**).

Hints:

- Be careful about out-of-bounds exceptions. Remember that the last item in an array is at position `array.length - 1`.
- Initialize a new array with `int[] array = new int[size];`
- Find the length of an array with `array.length` (no parentheses).

Examples:

Input	Output
<code>[]</code> , 16	false
<code>[4, 5, 1]</code> , 16	true
<code>[2, 87, 8]</code> , 16	true
<code>[1, 5, 91, 3]</code> , 16	false
<code>[2]</code> , 16	false

```
public boolean specialNumArray(int[] array) {
```

```
}
```

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

Writing: anagrams

Write a function that takes two Strings as input and returns a String. The first and last line are provided for you. **Read the hints.**

- The returned String should indicate if the two inputted Strings are anagrams or not (follow the example messages below).
Two Strings are anagrams if they contain the same letters (e.g., "later" and "latter" are NOT anagrams; "ten" and "net" ARE anagrams).

Hints: if you wish, you may use these built-in functions.

- For String `word`, `word.toCharArray()` will return an array of the characters in `word`.
- For array `array1`, `Arrays.sort(array1)` will change the original array by ordering the elements (it does not return anything, but the elements of `array1` will be ordered.)
- For arrays `array1` and `array2`, `Arrays.equals(array1, array2)` will return true if, for corresponding positions in each array, the elements are equal.

Examples:

Input	Output
<code>"abcd"</code> , <code>"adbc"</code>	'abcd' and 'adbc' have the same letters.
<code>"a"</code> , <code>"a"</code>	'a' and 'a' have the same letters.
<code>" "</code> , <code>" "</code>	" and " have the same letters.
<code>"abcd"</code> , <code>"adb"</code>	'abcd' and 'adb' do not have the same letters.
<code>"abda"</code> , <code>"adb"</code>	'abda' and 'adb' do not have the same letters.

```
public String anagrams(String word1, String word2) {
```

}

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

Writing: palindrome array

Write a function that takes an arrays of ints as input and returns a boolean. The first and last line are provided for you. **Read the hints.**

- Return true if the array is a palindrome with at least one element. An array is a palindrome if the first element is equal to the last element, and the second element is equal to the second-to-last element, etc.
- Return false otherwise.
- Do not change the original input array (that is, this function should be **non-destructive**).

Hints:

- Be careful about out-of-bounds exceptions. Remember that the last item in an array is at position `array.length - 1`.
- Initialize a new array with `int[] array = new int[size];`
- Find the length of an array with `array.length` (no parentheses).

Examples:

Input	Output
<code>[]</code>	FALSE
<code>[1]</code>	TRUE
<code>[2, 2]</code>	TRUE
<code>[1, 2, 3, 2, 1]</code>	TRUE
<code>[1, 2, 3, 2]</code>	FALSE

public boolean palindromeArray(int[] array) {

}

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

Writing: okNotOk

Write a function that takes two ints as input and returns a String. The first and last line are provided for you.

- When the first input divided by the second is 5 or larger, AND the first input is bigger than 10, return the String "Ok".
- Otherwise, return the String "Not Ok".

Hint: Be careful to avoid an exception caused by dividing by 0.

Examples:

Input	Output
36, 6	Ok
6, 36	Not Ok
2, 0	Not Ok

```
public String okNotOk(int num1, int num2) {
```

```
}
```

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

Writing: starts with A

Write a function that takes a String as input and returns a boolean. The first and last line are provided for you.

- When the input starts with "A", return true.
- Otherwise, return false.

Hint: `word.startsWith("x")` will return true when the String **word** starts with "x".

```
public boolean startsWithA(String word) {
```

```
}
```

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

Readability masterList

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐

```
public boolean func1() {  
    if (masterList.isEmpty())  
        return true;  
    return false;  
}
```
- ☐

```
public boolean func2() {  
    if (masterList.isEmpty())  
        return true;  
}
```

```

        else
            return false;
    }
    ☐ public boolean func3() {
        return masterList.isEmpty();
    }

```

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

master style

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐ public boolean func1() {
 if (masterList.isEmpty())
 return true;
 return false;
 }
☐ public boolean func2() {
 if (masterList.isEmpty())
 return true;
 else
 return false;
 }
☐ public boolean func3() {
 return masterList.isEmpty();
 }
☐ An expert would say they all have equal style.

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

Readability - discount - Oct 2018

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐ public static String discount1(String name, double price) {
 double discountPrice;
 double over50Discount = .2;
 double regDiscount = .1;
 double appliedDiscount;

 if (price > 50) {
 appliedDiscount = over50Discount;
 } else {
 appliedDiscount = regDiscount;
 }
 discountPrice = price * (1 - appliedDiscount);
 return "The original price of " + name
 + " was \$" + price
 + " but you get a discount of "
 + appliedDiscount
 + " so you only pay \$" + discountPrice;
 }
☐ public static String discount2(String name, double price) {
 double discountPrice;
 double over50Discount = .2;
 double regDiscount = .1;
 double appliedDiscount;

 if (price > 50) {
 discountPrice = price * (1 - over50Discount);
 appliedDiscount = over50Discount;
 } else {
 discountPrice = price * (1 - regDiscount);
 appliedDiscount = regDiscount;
 }
 return "The original price of " + name
 + " was \$" + price
 + " but you get a discount of "
 + appliedDiscount
 + " so you only pay \$" + discountPrice;
 }
☐

```
☒ public static String discount3(String name, double price) {
    double discountPrice;
    double over50Discount = .2;
    double regDiscount = .1;
    double appliedDiscount;

    if (price > 50) {
        discountPrice = price * (1 - over50Discount);
        appliedDiscount = over50Discount;
        return "The original price of " + name
            + " was $" + price
            + " but you get a discount of "
            + appliedDiscount
            + " so you only pay $" + discountPrice;
    } else {
        discountPrice = price * (1 - regDiscount);
        appliedDiscount = regDiscount;
        return "The original price of " + name
            + " was $" + price
            + " but you get a discount of "
            + appliedDiscount
            + " so you only pay $" + discountPrice;
    }
}
```

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

Style discount

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

```
☐ public static String discount1(String name, double price) {
    double discountPrice;
    double over50Discount = .2;
    double regDiscount = .1;
    double appliedDiscount;

    if (price > 50) {
        appliedDiscount = over50Discount;
    } else {
        appliedDiscount = regDiscount;
    }
    discountPrice = price * (1 - appliedDiscount);
    return "The original price of " + name
        + " was $" + price
        + " but you get a discount of "
        + appliedDiscount
        + " so you only pay $" + discountPrice;
}

☐ public static String discount2(String name, double price) {
    double discountPrice;
    double over50Discount = .2;
    double regDiscount = .1;
    double appliedDiscount;

    if (price > 50) {
        discountPrice = price * (1 - over50Discount);
        appliedDiscount = over50Discount;
    } else {
        discountPrice = price * (1 - regDiscount);
        appliedDiscount = regDiscount;
    }
    return "The original price of " + name
        + " was $" + price
        + " but you get a discount of "
        + appliedDiscount
        + " so you only pay $" + discountPrice;
}

☐ public static String discount3(String name, double price) {
    double discountPrice;
    double over50Discount = .2;
    double regDiscount = .1;
    double appliedDiscount;

    if (price > 50) {
        discountPrice = price * (1 - over50Discount);
        appliedDiscount = over50Discount;
        return "The original price of " + name
            + " was $" + price
            + " but you get a discount of "
            + appliedDiscount
            + " so you only pay $" + discountPrice;
    } else {
        discountPrice = price * (1 - regDiscount);
        appliedDiscount = regDiscount;
        return "The original price of " + name
            + " was $" + price
            + " but you get a discount of "
            + appliedDiscount
            + " so you only pay $" + discountPrice;
    }
}
```


- }
 }
☐ An expert would say they all have equal style.

