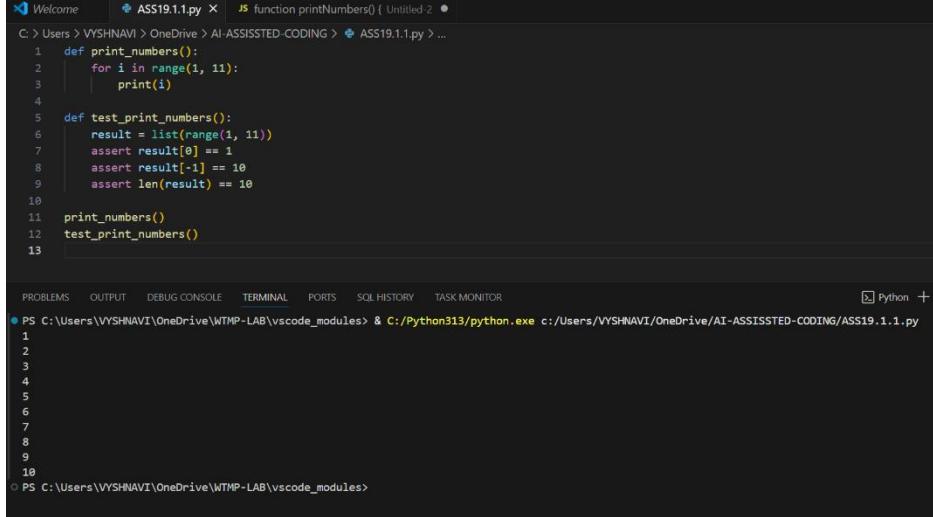


SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s) Name		Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)	
Course Code	24CS002PC215	Course Title	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week10 - Thursday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 19.4(Present assignment number)/24(Total number of assignments)			
Q.No.	Question		Expected Time to complete
1	Lab 19 – Code Translation: Converting Between Programming Languages Lab Objectives: <ul style="list-style-type: none"> Understand how AI tools can assist in translating code between different programming languages. 		Week10 - Thursday

	<ul style="list-style-type: none"> • Learn to verify correctness and functionality after translation. • Explore syntactic and semantic differences between languages (e.g., Python, Java, C++). • Practice debugging and optimizing AI-translated code. 	
	<p>Task 1: Translate a Simple Program (Python → JavaScript)</p> <ul style="list-style-type: none"> • Instructions: <ul style="list-style-type: none"> • Write a Python function <code>print_numbers()</code> that prints the first 10 natural numbers using a loop. • Translate the function into JavaScript as a reusable function <code>printNumbers()</code>. • Call the function in both languages to display results. • Expected Output: <ul style="list-style-type: none"> • 1 • 2 • 3 • ...10 <p>PROMPT: Write a Python function <code>print_numbers()</code> that prints the first 10 natural numbers using a loop. Translate it into JavaScript as <code>printNumbers()</code> and call both.</p> <p>CODE:</p>  <pre> Welcome ASS19.1.1.py X JS function printNumbers() (Untitled-2 • C: > Users > VYSHNAVI > OneDrive > AI-ASSISTED-CODING > ASS19.1.1.py > ... 1 def print_numbers(): 2 for i in range(1, 11): 3 print(i) 4 5 def test_print_numbers(): 6 result = list(range(1, 11)) 7 assert result[0] == 1 8 assert result[-1] == 10 9 assert len(result) == 10 10 11 print_numbers() 12 test_print_numbers() 13 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules> & C:/Python313/python.exe c:/Users/VYSHNAVI/OneDrive/AI-ASSISTED-CODING/ASS19.1.1.py 1 2 3 4 5 6 7 8 9 10 PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules> </pre>	

The screenshot shows a VS Code interface. The top bar has tabs for 'Welcome', 'ASS19.1.1.py', 'JS ASS19.1.2.js X', and 'function printNumbers() { Untitled-2'. The main editor area contains the following JavaScript code:

```
1 function printNumbers() {
2     for (let i = 1; i <= 10; i++) {
3         console.log(i);
4     }
5 }
6
7 printNumbers();
8
```

