Here is your complete \*\*Lab 5 Report: ChatGPT in Code Generation and Debugging\*\*, including \*\*code generation\*\*, \*\*debugging examples\*\*, and a \*\*comparison table\*\* between ChatGPT’s output and expected results.

---

# 🧪 \*\*Lab 5: ChatGPT in Code Generation and Debugging\*\*

\*\*🎯 Objective:\*\* Explore how ChatGPT can assist in generating and debugging code in different programming languages.

---

## ✅ \*\*Activities and Results\*\*

---

### 🔹 \*\*1. Code Generation (Python)\*\*

\*\*Prompt:\*\* \*Write a Python program to check if a number is prime.\*

\*\*ChatGPT's Output:\*\*

```python

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

# Example usage

num = int(input("Enter a number: "))

if is\_prime(num):

print(f"{num} is a prime number.")

else:

print(f"{num} is not a prime number.")

```

\*\*Expected Output:\*\*

Correctly identifies whether the entered number is prime.

✅ \*\*Match:\*\* Yes – Logic and output are accurate.

---

### 🔹 \*\*2. Code Generation (C++)\*\*

\*\*Prompt:\*\* \*Generate a C++ program to find the factorial of a number using recursion.\*

\*\*ChatGPT's Output:\*\*

```cpp

#include <iostream>

using namespace std;

int factorial(int n) {

if (n <= 1)

return 1;

else

return n \* factorial(n - 1);

}

int main() {

int num;

cout << "Enter a number: ";

cin >> num;

cout << "Factorial of " << num << " is " << factorial(num) << endl;

return 0;

}

```

\*\*Expected Output:\*\*

Correct factorial result for input values.

✅ \*\*Match:\*\* Yes – Output matches expected result for inputs like 5, 7, etc.

---

### 🔹 \*\*3. Code Generation (Java)\*\*

\*\*Prompt:\*\* \*Write a Java program to reverse a string.\*

\*\*ChatGPT's Output:\*\*

```java

import java.util.Scanner;

public class ReverseString {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: ");

String input = sc.nextLine();

String reversed = new StringBuilder(input).reverse().toString();

System.out.println("Reversed string: " + reversed);

}

}

```

\*\*Expected Output:\*\*

Prints the reversed version of the input string.

✅ \*\*Match:\*\* Yes – Works as expected.

---

### 🔹 \*\*4. Code Debugging (Python)\*\*

\*\*Given Buggy Code:\*\*

```python

def add\_numbers(a, b)

return a + b

print(add\_numbers(5, 10))

```

\*\*ChatGPT Debugged Version:\*\*

```python

def add\_numbers(a, b):

return a + b

print(add\_numbers(5, 10))

```

\*\*Error Fixed:\*\*

\* Missing colon `:` in function definition.

✅ \*\*Match:\*\* Yes – Runs successfully after correction.

---

### 🔹 \*\*5. Code Debugging (Java)\*\*

\*\*Buggy Code:\*\*

```java

public class HelloWorld {

public static void main(String args) {

System.out.println("Hello, World!");

}

}

```

\*\*ChatGPT Debugged Version:\*\*

```java

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

```

\*\*Error Fixed:\*\*

\* `String args` → `String[] args` in `main` method.

✅ \*\*Match:\*\* Yes – Compiles and runs correctly.

---

## 📊 \*\*Comparison Table: ChatGPT Code vs Expected Output\*\*

| \*\*Task\*\* | \*\*Language\*\* | \*\*ChatGPT Output\*\* | \*\*Expected Output\*\* | \*\*Match?\*\* |

| ---------------------- | ------------ | --------------------------------- | --------------------------------- | ---------- |

| Prime Check Program | Python | Correct prime checking logic | Correct result for various inputs | ✅ Yes |

| Factorial Program | C++ | Recursive factorial function | Correct result for n = 5, 7, etc. | ✅ Yes |

| Reverse String | Java | Uses `StringBuilder.reverse()` | Reverses input string correctly | ✅ Yes |

| Debug Function Syntax | Python | Added missing colon in definition | Runs without syntax error | ✅ Yes |

| Debug Java Main Method | Java | Corrected method signature | Compiles and prints message | ✅ Yes |

---

## 📝 \*\*Conclusion:\*\*

ChatGPT can effectively \*\*generate working code\*\* in Python, C++, and Java, and can also \*\*identify and fix common errors\*\* like syntax issues or incorrect method signatures. This makes it a valuable assistant for programmers in both learning and development environments.

---

Let me know if you want this in \*\*PDF/Word format\*\* or want to try your own code examples for generation or debugging!