SCHOOLOFCOMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName: M. Tech/MCA/MSC		**AssignmentType: Lab**	**AcademicYear:**2025-2026
CourseCoordinatorName		Venkataramana Veeramsetty	
CourseCode		**CourseTitle**	AI Assisted Problem Solving Using Python
Year/Sem	I/I	**Regulation**	R25
DateandDay of Assignment	08.12.2025	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	
AssignmentNumber:19.4(Presentassignmentnumber)/**24**(Totalnumberofassignments)			

Q.No.	Question	*ExpectedTime to complete*

Output

Status : Successfully executed

Time: 0.0100 secs

Memory: 8.968 Mb

Your Output

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

Javascript Version:

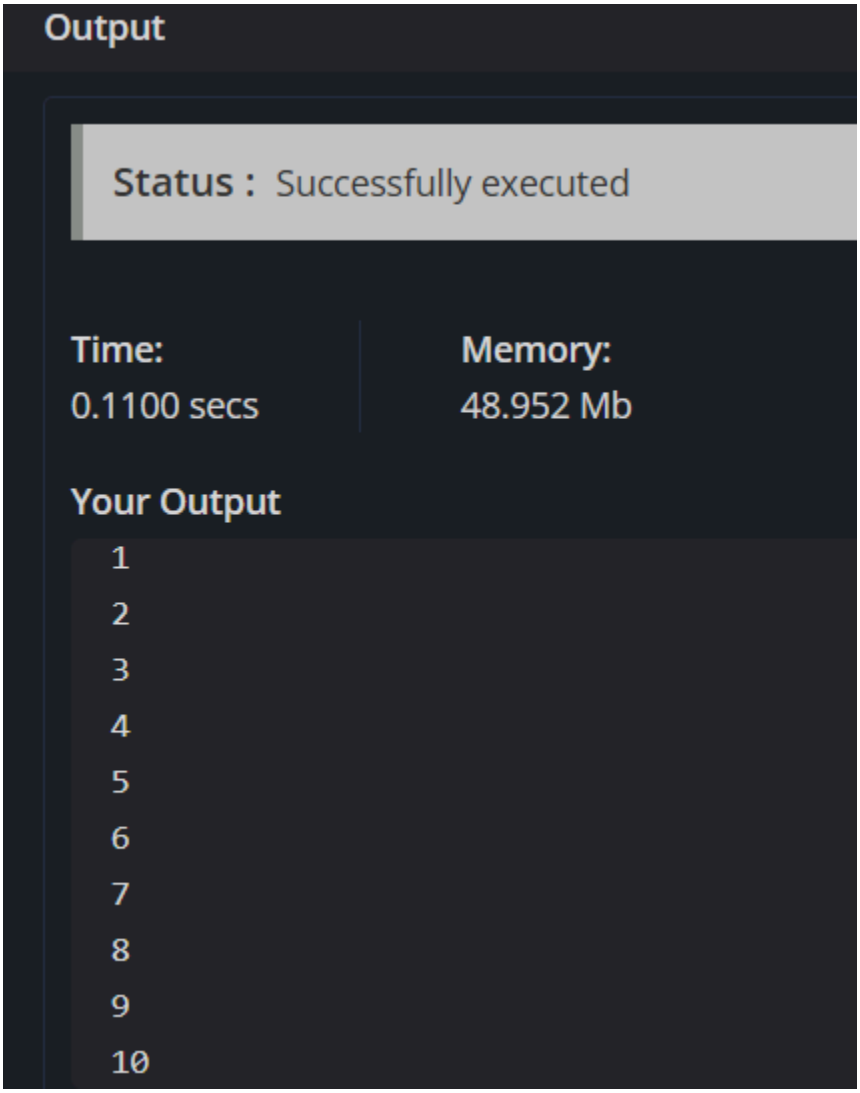
Code:

JavaScript

```
function printNumbers() {
    for (let i = 1; i <= 10; i++) {
        console.log(i);
    }
}

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

Output:



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

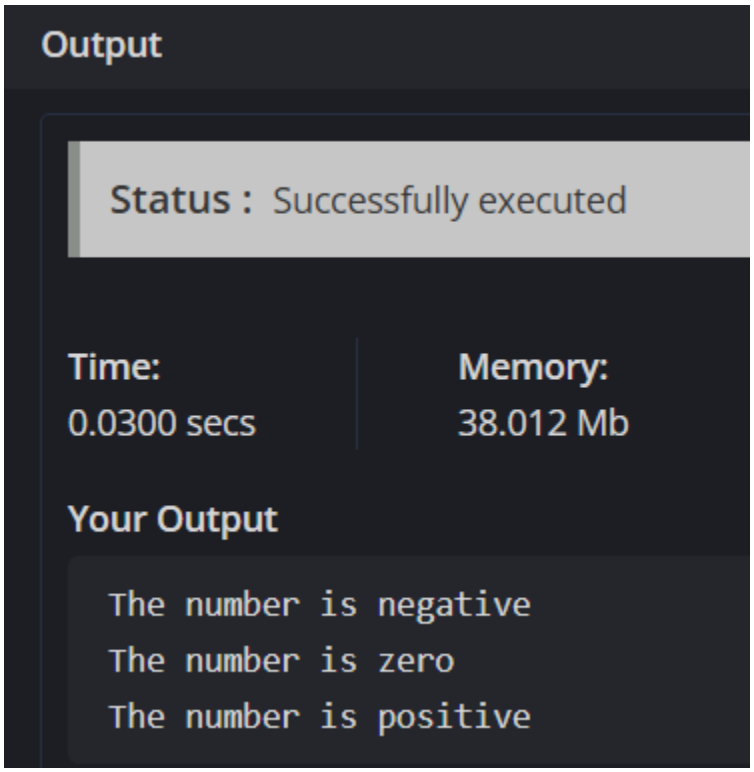
Java Version:

Code:

```java
public class Main {
    public 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) {
        checkNumber(-5);
        checkNumber(0);
        checkNumber(7);
    }
}
```

Output:



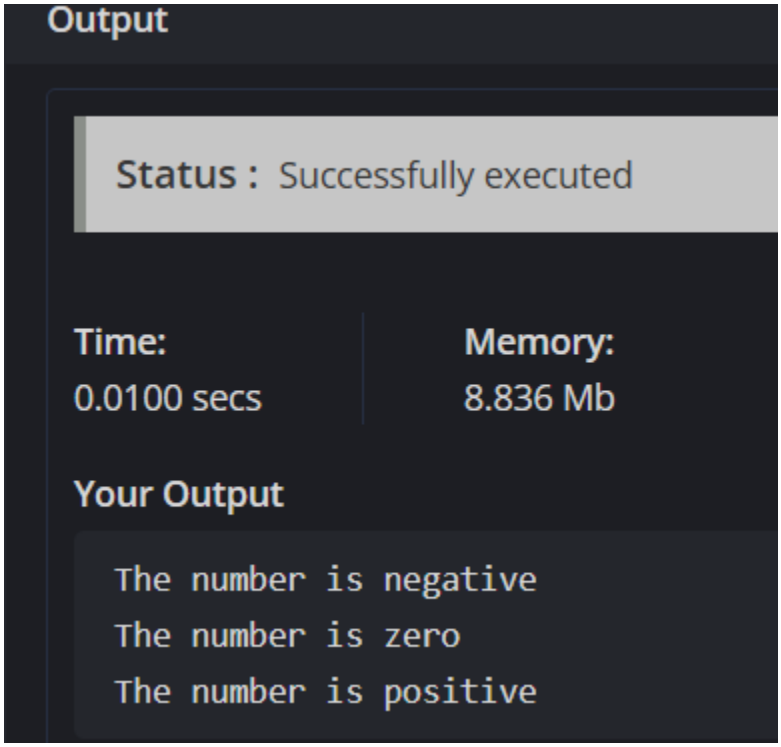
Python Version:

Code:

```python
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")