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

bigger read

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐ public static int bigger1(int[] array, int num, int position) {
 if (!(position > -1 && position < array.length)) {
 return num;
 }
 if (array[position] > num) {
 return array[position];
 }
 return num;
 }
☐ public static int bigger2(int[] array, int num, int position) {
 if (position > -1 && position < array.length && array[position] > num) {
 return array[position];
 }
 return num;
 }
☐ public static int bigger3(int[] array, int num, int position) {
 if (position > -1) {
 if (position < array.length) {
 if (array[position] > num) {
 return array[position];
 }
 }
 return num;
 }
 return num;
 }
☐ public static int bigger4(int[] array, int num, int position) {
 if (position > -1) {
 if (position < array.length) {
 if (array[position] > num) {
 return array[position];
 }
 }
 }
 return num;
 }
 }

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

bigger style

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐ public static int bigger1(int[] array, int num, int position) {
 if (!(position > -1 && position < array.length)) {
 return num;
 }
 if (array[position] > num) {
 return array[position];
 }
 return num;
 }
☐ public static int bigger2(int[] array, int num, int position) {
 if (position > -1 && position < array.length && array[position] > num) {
 return array[position];
 }
 return num;
 }
☐ public static int bigger3(int[] array, int num, int position) {
 if (position > -1) {
 if (position < array.length) {
 if (array[position] > num) {
 return array[position];
 }
 }
 return num;
 }
 return num;
 }
 }
☐

```
public static int bigger4(int[] array, int num, int position) {
    if (position > -1) {
        if (position < array.length) {
            if (array[position] > num) {
                return array[position];
            }
        }
    }
    return num;
}
```

☐ An expert would say they all have equal style.

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

Style Readability - size - Oct 2018

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐

```
public static String size1(int i) {
    String size = "";
    if (i < 10) {
        size = "small";
    } else if (i >= 10 && i < 20) {
        size = "medium";
    } else if (i >= 20) {
        size = "big";
    }
    return "The size is " + size;
}
```
- ☐

```
public static String size2(int i) {
    String size = "";
    if (i < 10) {
        size = "small";
    }
    if (i >= 10 && i < 20) {
        size = "medium";
    }
    if (i >= 20) {
        size = "big";
    }
    return "The size is " + size;
}
```
- ☐

```
public static String size3(int i) {
    String size = "";
    if (i < 10) {
        size = "small";
    } else if (i < 20) {
        size = "medium";
    } else {
        size = "big";
    }
    return "The size is " + size;
}
```

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

size style

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐

```
public static String size1(int i) {
    String size = "";
    if (i < 10) {
        size = "small";
    } else if (i >= 10 && i < 20) {
        size = "medium";
    } else if (i >= 20) {
        size = "big";
    }
    return "The size is " + size;
}
```
- ☐

```
public static String size2(int i) {
    String size = "";
    if (i < 10) {
        size = "small";
```

- ```

 }
 if (i >= 10 && i < 20) {
 size = "medium";
 }
 if (i >= 20) {
 size = "big";
 }
 return "The size is " + size;
}
☐ public static String size3(int i) {
 String size = "";
 if (i < 10) {
 size = "small";
 } else if (i < 20) {
 size = "medium";
 } else {
 size = "big";
 }
 return "The size is " + size;
}
☐ An expert would say they all have equal style.

```

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

### Readability - Palindrome

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ```

☐ public static boolean array1p(int[] array) {
    if (array.length == 0) {
        return false;
    }
    if (array.length == 1) {
        return true;
    }
    if (array.length == 2) {
        if (array[0] == array[1])
            return true;
        return false;
    }

    for (int i = 0; i < array.length/2; i++) {
        if (array[i] != array[array.length - (1 + i)])
            return false;
    }
    return true;
}
☐ public static boolean array2p(int[] array) {
    if (array.length == 0) {
        return false;
    }

    for (int i = 0; i < array.length; i++) {
        if (array[i] != array[array.length - (1 + i)])
            return false;
    }
    return true;
}
☐ public static boolean array3p(int[] array) {
    if (array.length == 0) {
        return false;
    }
    if (array.length == 1) {
        return true;
    }
    if (array.length == 2) {
        return array[0] == array[1];
    }

    for (int i = 0; i < array.length/2; i++) {
        if (array[i] != array[array.length - (1 + i)])
            return false;
    }
    return true;
}
☐ public static boolean array4p(int[] array) {
    if (array.length == 0) {
        return false;
    }
    if (array.length == 1) {
        return true;
    }

    for (int i = 0; i < array.length/2; i++) {
        if (array[i] != array[array.length - (1 + i)])
            return false;
    }
    return true;
}

```

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

Style - Palindrome

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐ public static boolean array1p(int[] array) {
 if (array.length == 0) {
 return false;
 }
 if (array.length == 1) {
 return true;
 }
 if (array.length == 2) {
 if (array[0] == array[1])
 return true;
 return false;
 }

 for (int i = 0; i < array.length/2; i++) {
 if (array[i] != array[array.length - (1 + i)])
 return false;
 }
 return true;
}
- ☐ public static boolean array2p(int[] array) {
 if (array.length == 0) {
 return false;
 }
 }

 for (int i = 0; i < array.length; i++) {
 if (array[i] != array[array.length - (1 + i)])
 return false;
 }
 return true;
}
- ☐ public static boolean array3p(int[] array) {
 if (array.length == 0) {
 return false;
 }
 if (array.length == 1) {
 return true;
 }
 if (array.length == 2) {
 return array[0] == array[1];
 }
 }

 for (int i = 0; i < array.length/2; i++) {
 if (array[i] != array[array.length - (1 + i)])
 return false;
 }
 return true;
}
- ☐ public static boolean array4p(int[] array) {
 if (array.length == 0) {
 return false;
 }
 }
 if (array.length == 1) {
 return true;
 }
 }

 for (int i = 0; i < array.length/2; i++) {
 if (array[i] != array[array.length - (1 + i)])
 return false;
 }
 return true;
}
- ☐ An expert would say they all have equal style.

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

Readability - reverse array

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐ public static int[] arrayFunction1(int[] array) {
 if (array.length == 0)
 return new int[] {};
 if (array.length == 1)
 return new int[] { array[0] };
 if (array.length == 2) {
 if (array[0] == array[1])
 return new int[] { array[0], array[1] };
 else
 return new int[] { array[1], array[0] };
 }

 int[] array2 = new int[array.length];
 for (int i = 0; i < array.length; i++) {
 array2[i] = array[array.length - (1 + i)];
 }

 return array2;
}
- ☐ public static int[] arrayFunction2(int[] array) {
 int[] array2 = new int[array.length];

```

        for (int i = 0; i < array.length; i++) {
            array2[i] = array[array.length - (1 + i)];
        }

        return array2;
    }

    ☐ public static int[] arrayFunction3(int[] array) {
        if (array.length == 0)
            return new int[] {};
        if (array.length == 1)
            return new int[] { array[0] };
        if (array.length == 2)
            return new int[] { array[1], array[0] };

        int[] array2 = new int[array.length];
        for (int i = 0; i < array.length; i++) {
            array2[i] = array[array.length - (1 + i)];
        }

        return array2;
    }

    ☐ public static int[] arrayFunction4(int[] array) {
        if (array.length == 0)
            return new int[] {};

        int[] array2 = new int[array.length];
        for (int i = 0; i < array.length; i++) {
            array2[i] = array[array.length - (1 + i)];
        }

