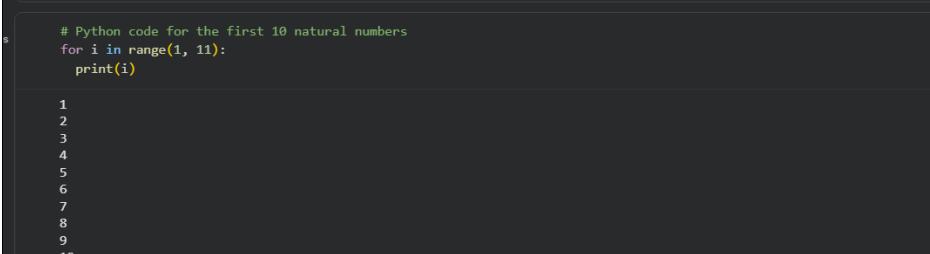
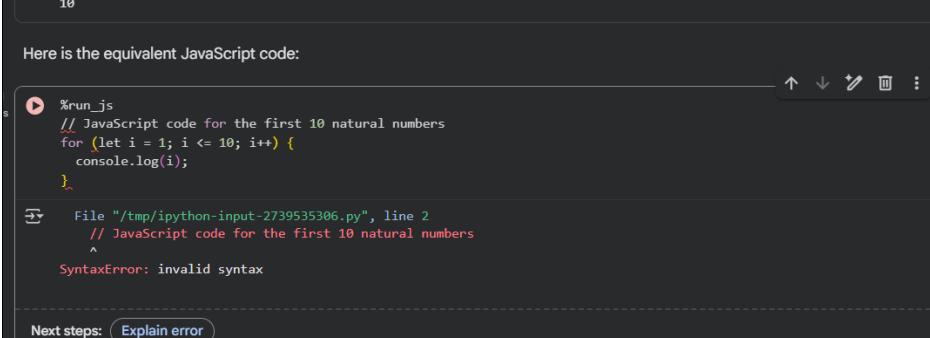


SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s) Name		Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij 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 :</p> <p>Translate the following Python code of 1st 10 natural numbers into equivalent JavaScript code. The JavaScript version should have the same logic, output, and structure.(</p> <p>Write a python code for 1st 10 natural numbers and then proceed with translation)</p>  <pre># Python code for the first 10 natural numbers for i in range(1, 11): print(i) 1 2 3 4 5 6 7 8 9 10</pre> <p>Here is the equivalent JavaScript code:</p>  <pre>%run_js // JavaScript code for the first 10 natural numbers for (let i = 1; i <= 10; i++) { console.log(i); } File "/tmp/ipython-input-2739535306.py", line 2 // JavaScript code for the first 10 natural numbers ^ SyntaxError: invalid syntax</pre> <p>Next steps: Explain error</p>	
	<p>Task 2: Convert Conditional Statements (Java → Python)</p> <ul style="list-style-type: none"> • 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 code to check whether user input number is positive, negeative or zero. Convert java code into python

The terminal window shows two parts of code. The top part is Java code named `NumberCheck.java`, which prompts the user for a number and prints whether it's positive, negative, or zero. The bottom part is the equivalent Python code, which does the same thing. A command-line interface shows the Python code being run and a user entering the number 2, with the output "The number is positive."

```
+ %> writefile NumberCheck.java
import java.util.Scanner;

public class NumberCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        double number = scanner.nextDouble();

        if (number > 0) {
            System.out.println("The number is positive.");
        } else if (number < 0) {
            System.out.println("The number is negative.");
        } else {
            System.out.println("The number is zero.");
        }
        scanner.close();
    }
}

Writing NumberCheck.java

Here is the equivalent Python code:

# Python code to check if a number is positive, negative, or zero
number = float(input("Enter a number: "))
if number > 0:
    print("The number is positive.")
elif number < 0:
    print("The number is negative.")
else:
    print("The number is zero.")

Enter a number: 2
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 a recursive function in python to calucate factorial of a

number .Convert the code into c++code

```
1] 0s
# Recursive function to calculate factorial in Python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

# Example usage
num = 5
print(f"The factorial of {num} is {factorial(num)}")

The factorial of 5 is 120
```

Here is the equivalent C++ code:

```
1] 0s
%%writefile main.cpp
#include <iostream>

// Recursive function to calculate factorial in C++
long long factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {
    int num = 5;
    std::cout << "The factorial of " << num << " is " << factorial(num) << std::endl;
    return 0;
}
Writing main.cpp
```

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 program to list student names and print them , convert the java script function into python using lists

```
%run_js
// JavaScript program to list and print student names
let studentNames = ["Alice", "Bob", "Charlie", "David"];

console.log("Student Names:");
for (let i = 0; i < studentNames.length; i++) {
    console.log(studentNames[i]);
}

File "/tmp/ipython-input-995076762.py", line 2
    ^
SyntaxError: invalid syntax
```

Next steps: [Explain error](#)

Here is the equivalent Python code:

```
# Python program to list and print student names using lists
student_names = ["Alice", "Bob", "Charlie", "David"]

print("Student Names:")
for name in student_names:
    print(name)
```

```
Student Names:
Alice
Bob
Charlie
David
```

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 to represent car with attributes and method to display details. Now , trnslate the python class into java

The screenshot shows a code editor interface with two panes. The left pane displays Python code for a 'Car' class, and the right pane displays the equivalent Java code generated by Gemini. The Python code is as follows:

```
# Python class to represent a car
class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def display_details(self):
        print(f"Make: {self.make}, Model: {self.model}, Year: {self.year}")

# Example usage
my_car = Car("Toyota", "Corolla", 2020)
my_car.display_details()

Make: Toyota, Model: Corolla, Year: 2020
```

Below the Python code, a note says 'Here is the equivalent Java code:' followed by the generated Java code:

```
❸ Gemini
❹ %%writefile Car.java
// Java class to represent a car
public class Car {
    String make;
    String model;
    int year;

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

    // Method to display details
    public void displayDetails() {
        System.out.println("Make: " + this.make + ", Model: " + this.model + ", Year: " + this.year);
    }

    // Main method for example usage (optional, but good for testing)
    public static void main(String[] args) {
        Car myCar = new Car("Honda", "Civic", 2023);
        myCar.displayDetails();
    }
}

❺ Writing Car.java
```

✓ 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.