# Calling the function
check_number(-5)
check_number(0)
check_number(7)
```

Output:



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.

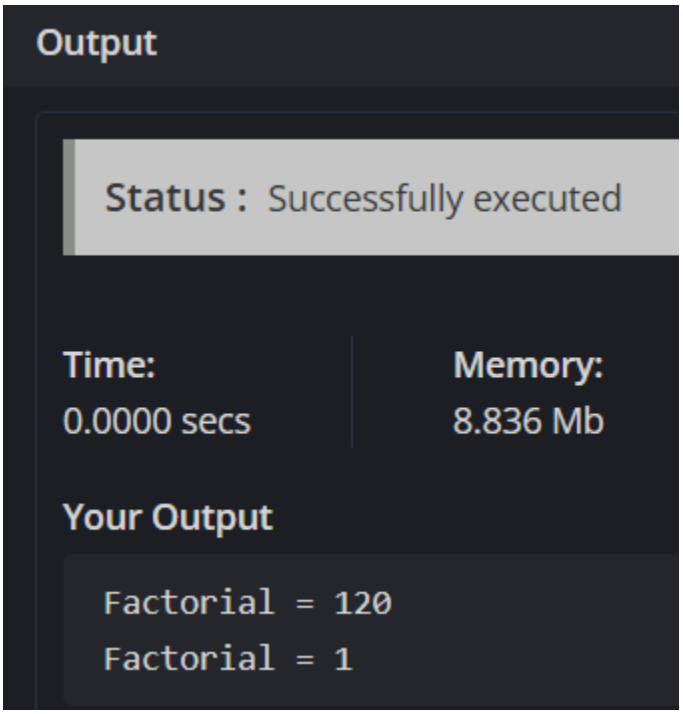
- **Expected Output:**
  - **Input: 5 → Output: Factorial = 120**
  - **Input: 0 → Output: Factorial = 1**

Python Version:
Code:

```
Python3

1  def factorial(n):
2      if n == 0:
3          return 1
4      else:
5          return n * factorial(n - 1)
6
7  # Calling the function
8  print("Factorial =", factorial(5))
9  print("Factorial =", factorial(0))
10
```

Output:



C++ Version:
Code:

C++

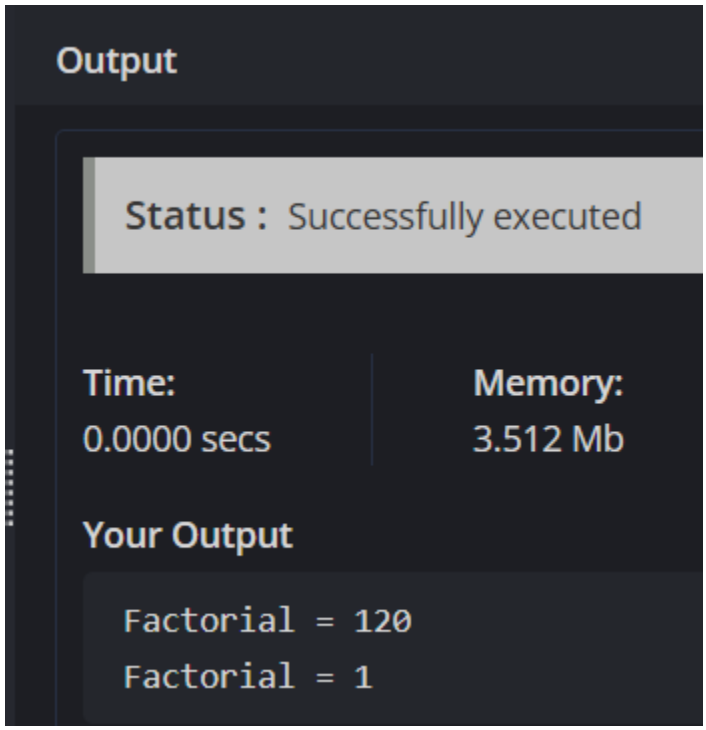
```cpp
#include <iostream>
using namespace std;