        return array2;
    }
}

```

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

Style - reverse array

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐ public static int[] arrayFunction1(int[] array) {
 if (array.length == 0)
 return new int[] {};
 if (array.length == 1)
 return new int[] { array[0] };
 if (array.length == 2) {
 if (array[0] == array[1])
 return new int[] { array[0], array[1] };
 else
 return new int[] { array[1], array[0] };
 }

 int[] array2 = new int[array.length];
 for (int i = 0; i < array.length; i++) {
 array2[i] = array[array.length - (1 + i)];
 }

 return array2;
}
- ☐ public static int[] arrayFunction2(int[] array) {
 int[] array2 = new int[array.length];
 for (int i = 0; i < array.length; i++) {
 array2[i] = array[array.length - (1 + i)];
 }

 return array2;
}
- ☐ public static int[] arrayFunction3(int[] array) {
 if (array.length == 0)
 return new int[] {};
 if (array.length == 1)
 return new int[] { array[0] };
 if (array.length == 2)
 return new int[] { array[1], array[0] };

 int[] array2 = new int[array.length];
 for (int i = 0; i < array.length; i++) {
 array2[i] = array[array.length - (1 + i)];
 }

 return array2;
}
- ☐ public static int[] arrayFunction4(int[] array) {
 if (array.length == 0)
 return new int[] {};

 int[] array2 = new int[array.length];
 for (int i = 0; i < array.length; i++) {
 array2[i] = array[array.length - (1 + i)];
 }

 return array2;
}
- ☐ An expert would say they all have equal style.

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

Readability - specialNum

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐ public boolean special1(int[] nums, int specialNum) {
 if (nums.length < 1) {
 return false;
 }
 if (nums.length == 1) {
 return nums[0] * nums[0] == specialNum;
 }
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
☐ public boolean special2(int[] nums, int specialNum) {
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
☐ public boolean special3(int[] nums, int specialNum) {
 if (nums.length < 1) {
 return false;
 }
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
☐ public boolean special1(int[] nums, int specialNum) {
 if (nums.length == 1) {
 return nums[0] * nums[0] == specialNum;
 }
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
 }

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

Style - specialNum

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐ public boolean special1(int[] nums, int specialNum) {
 if (nums.length < 1) {
 return false;
 }
 if (nums.length == 1) {
 return nums[0] * nums[0] == specialNum;
 }
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
☐

- ☐ public boolean special2(int[] nums, int specialNum) {
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
 }
☐ public boolean special3(int[] nums, int specialNum) {
 if (nums.length < 1) {
 return false;
 }
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
 }
☐ public boolean special1(int[] nums, int specialNum) {
 if (nums.length == 1) {
 return nums[0] * nums[0] == specialNum;
 }
 for (int i = 0; i < nums.length; i++) {
 for (int j = i; j < nums.length; j++) {
 if (nums[i] * nums[j] == specialNum) {
 return true;
 }
 }
 }
 return false;
 }
 }
☐ An expert would say they all have equal style.

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

Readability - anagrams - Oct 2018

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐ public static String anagram1(String first, String second) {
 if (first.isEmpty() && second.isEmpty()) {
 return first + " has the same letters as " + second;
 }
 else if (first.isEmpty() || second.isEmpty()) {
 return first + " and " + second + ": different letters."
 }
 if (first.length() != second.length()) {
 return first + " and " + second + ": different letters."
 }
 char letters1[] = first.toCharArray();
 char letters2[] = second.toCharArray();
 Arrays.sort(letters1);
 Arrays.sort(letters2);
 if (Arrays.equals(letters1, letters2)) {
 return first + " has the same letters as " + second;
 }
 return first + " and " + second + ": different letters."
 }
☐ public static String anagram2(String first, String second) {
 char letters1[] = first.toCharArray();
 char letters2[] = second.toCharArray();
 Arrays.sort(letters1);
 Arrays.sort(letters2);
 if (Arrays.equals(letters1, letters2)) {
 return first + " has the same letters as " + second;
 }
 return first + " and " + second + ": different letters."
 }
☐ public static String anagram3(String first, String second) {
 if (first.length() != second.length()) {
 return first + " and " + second + ": different letters."
 }
 char letters1[] = first.toCharArray();
 char letters2[] = second.toCharArray();
 Arrays.sort(letters1);
 Arrays.sort(letters2);
 if (Arrays.equals(letters1, letters2)) {
 return first + " has the same letters as " + second;
 }
 return first + " and " + second + ": different letters."
 }
☐ public static String anagram4(String first, String second) {
 if (first.isEmpty() && second.isEmpty()) {
 return first + " has the same letters as " + second;
 }
 else if (first.isEmpty() || second.isEmpty()) {
 return first + " and " + second + ": different letters."
 }
 char letters1[] = first.toCharArray();
 char letters2[] = second.toCharArray();
 Arrays.sort(letters1);
 Arrays.sort(letters2);
 if (Arrays.equals(letters1, letters2)) {
 return first + " has the same letters as " + second;
 }
 }
 }

```
        return first + " and " + second + ": different letters.";
    }
}
```

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

Style anagrams

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐

```
public static String anagram1(String first, String second) {
    if (first.isEmpty() && second.isEmpty()) {
        return first + " has the same letters as " + second;
    } else if (first.isEmpty() || second.isEmpty()) {
        return first + " and " + second + ": different letters.";
    }

    if (first.length() != second.length()) {
        return first + " and " + second + ": different letters.";
    }

    char letters1[] = first.toCharArray();
    char letters2[] = second.toCharArray();
    Arrays.sort(letters1);
    Arrays.sort(letters2);
    if (Arrays.equals(letters1, letters2)) {
        return first + " has the same letters as " + second;
    }
    return first + " and " + second + ": different letters.";
}
```
- ☐

```
public static String anagram2(String first, String second) {
    char letters1[] = first.toCharArray();
    char letters2[] = second.toCharArray();
    Arrays.sort(letters1);
    Arrays.sort(letters2);
    if (Arrays.equals(letters1, letters2)) {
        return first + " has the same letters as " + second;
    }
    return first + " and " + second + ": different letters.";
}
```
- ☐

```
public static String anagram3(String first, String second) {
    if (first.length() != second.length()) {
        return first + " and " + second + ": different letters.";
    }

    char letters1[] = first.toCharArray();
    char letters2[] = second.toCharArray();
    Arrays.sort(letters1);
    Arrays.sort(letters2);
    if (Arrays.equals(letters1, letters2)) {
        return first + " has the same letters as " + second;
    }
    return first + " and " + second + ": different letters.";
}
```
- ☐

```
public static String anagram4(String first, String second) {
    if (first.isEmpty() && second.isEmpty()) {
        return first + " has the same letters as " + second;
    } else if (first.isEmpty() || second.isEmpty()) {
        return first + " and " + second + ": different letters.";
    }

    char letters1[] = first.toCharArray();
    char letters2[] = second.toCharArray();
    Arrays.sort(letters1);
    Arrays.sort(letters2);
    if (Arrays.equals(letters1, letters2)) {
        return first + " has the same letters as " + second;
    }
    return first + " and " + second + ": different letters.";
}
```
- ☐

An expert would say they all have equal style.

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

Readability - Loops

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐

```
public boolean loops1(Matrix matrix) {
    if (matrix.numColumns != this.numColumns
        || matrix.numRows != this.numRows) {
        return false;
    }
}
```



```

int r = 0;
int c = 0;
while (r < this.numRows) {
    while (c < this.numColumns) {
        if (matrix.data[r][c] != this.data[r][c]) {
            return false;
        }
        c++;
    }
    r++;
    c = 0;
}
return true;
}

☐ public boolean loops2(Matrix matrix) {
    if (matrix.numColumns != this.numColumns
        || matrix.numRows != this.numRows) {
        return false;
    }

    for (int r = 0; r < this.numRows; r++) {
        for (int c = 0; c < this.numColumns; c++) {
            if (matrix.data[r][c] != this.data[r][c]) {
                return false;
            }
        }
    }
    return true;
}

☐ public boolean loops3(Matrix matrix) {
    if (matrix.numColumns != this.numColumns
        || matrix.numRows != this.numRows) {
        return false;
    }

    int r = 0;
    while (r < this.numRows) {
        for (int c = 0; c < this.numColumns; c++) {
            if (matrix.data[r][c] != this.data[r][c]) {
                return false;
            }
        }
        r++;
    }
    return true;
}

```

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

Style - Loops

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

```

☐ public boolean loops1(Matrix matrix) {
    if (matrix.numColumns != this.numColumns
        || matrix.numRows != this.numRows) {
        return false;
    }

    int r = 0;
    int c = 0;
    while (r < this.numRows) {
        while (c < this.numColumns) {
            if (matrix.data[r][c] != this.data[r][c]) {
                return false;
            }
            c++;
        }
        r++;
        c = 0;
    }
    return true;
}

☐ public boolean loops2(Matrix matrix) {
    if (matrix.numColumns != this.numColumns
        || matrix.numRows != this.numRows) {
        return false;
    }

    for (int r = 0; r < this.numRows; r++) {
        for (int c = 0; c < this.numColumns; c++) {
            if (matrix.data[r][c] != this.data[r][c]) {
                return false;
            }
        }
    }
    return true;
}

☐ public boolean loops3(Matrix matrix) {
    if (matrix.numColumns != this.numColumns
        || matrix.numRows != this.numRows) {

```

