SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
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:** • Understand how AI tools can assist in translating code between different programming languages.	Week10 - Thursday

- 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.

Task 1: Translate a Simple Program (Python → JavaScript)

- **Instructions:**
  - Write a Python function print_numbers() that prints the first 10 natural numbers using a loop.
  - Translate the function into JavaScript as a reusable function printNumbers().
  - Call the function in both languages to display results.
- **Expected Output:**
- **1**
- **2**
- **3**
- **...10**

Prompt:

Write a program to Print first 10 natural numbers in both Python and JavaScript

Code:

```python
# Function to print first 10 natural numbers
def print_numbers():
    for i in range(1, 11):
        print(i)

# Call the function
print_numbers()
```

```
1
2
3
4
5
6
7
8
9
10
```

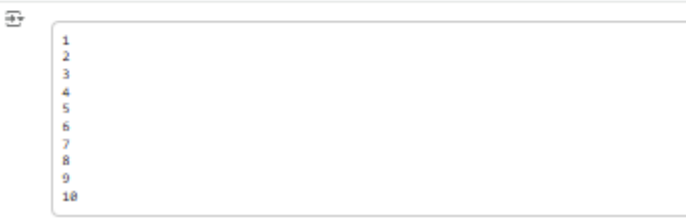
JS CODE:

```javascript
%%javascript
// Create a <div> element to display the output
let outputDiv = document.createElement('div');
outputDiv.style.fontFamily = 'monospace';
outputDiv.style.margin = '10px';
outputDiv.style.padding = '10px';
outputDiv.style.border = '1px solid #ccc';
outputDiv.style.borderRadius = '8px';

// Function to print numbers
function printNumbers() {
  let output = "";
  for (let i = 1; i <= 10; i++) {
    output += i + "<br>";
  }
  outputDiv.innerHTML = output;
}

// Add the outputDiv to the notebook cell
element.append(outputDiv);

// Call the function
printNumbers();
```



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 programs in Java and Python to check if a number is positive, negative, or zero using a separate function or method (checkNumber / check_number), but do it in another way **JS CODE:**

```
%%writefile Main.java
import java.util.Scanner;

// Java program to check if a number is positive, negative, or zero
public class Main {
    static void checkNumber(int num) {
        if (num > 0) {
            System.out.println("The number is positive");
        } else if (num < 0) {
            System.out.println("The number is negative");
        } else {
            System.out.println("The number is zero");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        checkNumber(num);
        sc.close();
    }
}
```

```
Overwriting Main.java
```

```
!javac Main.java
```

```
!java Main
```

```
Enter a number: 12
The number is positive
```

PYTHON CODE

```
# Python program to check if a number is positive, negative, or zero
def check_number(num):
    if num > 0:
        print("The number is positive")
    elif num < 0:
        print("The number is negative")
    else:
        print("The number is zero")

# Test cases
check_number(-5)
check_number(0)
check_number(7)
```

```
The number is negative
The number is zero
The number is positive
```

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 programs in Python and C++ to calculate the factorial of a number using recursion.Define a function (`factorial`).

PYTHON CODE:

```python
# Python program to find factorial using recursion
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

# Calling the function
print("Input: 5 → Output: Factorial =", factorial(5))
print("Input: 0 → Output: Factorial =", factorial(0))
```

```
Input: 5 → Output: Factorial = 120
Input: 0 → Output: Factorial = 1
```

.**C++ CODE:**

```
%%writefile factorial.cpp
#include <iostream>
using namespace std;

// Function to find factorial using recursion
int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {
    cout << "Input: 5 → Output: Factorial = " << factorial(5) << endl;
    cout << "Input: 0 → Output: Factorial = " << factorial(0) << endl;
    return 0;
}
```

```
Writing factorial.cpp
```

```
!g++ factorial.cpp -o factorial
```

```
!./factorial
```

```
Input: 5 → Output: Factorial = 120
Input: 0 → Output: Factorial = 1
```

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 takes an array of student names and prints each name. Translate it into a Python function print_students(students) using a list.

JSCODE:

```
%%javascript
// JavaScript function to print student names directly in the Colab cell
function printStudents(students) {
  element.innerHTML += "<b>Student List:</b><br>";
  for (let i = 0; i < students.length; i++) {
    element.innerHTML += students[i] + "<br>";
  }
}

// Test the function with sample names
let studentList = ["Alice", "Bob", "Charlie"];
printStudents(studentList);
```

```
Student List:
Alice
Bob
Charlie
```

PYTHON CODE:

```
# Python function to print student names
def print_students(students):
    print("Student List:")
    for name in students:
        print(name)

# Test the function with sample names
student_list = ["Alice", "Bob", "Charlie"]
print_students(student_list)
```

```
Student List:
Alice
Bob
Charlie
```

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 a method display_details() that prints car details. Translate the same into a Java class Car with attributes and a method displayDetails().

PYTHON CODE:

```
# Define a Python class Car
class Car:
    def __init__(self, brand, model, year):
        self.brand = brand
        self.model = model
        self.year = year

    def display_details(self):
        print("Car Details:")
        print("Brand:", self.brand)
        print("Model:", self.model)
        print("Year:", self.year)

# Create an object and call the method
car1 = Car("Toyota", "Corolla", 2020)
car1.display_details()
```

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

JS CODE:

```
%%writefile Car.java
// Java program with a Car class and displayDetails() method
public class Car {
    String brand;
    String model;
    int year;

    // Constructor
    public Car(String brand, String model, int year) {
        this.brand = brand;
        this.model = model;
        this.year = year;
    }

    // Method to display car details
    void displayDetails() {
        System.out.println("Car Details:");
        System.out.println("Brand: " + brand);
        System.out.println("Model: " + model);
        System.out.println("Year: " + year);
    }

    // Main method
    public static void main(String[] args) {
        Car car1 = new Car("Toyota", "Corolla", 2020);
        car1.displayDetails();
    }
}
```

```
Writing Car.java
```

```
!javac Car.java
```

```
!java Car
```

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

✅ 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.