# CASE STUDY

Find what could be needed to get them correct. Feed the definition/relevant parts first in the chat for incorrect answers. e.g., error state and exception related questions

1. For 12 incorrect answers, what qualities are they? E.g., are they all mainly coding related? all fault/failure/error related?
    a. 1a. Lacks knowledge
    b. 1b. Wrong assumption
    c. 1c. Both
2. For one of the 12 incorrect answers, could you give ChatGPT more information and gotten a correct answer?
3. EPC/AIC is shown. Show example for EIC/[AC|APC]

Lacks knowledge | Intersection | Makes wrong assumption
8 | 2 | 4

## ChatGPT lacks the knowledge

- Treats executing the fault as failure
- Treats executing the fault as failure & error state as exception/crash
- **Treats error state as exception/crash**
- Treats crash/not executing as failure
- Knows what subsumption is but cannot make a logical decision

## ChatGPT makes the wrong assumption

- ChatGPT focused on NullPointerException (NPE), when the question did not care about NPE
    - **treats executing the fault as failure**
    - **treats executing the fault as failure & error state as exception/crash**
- ChatGPT thinks Treeset needs to be redefined, when in reality Hashset is the one that needs to be fixed. So, the error state it mentions is incorrect and opposite.
- ChatGPT gives a solution using inheritance, while the actual solution should use composition to preserve all three properties (symmetry, transitivity, and substitution principle)

## Characteristics of incorrect answers from ChatGPT:

We concluded from our study that shared context is better than separate context in terms of giving correct answers. From the 14 incorrect answers (AIC) in shared context for the second iteration for chapter 1 to 5, we can categorize ChatGPT's responses into two categories: ChatGPT lacks the knowledge and ChatGPT makes the wrong assumption.

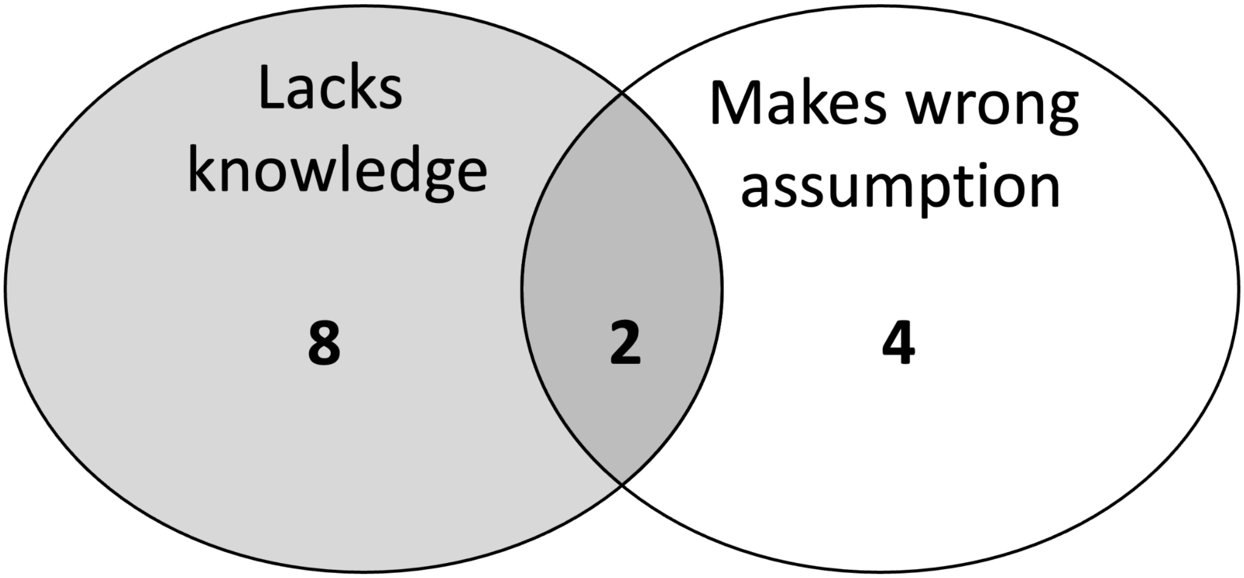

Figure 1: Venn diagram of characteristics of incorrect answers from ChatGPT

### ChatGPT lacks the knowledge:

This category includes instances where ChatGPT shows lack of correct knowledge to solve the problems given to it. For questions from chapters 1,2 and 3, ChatGPT mainly lacked understanding of fault, failure and error. It was sometimes confused with the differences between these definitions as well. For chapter 5, it seemed to lack understanding of how to conclude whether a test set satisfies a coverage criterion or not. Some of the mistakes made by ChatGPT because of lack of knowledge are-