```

        return false;
    }
    int r = 0;
    while (r < this.numRows) {
        for (int c = 0; c < this.numColumns; c++) {
            if (matrix.data[r][c] != this.data[r][c]) {
                return false;
            }
        }
        r++;
    }
    return true;
}

```

☐ An expert would say they all have equal style.

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

head null read

Readability: All of these code samples do the same thing. Which one is most readable to you: which one makes it easiest for YOU to figure out what the code does?

- ☐ public boolean func1() {
 if(head == null)
 return true;
 return false;
 }
- ☐ public boolean func2() {
 if (head == null)
 return true;
 else
 return false;
 }
- ☐ public boolean func3() {
 return head == null;
 }

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

head null style

Style: These are the same code samples as the previous question. All of these code samples do the same thing. **Which one would an expert say has the best style?** Style is the tasteful use of language that makes code elegant, efficient, idiomatic, and revealing of design intent.

- ☐ public boolean func1() {
 if(head == null)
 return true;
 return false;
 }
- ☐ public boolean func2() {
 if (head == null)
 return true;
 else
 return false;
 }
- ☐ public boolean func3() {
 return head == null;
 }
- ☐ An expert would say they all have equal style.

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

Comprehension - anagrams2a

For the function below, what is the output for each of these inputs?

What is the output for this input: "", "key"

- ☐ " * key"
- ☐ " / key"
- ☐ It will throw a runtime exception.

What is the output for this input: "yak", "kay"

- ☐ "yak * kay"
- ☐ "yak / kay"
- ☐ It will throw a runtime exception.

What is the output for this input: "", ""

- ☐ " * "
- ☐ " / "
- ☐ It will throw a runtime exception.

What is the output for this input: "deer", "sled"

- ☐ "deer * sled"
- ☐ "deer / sled"
- ☐ It will throw a runtime exception.

```
public static String anagram1(String first, String second) {  
    if (first.isEmpty() && second.isEmpty()) {  
        return first + " * " + second;  
    } else if (first.isEmpty() || second.isEmpty()) {  
        return first + " / " + second;  
    }  
    if (first.length() != second.length()) {  
        return first + " / " + second;  
    }  
    char letters1[] = first.toCharArray();  
    char letters2[] = second.toCharArray();  
    Arrays.sort(letters1);  
    Arrays.sort(letters2);  
    if (Arrays.equals(letters1, letters2)) {  
        return first + " * " + second;  
    }  
    return first + " / " + second;  
}
```

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

Comprehension - palindrome2a

For the function below, what is the output for each of these inputs?

What is the output for this input: [3, 3]

- ☐ true
- ☐ false
- ☐ It will throw a runtime exception.

What is the output for this input: [3, 6, 6]

- ☐ true
- ☐ false
- ☐ It will throw a runtime exception.

What is the output for this input: []

- ☐ true
- ☐ false
- ☐ It will throw a runtime exception.

```
public static boolean arrayIp(int[] array) {  
    if (array.length == 0) {  
        return false;  
    }  
    if (array.length == 1) {  
        return true;  
    }  
    if (array.length == 2) {  
        if (array[0] == array[1])  
            return true;  
        return false;  
    }  
  
    for (int i = 0; i < array.length/2; i++) {  
        if (array[i] != array[array.length - (1 + i)])  
            return false;  
    }  
    return true;  
}
```

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

Comprehension - reverse2

For the function below, what is the output for each of these inputs?

What is the output for this input: [6]

- ☐ [0]
- ☐ [1]
- ☐ Another output:
- ☐ Throws an exception

What is the output for this input: [1, 2, 3]

- ☐ [4, 3, 2]
- ☐ [3, 2, 1]
- ☐ Another output:
- ☐ Throws an exception

What is the output for this input: [2, 2]

- ☐ [2, 2]
- ☐ [-1, -1]
- ☐ Another output:
- ☐ Throws an exception

What is the output for this input: []

- ☐ []
- ☐ [0]
- ☐ Another output:
- ☐ Throws an exception

```
public static int[] arrayFunction2(int[] array) {  
    if (array.length == 0)  
        return new int[] {};  
    if (array.length == 1)  
        return new int[] { array[0] };  
    if (array.length == 2) {  
        if (array[0] == array[1])  
            return new int[] { array[0], array[1] };  
        else  
            return new int[] { array[1], array[0] };  
    }  
  
    int[] array2 = new int[array.length];  
    for (int i = 0; i < array.length; i++) {  
        array2[i] = array[array.length - (1 + i)];  
    }  
  
    return array2;  
}
```

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

Comprehension - specialNum2 - Oct 2018

For the function below, what is the output for each of these inputs?

What is the output for this input: [], 16

True
☐

False
☐

It will throw a runtime exception
☐

What is the output for this input: [4], 16

True
☐

False
☐

It will throw a runtime exception
☐

What is the output for this input: [2, 8, 11], 16

True
☐

False
☐

It will throw a runtime exception
☐

```
public boolean special2(int[] nums, int specialNum) {  
    if (nums.length < 1) {  
        return false;  
    }  
    if (nums.length == 1) {  
        return nums[0] * nums[0] == specialNum;  
    }  
    for (int i = 0; i < nums.length; i++) {  
        for (int j = i; j < nums.length; j++) {  
            if (nums[i] * nums[j] == specialNum) {  
                return true;  
            }  
        }  
    }  
}
```

```
    }  
  }  
  return false;  
}
```

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

Comprehension: greater

Decide if the code below accomplishes this task:

Task: The function takes as input a String (word), a char (letter) and an int (position), and returns a char.

- Return the letter at the inputted position for the inputted word IF that letter is larger than the inputted letter.
- Otherwise, return the given letter.

Examples:

Input	Output
"Word", 'A', 0	'W'
"Word", 'Z', 0	'Z'
"Word", 'A', -1	'A'
"Word", 'A', 10	'A'

```
public static char greater5(String word, char letter, int position) {  
    if (position > -1 && word.charAt(position) > letter && position < word.length()) {  
        return word.charAt(position);  
    }  
    return letter;  
}
```

Does the code above accomplish the task?

- ☐ Yes
☐ No

What is the output for this input: "Word", 'A', 0

- ☐ 'A'
☐ 'W'
☐ Another output:
☐ Throws an exception

You indicated that the code does not accomplish the task. Give an example input where the output is not correct (the checkboxes will check themselves as you fill in the text boxes - leave them checked).

- ☒ word (String):
☒ letter (char):
☒ position (int):
☒ desired output:
☒ actual output or exception:

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

Comprehension: greater 2

Decide if the code blocks below do the same thing. That is, for the same inputs, do they all have the same outputs (or throw the same exceptions)?

```
public static char greater1(String word, char letter, int position) {
    if (position > -1 && position < word.length() && word.charAt(position) > letter) {
        return word.charAt(position);
    }
    return letter;
}
```

```
public static char greater2(String word, char letter, int position) {
    if (position < word.length()) {
        if (position > -1) {
            if (word.charAt(position) > letter) {
                return word.charAt(position);
            }
        }
    }
    return letter;
}
```

```
public static char greater3(String word, char letter, int position) {
    if (position > -1) {
        if (word.charAt(position) > letter) {
            if (position < word.length()) {
                return word.charAt(position);
            }
        }
    }
    return letter;
}
```

Choose the statement that you agree with:

- ☐ All of the functions do the same thing.
- ☐ Two of the functions do the same thing, and one does something else.
- ☐ All three functions do different things.

You said that two of the functions did one thing, and one did something else. Which functions do the SAME thing? The functions are copied below.

- ☐ greater1 and greater2
- ☐ greater1 and greater3
- ☐ greater2 and greater3

```
public static char greater1(String word, char letter, int position) {
    if (position > -1 && position < word.length() && word.charAt(position) > letter) {
        return word.charAt(position);
    }
    return letter;
}
```

```
public static char greater2(String word, char letter, int position) {
    if (position < word.length()) {
        if (position > -1) {
            if (word.charAt(position) > letter) {
                return word.charAt(position);
            }
        }
    }
    return letter;
}
```

```
public static char greater3(String word, char letter, int position) {  
    if (position > -1) {  
        if (word.charAt(position) > letter) {  
            if (position < word.length()) {  
                return word.charAt(position);  
            }  
        }  
    }  
    return letter;  
}
```

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

Comprehension: greater 2.1

You said that all three functions did different things. Please provide the inputs and outputs that show the differences between the functions. If both functions throw an exception on your input, include details on how the exceptions are different.

There are spaces for three inputs. Please fill them all in. They do not need to be unique if one input shows the differences for multiple comparisons.

Give an example input where greater1 and greater2 have different outputs (or throw different exceptions). The functions are copied below (the checkboxes will check themselves as you fill in the text boxes - leave them checked).

- ☐ word (String):
- ☐ letter (char):
- ☐ position (int):
- ☐ output or exception for greater1:
- ☐ output or exception for greater2:

Give an example input where greater1 and greater3 have different outputs (or throw different exceptions). The functions are copied below (the checkboxes will check themselves as you fill in the text boxes - leave them checked).

- ☐ word (String):
- ☐ letter (char):
- ☐ position (int):
- ☐ output or exception for greater1:
- ☐ output or exception for greater3:

Give an example input where greater2 and greater3 have different outputs (or throw different exceptions). The functions are copied below (the checkboxes will check themselves as you fill in the text boxes - leave them checked).

- ☐ word (String):
- ☐ letter (char):
- ☐ position (int):
- ☐ output or exception for greater2:
- ☐ output or exception for greater3:

For the function greater1 (below), what is the output for this input: "Word", 'A', 10

- ☐ 'A'
- ☐ 'W'
- ☐ Another output:
- ☐ Throws an exception

