

Exercise 6 – Kaiden Pollesch

#1 Write a method `countToTen` that will use a for loop to count from 1 to 10, incrementing a local variable `ten` inside the loop, then return `ten`. Here is a buggy version of the method:

```
public int countToTen() {  
    int ten = 0;  
    for(int i = 0; i < 10; i +=2) {  
        ++ten;  
    }  
    return ten;  
}
```

Fixed Code from Me:

```
public int countToTen() {  
    int ten = 0;  
    for(int i = 0; i < 10; i++) {  
        ++ten;  
    }  
    return ten;  
}
```

Fixed code from ChatGPT:

```
public int countToTen() {  
    int ten = 0;  
    for (int i = 1; i <= 10; i++) {  
        ++ten;  
    }  
    return ten;  
}
```

Question: Did Chat-GPT give you the correct answer? If not, what is incorrect about it's response? Be specific, include code generated by Chat-GPT in your answer.

Yes it gave me the right answer.

#2 Write a method called `middleChar` that will get a `String` passed in as a parameter, get the number of characters in the `String`, and return the middle character of the `String`. If the `String` has an even number of letters, it should return the first of the two middle characters (i.e. for the word fluffy, the method will return 'u').

```
public char middleChar(String s) {  
    int length = s.length();  
    int middle = length / 2;  
    return s.substring(0, middle);  
}
```

Fixed Code from Me:

```
public char middleChar(String s) {  
    int length = s.length();  
    int middle = (int) length / 2;  
    if (length % 2 == 0) {  
        return s.charAt(middle - 1)  
    } else {  
        return s.charAt(middle);  
    }  
}
```

Fixed code from ChatGPT:

```
public char middleChar(String s) {  
    int length = s.length();  
    int middle = length / 2;  
  
    if (length % 2 == 0) {  
        // String has an even number of characters, return the first of the two  
        // middle characters  
        return s.charAt(middle - 1);  
    } else {  
        // String has an odd number of characters, return the middle character  
        return s.charAt(middle);  
    }  
}
```

Question: Did Chat-GPT give you the correct answer? If not, what is incorrect about it's response? Be specific, include code generated by Chat-GPT in your answer.

Yes it gave me the right code, the only difference is the type casting that I added.

#3 Write the equals method for the `String` class. Below is a buggy implementation.

```
public boolean equals(String s) {  
    if(this == s) { return true; }  
    if(!(s instanceof String)) { return false; }  
    for(int i = 0; i < s.length(); ++i) {  
        if(this.charAt(i) != s.charAt(i)) { return false; }  
    }  
    return true;  
}
```

Fixed Code from Me:

```
public boolean equals(String s) {  
    boolean flag = true;  
    for(int i = 0; i < s.length() && flag; i++) {  
        if(this.charAt(i) != s.charAt(i)) {  
            flag = false;  
        }  
    }  
    return flag;  
}
```

Fixed code from ChatGPT:

```
public boolean equals(Object obj) {  
    if (this == obj) {  
        return true;  
    }  
    if (obj == null || getClass() != obj.getClass()) {  
        return false;  
    }  
    String other = (String) obj;  
    // Cast the object to a String  
    if (this.length() != other.length()) {  
        return false;  
    }  
    for (int i = 0; i < this.length(); i++) {  
        if (this.charAt(i) != other.charAt(i)) {  
            return false;  
        }  
    }  
    return true;  
}
```

Question: Did Chat-GPT give you the correct answer? If not, what is incorrect about it's response? Be specific, include code generated by Chat-GPT in your answer.

So, they do the same thing but the code from ChatGPT checks the object before comparing the strings. Whereas I think that when creating the method, the input into the method will already check if it is a string or not. It will not allow you to enter something other than a string.

#4 Try and rewrite the prompts to explain to Chat-GPT what your code is attempting to do, then submit the new prompt with the same buggy code. Did Chat-GPT perform better? Worse? The same? Explain your answers with code examples.

#4.1- create a method called countToTen that will use a loop in order to count up from 1 to 10, returning the value of ten when the loop is no longer correct. Java

```
public class CountToTen {
    public static int countToTen() {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i); // Optional: Print the current count
            if (i == 10) {
                return 10;
            }
        }
        return -1; // This line should never be reached
    }
}
```

I believe that this preformed the same-ish because ChatGPT has more lines of code and added that wasn't necessary. But it preformed the same and got the same result for what we were looking for.

#4.2- create a method called middleCar that will find the length of a string and then find the middle of that string. then return the middle character. if the string does not have an odd amount of characters, it should return the left of the two characters. Java

```
public static char middleCar(String input) {
    int length = input.length();

    if (length % 2 == 0) {
        // If the string has an even number of characters, return the left of the two middle characters
        int middleIndex = length / 2 - 1;
        return input.charAt(middleIndex);
    } else {
        // If the string has an odd number of characters, return the middle character
        int middleIndex = length / 2;
        return input.charAt(middleIndex);
    }
}
```

This code does essentially the exact same thing, but instead of declaring the variable for the whole method it declares one for the if statement, and one for the else statement.

#4.3 - create a method with boolean return type that will check if one string is equal to another string. If it returns true, then they are equal. if it returns false, then they are not equal. Java

```
public static boolean areStringsEqual(String str1, String str2) {  
    // Use the equals method for case-sensitive comparison  
    // If you want case-insensitive comparison, use equalsIgnoreCase  
    return str1.equals(str2);  
}
```

Yes ChatGPT preformed much better because it cut down on the code by a bunch whereas my code was lengthy.