Below the editor is a navigation bar with links: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), PORTS, SQL HISTORY, and TASK MONITOR. The terminal window shows the following output:

```
● PS C:\Users\VYSHNAVI\OneDrive\WTEMP-LAB\vscode_modules> & C:/Python313/python.exe c:/Users/VYSHNAVI/OneDrive/WTEMP-LAB/vscode_modules>
1
2
3
4
5
6
7
8
9
10
```

Task 2: Convert Conditional Statements (Java → Python)

- **Instructions:**

- Write a Java method `checkNumber(int num)` that checks if a number is positive, negative, or zero.
- Translate the method into a Python function `check_number(num)`.
- Call the function/method with different inputs and compare outputs.

- **Expected Output:**

- Input: -5 → Output: The number is negative
- Input: 0 → Output: The number is zero
- Input: 7 → Output: The number is positive

PROMPT:

Write a Java method `checkNumber(int num)` to check if a number is positive, negative, or zero. Translate to Python and test with -5, 0, and 7.

CODE:

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a file tree with the current file being `ASS19.2.1.py`.
- Code Editor:** Displays the following Python code:

```
1 def check_number(num):
2     if num < 0:
3         return "negative"
4     elif num == 0:
5         return "zero"
6     else:
7         return "positive"
8
9
10 # Test cases
11 print(check_number(-5)) # negative
12 print(check_number(0)) # zero
13 print(check_number(7)) # positive
14
```
- Terminal:** Shows the output of running the script:

```
PS C:\Users\VYSHNAVI\OneDrive\WTMP-LAB\vscode_modules> & 'c:\Python313\python.exe' -m debugpy -2025.14.1-win32-x64\bundled\libs\debugpy\launcher' '50568' '--'
negative
zero
positive
PS C:\Users\VYSHNAVI\OneDrive\WTMP-LAB\vscode_modules>
```
- Bottom Navigation:** Includes links for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), PORTS, SQL HISTORY, and TASK MONITOR.

The screenshot shows a VS Code interface with the following details:

- Editor:** The main editor tab is titled "ASS19.2.1.py" (Python), but the code displayed is Java. The code defines a class `NumberCheck` with a `checkNumber` method that prints whether a number is positive, negative, or zero. It also contains a `main` method that calls `checkNumber` for -5, 0, and 7.
- Terminal:** The terminal tab is active and shows the output of running the Python script. The output is:

```
PS C:\Users\VYSHNAVI\OneDrive\WTEMP-LAB\vscode_modules> & 'c:\Python39\python.exe' ASS19.2.1.py
negative
zero
positive
```
- Bottom Navigation:** The bottom navigation bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS, SQL HISTORY, and TASKS.

Task 3: Translate Recursive Function (Python → C++)

- Instructions:**
 - Write a Python function `factorial(n)` that calculates factorial of a number using recursion.
 - Translate the same into a C++ function `int factorial(int n)`.
 - Call the function in both languages with inputs 5 and 0.
- Expected Output:**
 - Input: 5 → Output: Factorial = 120**
 - Input: 0 → Output: Factorial = 1**

PROMPT:

Write a Python function factorial(n) that calculates factorial using recursion. Translate to C++ and test with 5 and 0.

CODE:

The screenshot shows a VS Code interface. The top part displays a Python file named 'ASS19.3.1.py' with the following code:

```
C: > Users > VYSHNAVI > OneDrive > AI-ASSISTED-CODING > ASS19.3.1.py > ...
1 ✓ def factorial(n):
2     # Base case: factorial of 0 is 1
3     if n == 0:
4         return 1
5     # Recursive case
6     else:
7         return n * factorial(n - 1)
8
9 print("Factorial of 5:", factorial(5))
10 print("Factorial of 0:", factorial(0))
11
```