```
public static char greater1(String word, char letter, int position) {
    if (position > -1 && position < word.length() && word.charAt(position) > letter) {
        return word.charAt(position);
    }
    return letter;
}
```

For the function greater2 (below), what is the output for this input: "Word", 'A', 10

- ☐ 'A'
- ☐ 'W'
- ☐ Another output:
- ☐ Throws an exception

```
public static char greater2(String word, char letter, int position) {
    if (position < word.length()) {
        if (position > -1) {
            if (word.charAt(position) > letter) {
                return word.charAt(position);
            }
        }
    }
    return letter;
}
```

For the function greater3 (below), what is the output for this input: "Word", 'A', 10

- ☐ 'A'
- ☐ 'W'
- ☐ Another output:
- ☐ Throws an exception

```
public static char greater3(String word, char letter, int position) {
    if (position > -1) {
        if (word.charAt(position) > letter) {
            if (position < word.length()) {
                return word.charAt(position);
            }
        }
    }
    return letter;
}
```

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

Comprehension - word 1-2 - Oct 2018

Do the two functions below do the same thing? That is, if they are each given the same input, will they have the same output (or throw the same exception)?

- ☐ Yes, the functions do the same things.
- ☐ No, the functions do different things.

```
public static String word1(String word) {
    if (word.endsWith("?")) {
        word = word.substring(0, word.length() - 1);
    }
    if (word.endsWith("!")) {
        word = word.substring(0, word.length() - 1);
    }
    return word;
}

public static String word2(String word) {
    if (word.endsWith("?")) {
        word = word.substring(0, word.length() - 1);
    } else if (word.endsWith("!")) {
        word = word.substring(0, word.length() - 1);
    }
    return word;
}
```

You said the functions do different things. Give an input where the functions will have different outputs or throw different exceptions (the checkboxes will check themselves as you fill in the text boxes - leave them checked).

- ☐ Input:
- ☐ word1 output or exception:
- ☐ word2 output or exception:

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

Comprehension - word 1-2 comparison - Oct 2018

These are the same functions as the previous question. Which will perform more comparison operations for this input: "Question?"

```
public static String word1(String word) {
    if (word.endsWith("?")) {
        word = word.substring(0, word.length() - 1);
    }
    if (word.endsWith("!")) {
        word = word.substring(0, word.length() - 1);
    }
    return word;
}
```

```
public static String word2(String word) {
    if (word.endsWith("?")) {
        word = word.substring(0, word.length() - 1);
    } else if (word.endsWith("!")) {
        word = word.substring(0, word.length() - 1);
    }
    return word;
}
```

Which function will perform more comparison operations for this input: "Question?"

- ☐ word1
- ☐ word2
- ☐ They will both perform the same number of comparison operations.

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

Comprehension - combo 1-2

Do these two functions do the same thing?

```
public static String combo1(int[] nums1, int[] nums2) {
    String combinations = "";
    for (int i = 0; i < nums1.length; i++) {
        for (int j = 0; j < nums2.length; j++) {
            combinations += " (" + nums1[i] + ", " + nums2[j] + ")";
        }
        combinations += "\n";
    }
    return combinations;
}
```

```
public static String combo2(int[] nums1, int[] nums2) {
    String combinations = "";
    int i = 0;
    int j = 0;
    while ( i < nums1.length) {
        while (j < nums2.length) {
            combinations += " (" + nums1[i] + ", " + nums2[j] + ")";
            j++;
        }
        combinations += "\n";
        i++;
    }
    return combinations;
}
```

Do the two functions above do the same thing? That is, if they are each given the same input, will they have the same output (or throw the same exception)?

- ☐ Yes, the functions do the same things.
- ☐ No, the functions do different things.

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

Comprehension loops1-2

Do these two functions do the same thing?

```
public static int[] loops1(int[] array) {
    for (int i = 2; i < array.length; i++) {
        array[i] = array[i] + 2;
    }
    return array;
}
```

```
public static int[] loops2(int[] array) {
    int i = 2;
    while (i < array.length){
```

```
        array[i] = array[i] + 2;
        i++;
    }
    return array;
}
```

Do the two functions below do the same thing? That is, if they are each given the same input, will they have the same output (or throw the same exception)?

- ☐ Yes, the functions do the same things.
- ☐ No, the functions do different things.

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

Comprehension - magic - Oct 2018

Do these two functions do the same thing? That is, if they are each given the same input, will they have the same output (or throw the same exception)?

```
public static String magic1(int i) {
    int out = 0;
    if (i > 100) {
        out = i + 15;
    }
    if (i < 100) {
        out = i - 15;
    }
    if (i == 100) {
        out = i + 37;
    }
    return "Your number is " + i + " and your magic number is: " + out;
}
```

```
public static String magic2(int i) {
    int out = 0;
    if (i > 100) {
        out = i + 15;
    } else if (i < 100) {
        out = i - 15;
    } else {
        out = i + 37;
    }
    return "Your number is " + i + " and your magic number is: " + out;
}
```

- ☐ Yes, the functions above do the same things.
- ☐ No, the functions above do different things.

You said the functions do different things. Give an input where the functions will have different outputs or throw different exceptions (the checkboxes will check themselves as you fill in the text boxes - leave them checked).

- ☐ Input:
- ☐ magic1 output or exception:
- ☐ magic2 output or exception:

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

Comprehension - magic comparison - Oct 2018

These are the same functions as the previous question. For the inputs below, which function will perform more comparison operations?

```
public static String magic1(int i) {
    int out = 0;
    if (i > 100) {
        out = i + 15;
    }
    if (i < 100) {
        out = i - 15;
    }
    if (i == 100) {
        out = i + 37;
    }
    return "Your number is " + i + " and your magic number is: " + out;
}
```

```
public static String magic2(int i) {
    int out = 0;
    if (i > 100) {
        out = i + 15;
    } else if (i < 100) {
        out = i - 15;
    } else {
        out = i + 37;
    }
    return "Your number is " + i + " and your magic number is: " + out;
}
```

Which function will perform more comparison operations for this input: 100

- ☐ magic1
- ☐ magic2
- ☐ They will both perform the same number of comparisons.

Which function will perform more comparison operations for this input: 110

- ☐ magic1
- ☐ magic2
- ☐ They will both perform the same number of comparisons.

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

Comprehension - anagrams1a

For the function below, what is the output for each of these inputs?

```
public static String anagram1(String first, String second) {
    char letters1[] = first.toCharArray();
    char letters2[] = second.toCharArray();
    Arrays.sort(letters1);
    Arrays.sort(letters2);
    if (Arrays.equals(letters1, letters2)) {
        return first + " before " + second;
    }
}
```

```
    return first + " precedes " + second;
}
```

What is the output for this input: "phone", ""

- ☐ "phone before "
- ☐ "phone precedes "
- ☐ It will throw a runtime exception.

What is the output for this input: "staple", "pleats"

- ☐ "staple before pleats"
- ☐ "staple precedes pleats"
- ☐ It will throw a runtime exception.

What is the output for this input: "", ""

- ☐ " before "
- ☐ " precedes "
- ☐ It will throw a runtime exception.

What is the output for this input: "puppy", "kitty"

- ☐ "puppy before kitty"
- ☐ "puppy precedes kitty"
- ☐ It will throw a runtime exception.

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*

Comprehension - palindrome1a

For the function below, what is the output for each of these inputs?