- Treats executing the fault as failure
- Treats executing the fault as failure & error state as exception/crash
- **Treats error state as exception/crash**
- Treats crash/not executing as failure
- Knows what subsumption is but cannot make a logical decision

An example of lack of information is in Section~\ref{sec:background:example}. It seemed to know what coverage criterion and subsumption means but did not know how to make a logical final decision about test cases and subsumptions.

## ChatGPT makes the wrong assumption:

Sometimes, ChatGPT focuses on a different part of the problem and gives the wrong solution by making incorrect assumptions. Some of the incorrect assumptions made by ChatGPT are-

- ChatGPT focused on NullPointerException (NPE), when the question did not care about NPE
- ChatGPT thinks Treeset needs to be redefined, when in reality Hashset is the one that needs to be fixed. So, the error state it mentions is incorrect and opposite.
- ChatGPT gives a solution using inheritance, while the actual solution should use composition to preserve all three properties (symmetry, transitivity, and substitution principle)

## Both:

There is some overlap in the two categories as well. ChatGPT seems to have both a lack of knowledge and makes wrong assumptions for two of the questions. It first made a wrong assumption and focused on a different fault, then also lacked knowledge of failure and error state in the subsequent questions. The overlap in category of incorrect response in those two questions were that-

- ChatGPT focused on NullPointerException (NPE), when the question did not care about NPE
    - **treats executing the fault as failure (1.5.4.b)**
    - **treats executing the fault as failure & error state as exception/crash(1.5.4.c)**

Since the sub-questions are related to each other, if the first sub-question is wrong(i.e., incorrectly detected fault), the subsequent questions will also be answered incorrectly in a shared state as the context is common. So, not only did it focus on a wrong fault and kept working on it in the subquestions, it also treated failure and error in a wrong way for lack of understanding.

What is lack of knowledge (first example subsumption), wrong assumption (odd or positive) & intersection

## Analysis:

We can conclude from Figure 1 that eight out of the fourteen incorrect answers are related to ChatGPT lacking knowledge, two of the remaining six questions overlap with lack of knowledge as well. ChatGPT's lack of knowledge in chapter 1 to 5 are mainly related to fault, failure and error. Errors are incorrect internal state that is caused by the fault; ChatGPT is not able to correctly determine what an error is and find the first error state in a code. It treats errors as throwing exceptions or crashing the code. Also, treating execution of fault as failure is also another common reason for incorrect answers. Faults are defective lines of code and failures are observable incorrect behavior, but ChatGPT seems to think executing the faulty line of code and showing incorrect behavior is the same. For the remaining four questions, ChatGPT seems to make wrong assumptions and give an incorrect solution.

What is lack of knowledge (first example subsumption), wrong assumption (odd or positive) & intersection.

## Prompt Modifications

We further experimented with one of the 14 incorrect answers. Upon making some changes to the prompt given to ChatGPT, the correct answer to one of them could be obtained by it. As ChatGPT performs better in a shared context, we fed it the entire question, including the common context and all subquestions. The code is given below, the function OddorPos() is about finding odd or positive numbers and there is a test case commented out as well on which part of the question was asked. We chose the subquestion (f) which is "Implement your repair and verify that the given test now produces the expected output. Submit a screen printout or other evidence that your new program works." ChatGPT needs to find a repair to the fault in the code and show that the test case gives the correct answer for the repaired program. The original response from ChatGPT to this sub-question was to add a null check before the for loop to ensure that the input array is not null. But the actual fault is that it does not take account of negative odd values.
So, we tried to modify the prompt a little and see if the correct repair could be obtained by ChatGPT. We tried two different modifications:

### Modification 1:

The entire modified prompt fed to ChatGPT is given below.