The bottom part shows the terminal tab with the output of running the script:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR Python Debug Console
● PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules> & 'c:\Python313\python.exe' 'c:\Users\VYSHNAVI\OneDrive\AI-ASSISTED-CODING\ASS19.3.1.py'
Factorial of 5: 120
Factorial of 0: 1
○ PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules>
```

The screenshot shows a VS Code interface with two tabs open: 'ASS19.3.1.py' and 'ASS19.3.2.cpp'. The 'ASS19.3.2.cpp' tab contains the following C++ code:

```
C: > Users > VYSHNAVI > OneDrive > AI-ASSISTED-CODING > ASS19.3.2.cpp
1 #include <iostream>
2 using namespace std;
3
4 int factorial(int n) {
5     if (n == 0)
6         return 1;
7     else
8         return n * factorial(n - 1);
9 }
10
11 int main() {
12     cout << "Factorial of 5: " << factorial(5) << endl;
13     cout << "Factorial of 0: " << factorial(0) << endl;
14     return 0;
15 }
16
```

The terminal below shows the execution of the code and its output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR Python D
● PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules> & 'c:\Python313\python.exe' 'c:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules\ASS19.3.1.py'
Factorial of 5: 120
Factorial of 0: 1
○ PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules>
```

Task 4: Data Structures with Functions (JavaScript → Python)

- **Instructions:**

- Write a JavaScript function `printStudents(students)` that takes an array of student names and prints each name.
- Translate it into a Python function `print_students(students)` using a list.
- Test both functions with sample student names.

- **Expected Output:**

- Student List:
- Alice
- Bob
- Charlie

PROMPT:

Write a JavaScript function `printStudents(students)` that prints each student name. Translate to Python and test with `["Alice", "Bob", "Charlie"]`.

CODE:

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. On the left, there's a large empty workspace area. In the center, there's a dark-themed code editor window. At the top of the editor, it says "Welcome" and "ASS19.4.1.py X". Below that, the file path is "C: > Users > VYSHNAVI > OneDrive > AI-ASSISTED-CODING > ASS19.4.1.py > ...". The code itself is:

```
1 def print_students(students):
2     # Print each student name
3     print("Student List:")
4     for student in students:
5         print(student)
6
7 print_students(["Alice", "Bob", "Charlie"])
8
```

Below the code editor, there's a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, SQL HISTORY, and TASK MONITOR. The TERMINAL tab is currently selected, indicated by a blue underline. In the terminal pane, the output of the Python script is shown:

```
● PS C:\Users\VYSHNAVI\OneDrive\WTEMP-LAB\vscode_modules> & C:/Python313/python.exe c:/Users/VYS
/AI-ASSISTED-CODING/ASS19.4.1.py
Student List:
Alice
Bob
Charlie
○ PS C:\Users\VYSHNAVI\OneDrive\WTEMP-LAB\vscode_modules>
```

The screenshot shows a VS Code interface with two tabs open: `ASS19.4.1.py` and `ASS19.4.2.js`. The `ASS19.4.1.py` file contains the following Python code:

```
1 function printStudents(students) {
2     console.log("Student List:");
3     for (let i = 0; i < students.length; i++) {
4         console.log(students[i]);
5     }
6 }
7
8 printStudents(["Alice", "Bob", "Charlie"]);
```

The `ASS19.4.2.js` tab is currently active. Below the code editor, the terminal window displays the output of the Python script:

```
PS C:\Users\vyshnavi\OneDrive\AI-ASSISTED-CODING> & C:/Python313/python.exe c:/Users/VYSHNAVI/OneDrive/AI-ASSISTED-CODING/ASS19.4.1.py
Student List:
Alice
Bob
Charlie
PS C:\Users\vyshnavi\OneDrive\WTP-LAB\vscode_modules>
```

Task 5: Class & Object Translation (Python → Java)

- **Instructions:**

1. Write a **Python class** `Car` with attributes: brand, model, year.
2. Add a **method** `display_details()` that prints car details.
3. Translate the same into a **Java class** `Car` with attributes and a **method** `displayDetails()`.
4. Create an object in both languages and call the method.

