
Lab 5: ChatGPT in Code Generation and Debugging

Objective:

Explore how ChatGPT can assist in generating and debugging code in languages like Python, C++, or Java.

Activity 1: Code Generation (Python)

Prompt Given to ChatGPT:

Write a Python function to check whether a number is prime.

ChatGPT's Output:

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

Test Cases & Expected Output:

```
print(is_prime(2)) # True
print(is_prime(9)) # False
print(is_prime(17)) # True
```

Comparison:

- ✓ Output is correct for all test cases
 - ✓ Logic is efficient (uses \sqrt{n} for optimization)
 - ✓ Clean, readable code
-

Activity 2: Code Generation (C++)

Prompt Given to ChatGPT:

Write a C++ program to reverse a string.

ChatGPT's Output:

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    string str = "hello";
    reverse(str.begin(), str.end());
    cout << "Reversed string: " << str << endl;
    return 0;
}
```

Expected Output:

Reversed string: olleh

Comparison:

- ✓ Output is correct
 - ✓ Uses STL `reverse` for simplicity
 - ✓ Efficient and concise solution
-

Activity 3: Code Debugging (Python)

Given Code (Buggy):

```
def factorial(n):
    if n == 0:
        return 0
    else:
        return n * factorial(n - 1)
```

Bug:

The base case should return `1`, not `0`. Returning `0` always gives `0` as the final result.

Prompt Given to ChatGPT:

Debug this factorial function and explain the issue.

ChatGPT's Output:

```
def factorial(n):
```

```
if n == 0:  
    return 1 # Fixed: base case for factorial should return 1  
else:  
    return n * factorial(n - 1)
```

Explanation Provided:

The base case `factorial(0)` should return `1`, because by definition, $0! = 1$.
Returning 0 would cause all recursive multiplications to result in 0.

Test Cases:

```
print(factorial(0)) # 1  
print(factorial(5)) # 120
```

Comparison:

- ✓ Bug identified and fixed
 - ✓ Clear explanation
 - ✓ Correct output on test cases
-

Conclusion:

ChatGPT is highly effective in both **code generation** and **debugging**. It:

- Writes syntactically correct and optimized code
- Identifies logical and runtime errors in buggy snippets
- Explains bugs clearly for student learning

This makes ChatGPT a powerful support tool in programming education and real-world development.