Answer the following questions about each program.

```
/**
  * Count odd or positive elements in an array
  *
  * @param x array to search
  * @return count of odd or positive elements in x
  * @throws NullPointerException if x is null
  */
  public static int oddOrPos (int[] x)
  {  // Effects:  if x is null throw NullPointerException
    // else return the number of elements in x that
    //      are either odd or positive (or both)
    int count = 0;

    for (int i = 0; i < x.length; i++)
    {
      if (x[i]%2 == 1 || x[i] > 0)
      {
        count++;
      }
    }
    return count;
  }
```

// test: x = [-3, -2, 0, 1, 4]; *Expected = 3 because the odd number is -3 and the positive numbers are 1,4.*

(a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.
(b) If possible, give a test case that does not execute the fault. If not, briefly explain why not.
(c) If possible, give a test case that executes the fault, but does not result in an error state. If not, briefly explain why not.
(d) If possible give a test case that results in an error state, but not a failure. Hint: Don't forget about the program counter. If not, briefly explain why not.
(e) For the given test case, describe the first error state. Be sure to describe the complete state.
(f) *The answer does not involve having a null check and zero is not a positive number.* Implement your repair and verify that the given test now produces the expected output. Submit a screen printout or other evidence that your new program works."

The changes were made to the subquestion (f), the original prompt did not mention that the answer does not involve a null check and that zero is not a positive number. The expected output of the test case was also described in more detail. These were the extra information given to ChatGPT to get the correct repair. The following is where we modified the prompt: "// test: x = [-3, -2, 0, 1, 4]; Expected = 3 because the odd number is -3 and the positive numbers are 1,4." and "The answer does not involve having a null check and zero is not a positive number. "

## ChatGPT's response:

"(a) The problem with the given code is that it is counting zero as a positive number, and not considering negative odd numbers as odd. A modification to the code would be to change the if statement to check if the number is odd (x[i] % 2 != 0) or positive (x[i] > 0) instead of odd or positive.

(b) A test case that does not execute the fault would be x = [1, 2, 3, 4], expected output = 3.

(c) A test case that executes the fault but does not result in an error state would be x = [0, 2, 4, 6], expected output = 1 (the program would count zero as a positive number, but the output would still be correct)

(d) A test case that results in an error state but not a failure would be passing a null value for x, which would throw a NullPointerException, but the program would not crash.

(e) For the given test case, the first error state is that the program counts zero as a positive number, resulting in an unexpected output of 4 instead of 3.

(f) To repair the code, we can change the if statement to check if the number is odd (x[i] % 2 != 0) or positive (x[i] > 0) instead of odd or positive.
And also add a null check to check if the passed array is null or not if it is then we can throw an exception.

```
public static int oddOrPos (int[] x)
{  // Effects:  if x is null throw NullPointerException
   // else return the number of elements in x that
   //      are either odd or positive (or both)
   if(x==null)
    throw new NullPointerException();
   int count = 0;
   for (int i = 0; i < x.length; i++)
```

```
    {
      if (x[i] % 2 != 0 || x[i] > 0)
      {
        count++;
      }
    }
    return count;
  }
```

The modified code will return the expected output of 3 for the test case x = [-3, -2, 0, 1, 4].”

The textbook solution is modifying the if condition to: if (x[i]%2 == -1 || x[i] > 0), the positive odd numbers are taken care of by the second test.
Here, ChatGPT’s response is correct as well given that it is checking for negative odd numbers as well in the condition x[i] % 2 != 0. Although the solution adds an extra null check at the beginning even after mentioning the solution does not involve null check, still the modification to the condition is good enough to pass the test case.

## Modification 2:

The second modification is the same as before for the other sub-questions, only the sub-question (f) is changed differently. The test case is also changed as modification 1.

“// test: x = [-3, -2, 0, 1, 4]; *Expected = 3 because the odd number is -3 and the positive numbers are 1,4.*

(f) *The fault in the above code is that the if condition needs to take account of negative odd values (positive odd numbers are taken care of by the second test)*
Implement your repair and verify that the given test now produces the expected output. Submit a screen printout or other evidence that your new program works.”