```
public static boolean arraylp(int[] array) {
    if (array.length == 0) {
        return false;
    }

    for (int i = 0; i < array.length/2; i++) {
        if (array[i] != array[array.length - (1 + i)])
            return false;
    }
    return true;
}
```

What is the output for this input: [7, 7]

- ☐ true
- ☐ false
- ☐ It will throw a runtime exception.

What is the output for this input: [1, 2, 2]

- ☐ true
- ☐ false
- ☐ It will throw a runtime exception.

What is the output for this input: []

- ☐ true
- ☐ false
- ☐ It will throw a runtime exception.

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

Comprehension - reverse1

For the function below, what is the output for each of these inputs?

```
public static int[] arrayFunction1(int[] array) {  
    int[] array2 = new int[array.length];  
    for (int i = 0; i < array.length; i++) {  
        array2[i] = array[array.length - (1 + i)];  
    }  
  
    return array2;  
}
```

What is the output for this input: [1]

- ☐ [0]
- ☐ [1]
- ☐ Another output:
- ☐ Throws an exception

What is the output for this input: [15, 20, 25]

- ☐ [25, 20, 15]
- ☐ [16, 21, 26]
- ☐ Another output:
- ☐ Throws an exception

What is the output for this input: [10, 10]

- ☐ [9, 10]
- ☐ [11, 11]
- ☐ Another output:
- ☐ Throws an exception

What is the output for this input: []

- ☐ [0]
- ☐ []
- ☐ Another output:
- ☐ Throws an exception

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

Comprehension - specialNum1 - Oct 2018

For the function below, what is the output for each of these inputs?

What is the output for this input: [], 36

True
☐

False
☐

It will throw a runtime exception
☐

What is the output for this input: [6], 36

True
☐

False
☐

It will throw a runtime exception
☐

What is the output for this input: [3, 12, 19], 36

True
☐

False
☐

It will throw a runtime exception
☐

```
public boolean special1(int[] nums, int specialNum) {
    for (int i = 0; i < nums.length; i++) {
        for (int j = i; j < nums.length; j++) {
            if (nums[i] * nums[j] == specialNum) {
                return true;
            }
        }
    }
    return false;
}
```

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

Edit: between

Please copy and paste the function into the text box below, and then edit the function so that it has good style, as an expert would view it. Do not change the functionality (that is, if your new code and the original code were called with the same input, they should have the same output).

If the function is already written with the best possible style, simply copy and paste it without editing.

Note: you do not need to add comments.

```
public static boolean between(int min, int max, int num) {
    if (min < num) {
        if (max > num) {
            return true;
        }
    }
    return false;
}
```

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

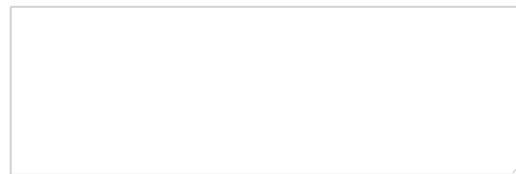
Edit: ending3

Please copy and paste the function into the text box below, and then edit the function so that it has good style, as an expert would view it. Do not change the functionality (that is, if your new code and the original code were called with the same input, they should have the same output).

If the function is already written with the best possible style, simply copy and paste it without editing.

Note: you do not need to add comments.

```
public static boolean ending(String word) {  
    if (word.endsWith("ing") || word.endsWith("ed"))  
        return true;  
    else  
        return false;  
}
```



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

Edit: diff

When `diff("Gzape", "Grapple")` is called, the function correctly returns "G+ap-". When `diff("Grapple", "Gzape")` is called, the desired output is "G-ap+_" [note the two underscores at the end]. However, for this input, the code throws a runtime exception. The message reads "Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 5". Please copy and paste the function into the text box below, and then fix the function so that it will run correctly without errors. Aim for a solution that has good style, as an expert would view it.

Note: you do not need to add comments.

```
public static String diff(String item, String otherItem) {  
    String difference = "";  
    for (int i = 0; i < item.length(); i++) {  
        if (i >= otherItem.length()) {  
            difference += "_";  
        }  
        if (item.charAt(i) < otherItem.charAt(i)) {  
            difference += "-";  
        }  
        if (item.charAt(i) > otherItem.charAt(i)) {  
            difference += "+";  
        }  
        if (item.charAt(i) == otherItem.charAt(i)) {  
            difference += item.charAt(i);  
        }  
    }  
    return difference;  
}
```

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

Edit: Special Num

Copy and paste the code into the box below, and then **edit this function to improve the code's style. Do not change the functionality** (that is, if your new code and the original code were called with the same input, they should have the same output).

Aim for a solution that has good style, as an expert would view it. In particular, remove any unnecessary code.

This code checks if there exist two numbers in the array that multiply to the Special Number,
OR if one of the numbers in the array is the square root of the Special Number.

Input	Output
[], 100	false
[15, 25, 6, 32], 100	false
[10], 100	true
[10, 50, 2, 3], 100	true
[25, 0, 1, 2, 3, 4], 100	true

```
public boolean special2(int[] nums, int specialNum) {  
    //if the array is empty, always FALSE  
    if (nums.length < 1) {  
        return false;  
    }  
  
    //if the array has only one element, check if it is the square root  
    if (nums.length == 1) {  
        return nums[0] * nums[0] == specialNum;  
    }  
  
    //the array has more than one element  
    //Iterate through the array, multiplying each element by itself  
    //and all elements that come after it  
    for (int i = 0; i < nums.length; i++) {  
        for (int j = i; j < nums.length; j++) {  
            if (nums[i] * nums[j] == specialNum) {  
                return true;  
            }  
        }  
    }  
    return false;  
}
```

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

Edit: Anagram

Copy and paste the code into the box below, and then **edit this function to improve the code's style. Do not change the functionality** (that is, if your new code and the original code were called with the same input, they should have the same output).

Aim for a solution that has good style, as an expert would view it. In particular, remove any unnecessary code.

This code checks if the two inputted Strings have the same letters (regardless of order).

Input	Output
"", ""	" has the same letters as "
"ear", "are"	"ear has the same letters as are"
"", "hat"	" and hat: different letters"
"ear", "earr"	"ear and earr: different letters"

```
public static String anagram1(String first, String second) {  
    //if both Strings are empty, they have the same letters.  
    //if only one String is empty, they have different letters.  
    if (first.isEmpty() && second.isEmpty()) {  
        return first + " has the same letters as " + second;  
    } else if (first.isEmpty() || second.isEmpty()) {  
        return first + " and " + second + ": different letters."  
    }  
  
    //if the Strings are not the same length, they can't have the same letters.  
    if (first.length() != second.length()) {  
        return first + " and " + second + ": different letters."  
    }  
  
    //Both Strings have the same number of letters  
    char letters1[] = first.toCharArray();  
    char letters2[] = second.toCharArray();
```

```
//Sort the arrays of characters in each String, and then decide if the arrays are equal.
Arrays.sort(letters1);
Arrays.sort(letters2);
if (Arrays.equals(letters1, letters2)) {
    return first + " has the same letters as " + second;
}
return first + " and " + second + ": different letters.";
}
```

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

Edit: palindromeArray1

Copy and paste the code into the box below, and then **edit this function to improve the code's style. Do not change the functionality** (that is, if your new code and the original code were called with the same input, they should have the same output).

Aim for a solution that has good style, as an expert would view it. In particular, remove any unnecessary code.

This code checks if the input array is a palindrome.

Input	Output
[]	false
[1]	true
[2, 2]	true
[1, 2, 3, 2, 1]	true
[1, 2, 3, 2]	false

```
public static boolean palindromeArray(int[] array) {
//an array is never a palindrome if it is length 0
    if (array.length == 0)
        return false;

//an array is always a palindrome if it is length 1
```

```
    if (array.length == 1)
        return true;

//if an array is length 2, it is only a palindrome if both items are the same
    if (array.length == 2) {
        if (array[0] == array[1])
            return true;
        return false;
    }