int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {
    cout << "Factorial = " << factorial(5) << endl;
    cout << "Factorial = " << factorial(0) << endl;
    return 0;
}

```

Output:



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

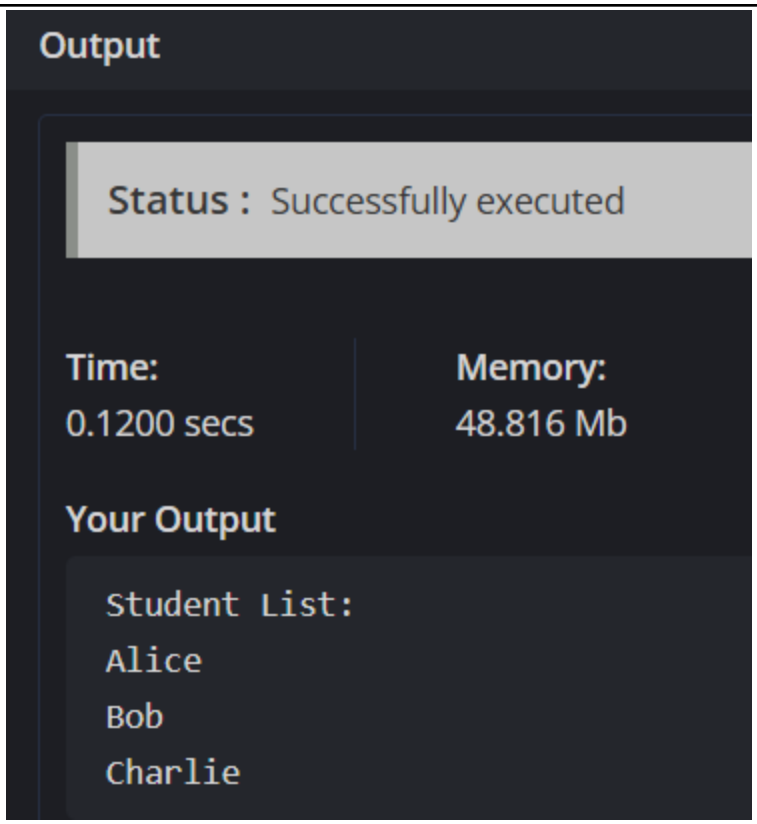
Javascript Version:

Code:

```javascript
function printStudents(students) {
    console.log("Student List:");
    for (let i = 0; i < students.length; i++) {
        console.log(students[i]);
    }
}

// Calling the function
printStudents(["Alice", "Bob", "Charlie"]);
```

Output:



Python Version:

Code:

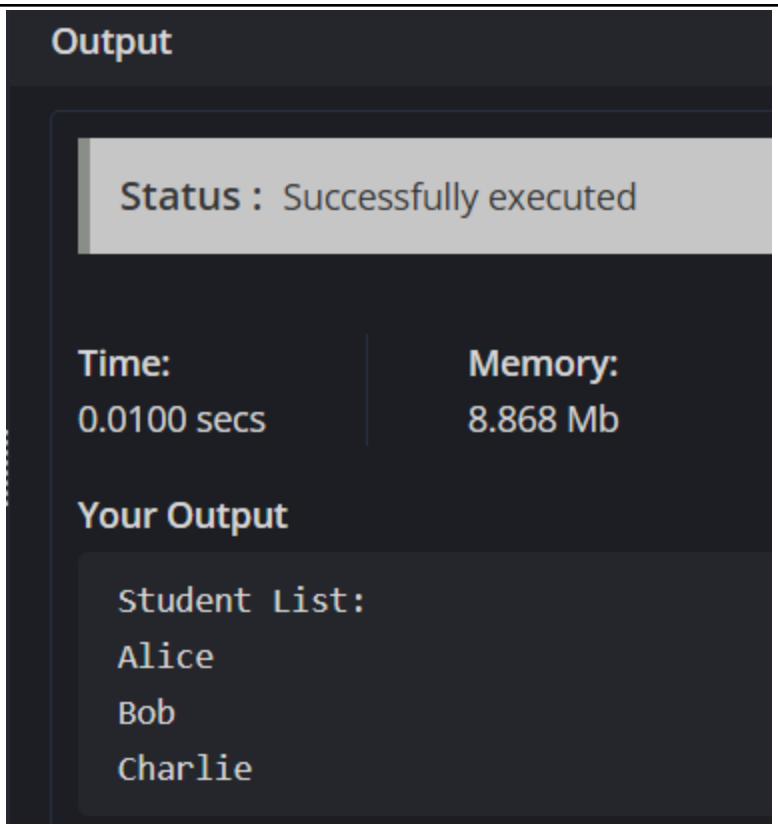
```
Python3

def print_students(students):
    print("Student List:")
    for student in students:
        print(student)