The changes were made to the subquestion (f), the original prompt did not explicitly mention the fault. The changes we made included explicitly mentioning the fault in the prompt, so that ChatGPT can give the correct repair. The following is what we added to the prompt: “The fault in the above code is that the if condition needs to take account of negative odd values (positive odd numbers are taken care of by the second test)”. The expected output of the test case was also described in more detail. The following is where we modified the prompt: “// test: x = [-3, -2, 0, 1, 4]; Expected = 3 because the odd number is -3 ”

## ChatGPT’s response:

“(a) The code doesn't take into account negative odd numbers. It only counts positive numbers or positive odd numbers.
A modification to the code would be to include negative odd numbers by adding an additional condition to the if statement:

```
if (x[i]%2 == 1 || x[i] > 0 || (x[i]%2 == -1 && x[i] < 0))
```

(b) A test case that does not execute the fault would be:

```
int[] x = {2, 4, 6};
oddOrPos(x);  // Output: 0
```

(c) There is no test case that executes the fault but does not result in an error state, as the error state is caused by a lack of consideration for negative odd numbers.

(d) There is no test case that results in an error state but not a failure, as the error state (NullPointerException) is thrown if the input array is null.

(e) For the given test case, the first error state is the expected output not matching the actual output.

(f) The corrected code is as follows:

```
public static int oddOrPos (int[] x)
{
  if (x == null) {
    throw new NullPointerException("Array is null");
  }
  int count = 0;
  for (int i = 0; i < x.length; i++)
  {
    if (x[i]%2 == 1 || x[i] > 0 || (x[i]%2 == -1 && x[i] < 0))
    {
      count++;
    }
  }
  return count;
}
```

This code can be tested with the given test case and produces the expected output:

```
int[] x = {-3, -2, 0, 1, 4};
```

```
System.out.println(oddOrPos(x));  // Output: 3”
```

The textbook solution is modifying the if condition to: if (x[i]%2 == -1 || x[i] > 0), the positive odd numbers are taken care of by the second test.
Here, ChatGPT’s response is correct as well given that it is checking for negative odd numbers in the later part of the if condition: x[i]%2 == -1 && x[i] < 0. Although the book solution is shorter and enough to identify negative odd numbers, ChatGPT’s answer is correct as well, but extraneous (the first and fourth predicate of the if condition). Also, the solution adds an extra null check at the beginning again.

## Example of AC-EIC:

Below is an example of a question whose explanation given by ChatGPT was incorrect, but the answer was correct. The example was run as a shared context but to discuss an example of AC-EIC category, just the relevant part of the question is shown below.

```
“
/**
  * Count odd or positive elements in an array
  *
  * @param x array to search
  * @return count of odd or positive elements in x
  * @throws NullPointerException if x is null
  */
  public static int oddOrPos (int[] x)
  {  // Effects:  if x is null throw NullPointerException
        // else return the number of elements in x that
        //          are either odd or positive (or both)
        int count = 0;

        for (int i = 0; i < x.length; i++)
        {
        if (x[i]%2 == 1 || x[i] > 0)
        {
        count++;
        }
        }
        return count;
  }
// test: x = [-3, -2, 0, 1, 4]; Expected = 3
```

If possible give a test case that results in an error, but not a failure. If not, briefly explain why not. Hint: Don't forget about the program counter."

## ChatGPT's response:

"It is not possible to give a test case that results in an error, but not a failure because when the code throws a NullPointerException, it results in an error and the program will stop execution. An error state refers to an unexpected behavior of the program and in this case, it is the program throwing an exception and stopping execution, which is a failure. The program counter will not continue execution past the point where the exception is thrown.

In this case, the program will stop execution when the exception is thrown, hence it is not possible to have a test case that results in an error but not a failure."

The answer is correct because it is not possible to give a test case that results in an error but not a failure. But the explanation is not what is expected. The problem is that ChatGPT gives the wrong fault at the beginning, it thinks the fault is that there is no null check implemented in the code, hence it throws a NullPointerException if the input array x is null. Whereas, the actual fault in this code is that the if condition needs to take account of negative odd values. Thus, it gives an incorrect explanation of the answer. The error state and failure both are related to the fault of this code.

## Correct answer from textbook:

"For this particular program, every input that results in error also results in failure. The reason is that error states are not repairable by subsequent processing. If there is a negative value in x, all subsequent states (after processing the negative value) will be error states no matter what else is in x."

For grading this type of response, where the answer itself is correct but the explanation is wrong, correctness of explanation should take precedence over the answer's correctness as the answer could be random and guessed by the student as it is a yes/no question. So, the explanation is what matters as that would reveal if the student has the correct idea of the solution. So, it is more important for ChatGPT to get an explanation correct.