- **Expected Output:**

- Car Details:
- Brand: Toyota
- Model: Corolla
- Year: 2020

PROMPT:

Write a Python class `Car` with attributes: brand, model, year. Add method `display_details()`. Translate to Java and test with Toyota Corolla 2020.

CODE:

```
C: > Users > VYSHNAVI > OneDrive > AI-ASSISTED-CODING > ASS19.5.1.py > ...
1  class Car:
2      def __init__(self, brand, model, year):
3          # Initialize car attributes
4          self.brand = brand
5          self.model = model
6          self.year = year
7
8      def display_details(self):
9          # Return formatted car details
10         print("Car Details:")
11         print("Brand:", self.brand)
12         print("Model:", self.model)
13         print("Year:", self.year)
14
15     car = Car("Toyota", "Corolla", 2020)
16     car.display_details()
17

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    SQL HISTORY    TASK MONITOR    Python Debug Console    + v

PS C:\Users\VYSHNAVI\OneDrive\WTMP-LAB\vscode_modules> & 'c:\Python313\python.exe' 'c:\Users\VYSHNAVI\vscode\extensions\ms-python.debugpy-2025.14.1-win32-x64\bundled\libs\debugpy\launcher' '58138' '--' 'C:\Users\VYSHNAVI\OneDrive\AI-ASSISTED-CODING\ASS19.5.1.py'
PS C:\Users\VYSHNAVI\OneDrive\WTMP-LAB\vscode_modules> c;; cd 'c:\Users\VYSHNAVI\OneDrive\WTMP-LAB\vscode\extensions\ms-python.debugpy-2025.14.1-win32-x64\bundled\launcher' '58137' '--' 'C:\Users\VYSHNAVI\OneDrive\AI-ASSISTED-CODING\ASS19.5.1.py'
Car Details:
Brand: Toyota
Model: Corolla
Year: 2020
PS C:\Users\VYSHNAVI\OneDrive\WTMP-LAB\vscode_modules>
```

The screenshot shows a VS Code interface with two tabs open: 'ASS19.5.1.py' and 'ASS19.5.2.java'. The 'ASS19.5.2.java' tab contains the following Java code:

```
C: > Users > VYSHNAVI > OneDrive > AI-ASSISTED-CODING > ASS19.5.2.java
1  public class Car {
2      public Car(String brand, String model, int year) {
3          ...
4      }
5
6      public void displayDetails() {
7          System.out.println("Car Details:");
8          System.out.println("Brand: " + brand);
9          System.out.println("Model: " + model);
10         System.out.println("Year: " + year);
11     }
12
13     public static void main(String[] args) {
14         Car car = new Car("Toyota", "Corolla", 2020);
15         car.displayDetails();
16     }
17 }
18
19 }
```

The 'TERMINAL' tab shows the execution of the Java code and its output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR Python D
● PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules> & 'c:\Python313\python.exe' 'c:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules\ms-python.debugpy-2025.14.1-win32-x64\bundled\libs\debugpy\launcher' '58138' '--' 'C:\Users\VYSHNAVI\OneDrive\AI-ASSISTED-CODING\ASS19.5.1.py'
● PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules> c:; cd 'c:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules\ms-python.debugpy-2025.14.1-win32-x64\bundled\libs\debugpy\launcher' '58138' '--' 'C:\Users\VYSHNAVI\OneDrive\AI-ASSISTED-CODING\ASS19.5.1.py'
Car Details:
Brand: Toyota
Model: Corolla
Year: 2020
○ PS C:\Users\VYSHNAVI\OneDrive\WTP-LAB\vscode_modules>
```

✓ Deliverables (For All Tasks)

1. AI-generated prompts for code and test case generation.
2. At least 3 assert test cases for each task.
3. AI-generated initial code and execution screenshots.
4. Analysis of whether code passes all tests.
5. Improved final version with inline comments and explanation.
6. Compiled report (Word/PDF) with prompts, test cases, assertions, code, and output.