//iterate through the array, comparing the first item and last item,
//then second item with second-to-last, etc. checking that each pair is the same.
//if any pair does not match, the array is not a palindrome
    for (int i = 0; i < array.length/2; i++) {
        if (array[i] != array[array.length - (1 + i)])
            return false;
    }
    return true;
}
```

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

Edit: reverse

Copy and paste the code into the box below, and then **edit this function to improve the code's style. Do not change the functionality** (that is, if your new code and the original code were called with the same input, they should have the same output). This code does not change the original input array - your code shouldn't either.

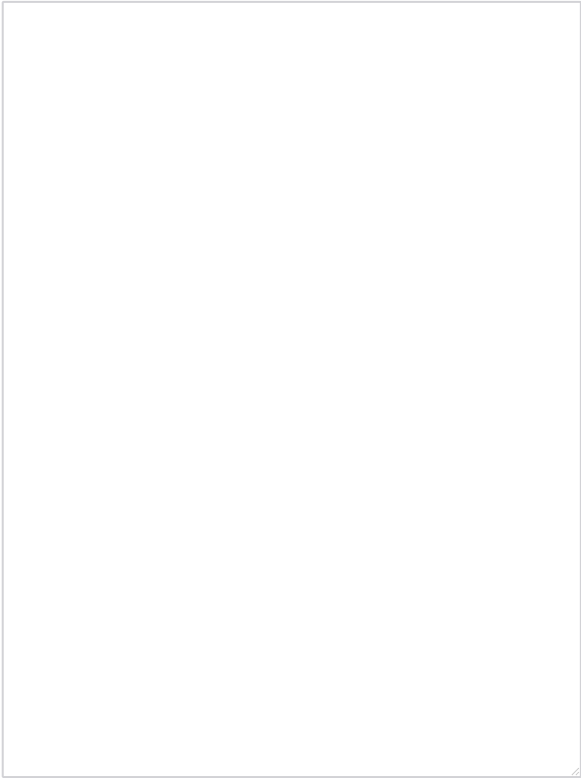
Aim for a solution that has good style, as an expert would view it. In particular, remove any unnecessary code.

This code reverses an array.

Input	Output
[]	[]
[1]	[1]
[2, 2]	[2, 2]

[1, 2, 3]	[3, 2, 1]
[4, 5, 6, 7]	[7, 6, 5, 4]

```
public static int[] arrayFunction1(int[] array) {  
    //if the array is length 0, return an array of length 0.  
    if (array.length == 0)  
        return new int[] {};  
  
    //if the array has just one element, return an array with that element.  
    if (array.length == 1)  
        return new int[] { array[0] };  
  
    //if the array has two elements and they are the same, return them in that order.  
    //if the elements are different, return them with the order switched.  
    if (array.length == 2) {  
        if (array[0] == array[1])  
            return new int[] { array[0], array[1] };  
        else  
            return new int[] { array[1], array[0] };  
    }  
  
    //The array has more than two elements. Make a new array with the same length.  
    //Iterate through the new array, putting elements in from the end of the original array  
    //so the first element of the new array was the last element of the original array, etc.  
    int[] array2 = new int[array.length];  
    for (int i = 0; i < array.length; i++) {  
        array2[i] = array[array.length - (1 + i)];  
    }  
  
    return array2;  
}
```



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

Edit: sales1

The code below returns a message saying the kind of discount a customer will get, based on what item they are purchasing, the price, and if they have a coupon.

Copy and paste the code into the box below, and then **edit this function to improve the code's style. Do not change the functionality.** If your new code and the original code were called with the same input, they should have the same output.

Edit the code to make it easy for a future programmer to make these future modifications:

- Adding "Congratulations!!" to all messages
- Adding "you saved X!" to the message (where X is the amount that the customer saved).
- **Do not actually make either of these changes.**

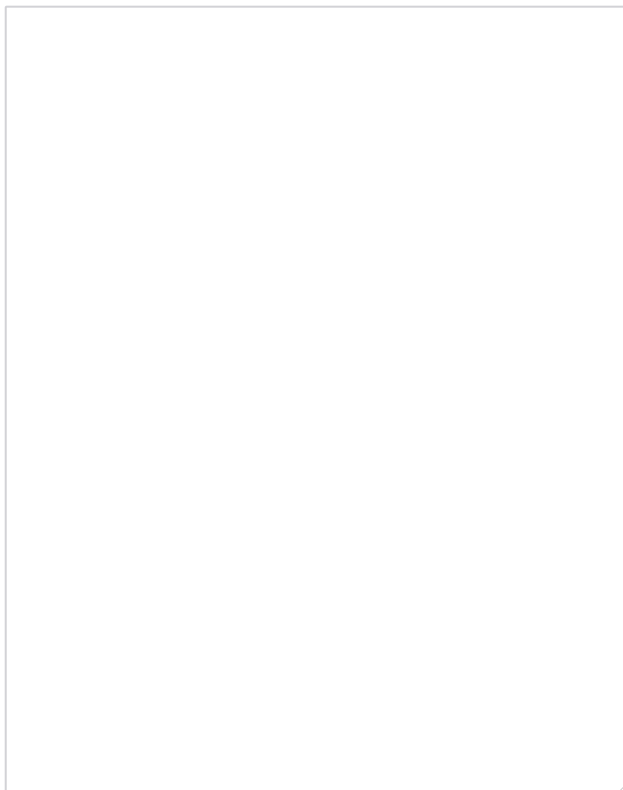
Aim for a solution that has good style, as an expert would view it. In particular, remove any unnecessary code.

In this task, are you supposed to change the functionality of the code? (the rest of the question will appear after you answer correctly. Do not change your answer after you answer correctly).

- ☐ Yes, I need to make the modifications described above.
- ☐ No, I need to keep the current functionality exactly the same.

```
public static String salePrice(String item, double price, boolean coupon) {
    double sale = .25;
    double specialSale = .5;
    double over50sale = .35;

    String beginning = "Your item, " + item + ", usually costs " + price
        + ", but ";
    String ending = "you are getting it for ";
    if (item.equals("socks")) {
        if (coupon) {
            return beginning + "since you have a coupon, "
                + ending + price * (1 - specialSale);
        }
    }
    if (price > 50) {
        return beginning + "since it is over 50, "
            + ending + price * (1 - over50sale);
    }
    if (price <= 50) {
        return beginning
            + ending + price * (1 - sale);
    }
    return "Error.";
}
```



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

revision chance isSeven

Your answer to an earlier question is copied below. The task was to write a function that returns true when the input is 7, and false otherwise.

```
public boolean isSeven(int num) {  
  ${q://QID83/ChoiceTextEntryValue}  
}
```

Copy and paste your code into the box below, and see if you can improve its style. If you see any other errors, please correct them.



revision plan isSeven

Your revision is copied below. The task was to write a function that returns true when the input is 7, and false otherwise.

```
${q://QID327/ChoiceTextEntryValue}
```

Can you improve the style of your code by re-writing it without an `if` statement?

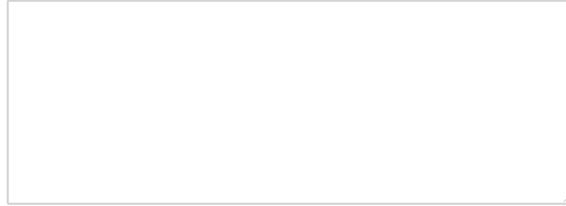
- ☐ Yes, I could do that.
- ☐ No, I don't know how to do that.

revision isSeven

Your answer to an earlier question is copied below. The task was to write a function that returns true when the input is 7, and false otherwise.

```
${q://QID327/ChoiceTextEntryValue}
```

Copy and paste your code into the box below, and then edit your code so that it does not include an `if` statement. If you notice any other errors, please correct them.

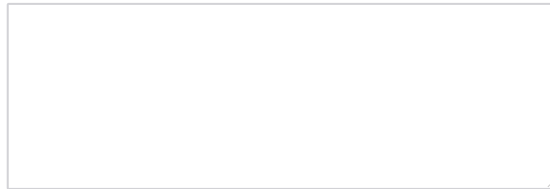


Check out this example - these two code blocks do the same thing.

```
public static boolean isGreaterThan9a(int num) {  
    if (num > 9)  
        return true;  
    return false;  
}
```

```
public static boolean isGreaterThan9b(int num) {  
    return num > 9;  
}
```


Copy and paste your code into the box below, then follow the example to remove the `if` statement. If you notice any other errors, please correct them.

**revision chance startsWithA**

Your answer to an earlier question is copied below. The task was to write a function that returns true when the input String starts with "A", and false otherwise.

```
public boolean startsWithA(String word) {  
${q://QID71/ChoiceTextEntryValue}  
}
```

Copy and paste your code into the box below, and see if you can improve its style. If you see any other errors, please correct them.

**revision plan starts with A**

Your answer to an earlier question is copied below. The task was to write a function that returns true when the input String starts with "A", and false otherwise.