# Calling the function
print_students(["Alice", "Bob", "Charlie"])

```

Output:



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

Python Version:

Code:

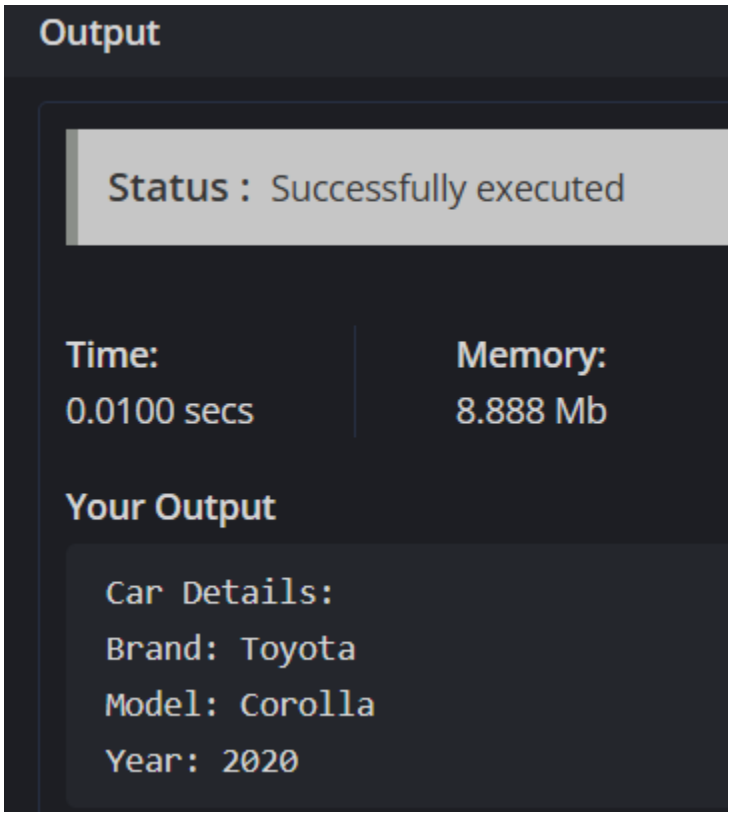
Python3

```python
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)

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

Output:



Java Version:

Code:

Java

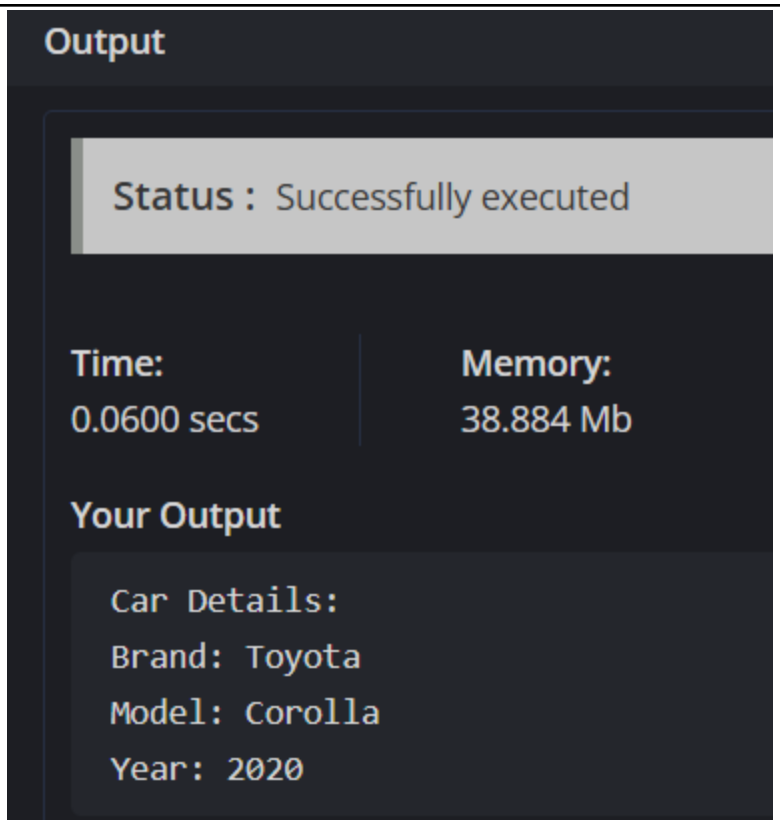
```java
class Car {
    String brand;
    String model;
    int year;

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

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

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

Output:



✅ 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.