```
${q://QID352/ChoiceTextEntryValue}
```

Can you improve the style of your code by re-writing it without an `if` statement?

- ☐ Yes, I could do that.
- ☐ No, I don't know how to do that.

revision startsWithA

Your answer to an earlier question is copied below. The task was to write a function that returns true when the input is 7, and false otherwise.

`${q://QID352/ChoiceTextEntryValue}`

Copy and paste your code into the box below, and then edit your code so that it does not include an `if` statement. If you notice any other errors, please correct them.

Check out this example - these two code blocks do the same thing.

```
public static boolean equalsABC1(String word) {  
    if (word.equals("ABC"))  
        return true;  
    return false;  
}
```

```
public static boolean equalsABC2(String word) {  
    return word.equals("ABC");  
}
```

Copy and paste your code into the box below, then follow the example to remove the `if` statement. If you notice any other errors, please correct them.

Revision Chance Fish-ishness

This is your code from a previous question. The question prompt is below. Copy your code into the box below and try to improve the code's style. If you see any other errors, please fix them.

```
public String ishness(String word) {  
    ${q://QID74/ChoiceTextEntryValue}  
}
```

Write a function that takes a `String` as input and returns a `String`. The first and last lines are provided for you.

- For input `Strings` that end in "sh", concatenate the ending "-ishness" and return a message saying how long the new word is.
- For all other input `Strings`, return a message saying how long the given word is.
- Write the function so that it would **easy for someone else to modify** (e.g., if someone else wanted to look for `Strings` that end with different letters, or concatenate a different ending)

HINTS:

- `word.endsWith("sh")` will return true if the `String word` ends with "sh".
- The messages should be **returned**, not printed.
- You may use `+` to concatenate strings. For example:

```
String word2 = word1 + " there";
```

If word1 is "hello", word2 is "hello there".

The messages should follow the format of the examples:

Input	Output
Fish	Your word was Fish. The length of Fish-ishness is 12.
Hat	Your word was Hat. The length of Hat is 3.

Copy your code into the box below and try to improve the code's style. If you see any other errors, please fix them.

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

Revision Plan Fish-ishness

Your code from a previous question is copied below, along with the question prompt. We detected at least one of these things about your code:

- The code has an `else`
- The code has more than one `return`
- The code has a `print` statement

For correct functionality, there should be no `print` statement. For optimal style, there should be no `else`, and only one `return` statement.

- ☐ I can edit my code for optimal style and/or correct functionality.
- ☐ I don't know how to edit my code to do that.

`${q://QID360/ChoiceTextEntryValue}`

Write a function that takes a String as input and returns a String. The first and last lines are provided for you.

- For input Strings that end in "sh", concatenate the ending "-ishness" and return a message saying how long the new word is.
- For all other input Strings, return a message saying how long the given word is.
- Write the function so that it would **easy for someone else to modify** (e.g., if someone else wanted to look for Strings that end with different letters, or concatenate a different ending)

HINTS:

- `word.endsWith("sh")` will return true if the String `word` ends with "sh".
- The messages should be **returned**, not printed.
- You may use `+` to concatenate strings. For example:

```
String word2 = word1 + " there";  
If word1 is "hello", word2 is "hello there".
```

The messages should follow the format of the examples:

Input	Output
Fish	Your word was Fish. The length of Fish-ishness is 12.
Hat	Your word was Hat. The length of Hat is 3.

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

Revision Fish-ishness

Your code from a previous question is copied below, along with the question prompt. We detected at least one of these things about your code:

- The code has an `else`
- The code has more than one `return`
- The code has a `print` statement

For correct functionality, there should be no `print` statement. For optimal style, there should be no `else`, and only one `return` statement. Please edit your code for functionality and/or style.

```
#{q://QID360/ChoiceTextEntryValue}
```

Write a function that takes a `String` as input and returns a `String`. The first and last lines are provided for you.

- For input `Strings` that end in "sh", concatenate the ending "-ishness" and return a message saying how long the new word is.
- For all other input `Strings`, return a message saying how long the given word is.
- Write the function so that it would **easy for someone else to modify** (e.g., if someone else wanted to look for `Strings` that end with different letters, or concatenate a different ending)

HINTS:

- `word.endsWith("sh")` will return `true` if the `String word` ends with "sh".
- The messages should be **returned**, not printed.
- You may use `+` to concatenate strings. For example:

```
String word2 = word1 + " there";  
If word1 is "hello", word2 is "hello there".
```

The messages should follow the format of the examples:

Input	Output
Fish	Your word was Fish. The length of Fish-ishness is 12.
Hat	Your word was Hat. The length of Hat is 3.

Copy and paste your code into the box below, and edit for style and/or functionality.

Check out these examples - these two code blocks do the same thing.

```
public String BBB1(String word) {  
    if (word.startsWith("b"))  
        return word + " is a fun word. BBB" + word + " has "  
            + (word.length() + 3) + " letters.";   
    else  
        return word + " is a fun word. " + word + " has " + word.length()  
            + " letters.";   
}
```

```
public String BBB2(String word) {  
    String secondWord = word;  
    if (word.startsWith("b"))  
        secondWord = "BBB" + word;  
    return word + " is a fun word. " + secondWord + " has "  
        + secondWord.length() + " letters.";  
}
```

Copy and paste your code into the box below, then follow the example to remove the second `return`, the `else`, and/or `print` statements, as applicable. If you notice any other errors, please fix them.

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

Demographics

Would you be interested in doing a follow-up interview about style and readability?

☐ Yes, Dr. Eliane Wiese may email me about participating in a follow-up interview. Optional: My preferred name is:

☐ No.

Please select the gender you identify with:

☐ Female

☐ Male

☐ Another gender:

☐ Prefer not to say

Please select your race/ethnicity (please check all that apply)

☐ American Indian or Alaskan Native

☐ African American or African

☐ Asian American or Asian

☐ Hispanic or Latino

☐ Native Hawaiian or Other Pacific Islander

☐ White

☐ Additional ethnicity/race:

☐ Prefer not to say

How old are you?

☐ Years:

☐ Prefer not to say

What year are you according to number of hours completed at the U?

- ☐ First year
- ☐ Sophomore
- ☐ Junior
- ☐ Senior
- ☐ Graduate student
- ☐ Other:
- ☐ Prefer not to say

Have you ever had a job or internship where one of your primary responsibilities was reading or modifying code?

- ☐ Yes
- ☐ No
- ☐ Prefer not to say

When and where did you first learn to code?

- ☐ In a class in school, prior to college
- ☐ Outside of any formal class, prior to college
- ☐ In an intro class in college
- ☐ Outside of any formal class, during college
- ☐ Other:

Select all Computer Science classes you are currently enrolled in:

- ☐ CS 1030: Foundations of CS
- ☐ CS 1410: Object Oriented Programming
- ☐ CS 2420: Algorithms/Data Structures
- ☐ CS 3500: Software Practice 1
- ☐ CS 3505: Software Practice 2
- ☐ CS3810: Computer Organization
- ☐ Other:
- ☐ Prefer not to say

For those computer science classes below that you have taken before, please enter the semester when you took it (enter more than one semester if you took the class more than once):

- ☐ CS 1030: Foundations of CS
- ☐ CS 1410: Object Oriented Programming
- ☐ CS 2420: Algorithms/Data Structures
- ☐ CS 3500: Software Practice 1
- ☐ CS 3505: Software Practice 2
- ☐ CS3810: Computer Organization
- ☐ Other:
- ☐ Prefer not to say

Select all computer science classes you plan to take NEXT semester:

- ☐ CS 1030: Foundations of CS
- ☐ CS 1410: Object Oriented Programming
- ☐ CS 2420: Algorithms/Data Structures
- ☐ CS 3500: Software Practice 1
- ☐ CS 3505: Software Practice 2
- ☐ CS3810: Computer Organization
- ☐ Other:
- ☐ Prefer not to say

What are your plans for your major?

- ☐ I declared my major in Computer Science
- ☐ I am not majoring in Computer Science. I declared my major in:
- ☐ I have not declared a major. I am planning to major in Computer Science.
- ☐ I have not declared a major. I am not planning to major in Computer Science. I am planning to major in:
- ☐ I'm firmly undecided about my major.
- ☐ Prefer not to say.
-

What are your plans for minoring in Computer Science?

- ☐ I am currently minoring in Computer Science (or plan to do so).
- ☐ I am not planning to minor in Computer Science.
- ☐ Prefer not to say.

Powered by Qualtrics