ASSIGNMENT-19.4

TASK-1: Translate a Simple Program (Python →JavaScript)

Prompt:

Write a python program to print natural numbers from 1 to 10 with loops.

CODE WITH OUTPUT:

Python code converting in to Java script code:

Python program:

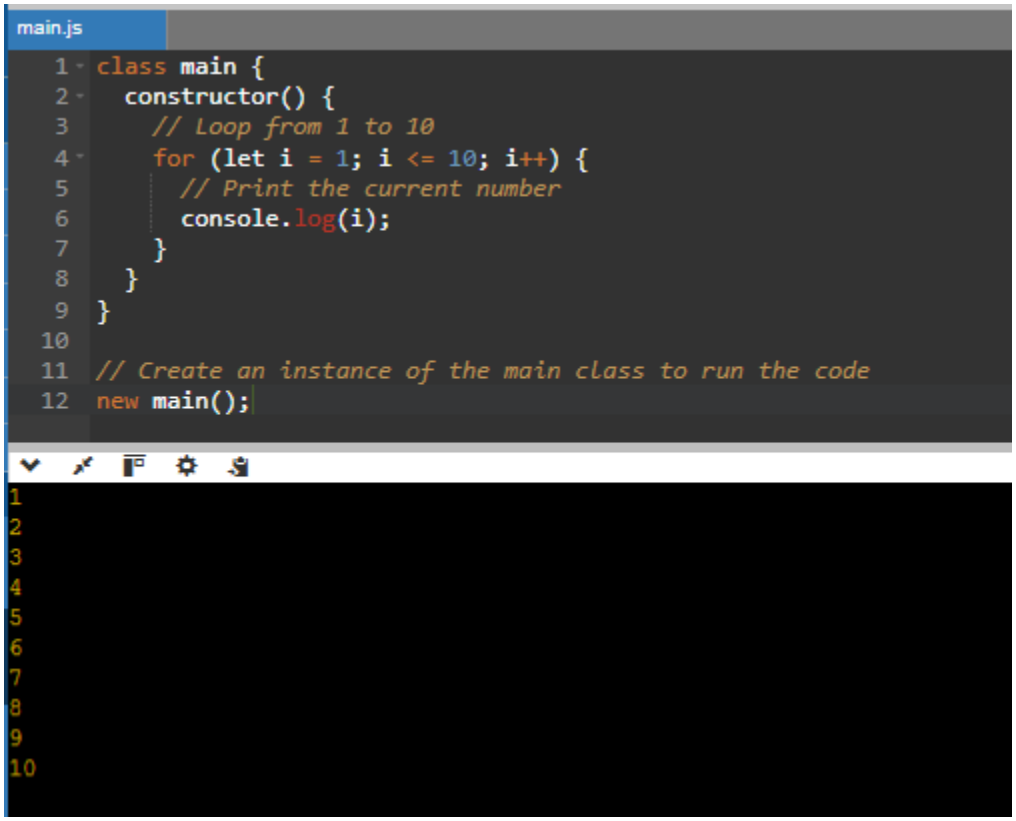
```
def print_numbers_1_to_10():
  """Prints numbers from 1 to 10 using a for loop."""
  # This loop iterates through numbers from 1 up to (but not including) 11
  for i in range(1, 11):
    # This line prints the current value of i in each iteration
    print(i)

print_numbers_1_to_10()
```

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

Java script Program:

Write a program in java script to print from 1 to 10 having class name main.



TASK-2: Convert Conditional Statements (Java → Python)

Prompt:

Write a java program with a function checknumber(int num) that prints positive, negative or zero demonstrate with examples.

CODE WITH OUTPUT:

Java code converting in to Python code:

Java program:

Main.java

```java
/**
 * This class contains a method to check if a number is positive, negative, or zero.
 */
class Main {

    /**
     * Checks if a number is positive, negative, or zero and prints the result.
     * @param num The integer to check.
     */
    public static void checknumber(int num) {
        if (num > 0) {
            System.out.println(num + " is positive.");
        } else if (num < 0) {
            System.out.println(num + " is negative.");
        } else {
            System.out.println(num + " is zero.");
        }
    }

    public static void main(String[] args) {
        // Demonstrate with examples
        checknumber(-5);
        checknumber(0);
        checknumber(7);
    }
}
```

input

```
-5 is negative.
0 is zero.
7 is positive.
```

Python Program:

Write a java program with a function checknumber(int num) that prints positive, negative or zero demonstrate with examples.

```python
def check_number(num):
  """
  Checks if a number is positive, negative, or zero and prints the result.

  Args:
    num: The integer to check.
  """
  if num > 0:
    print(f"{num} is positive.")
  elif num < 0:
    print(f"{num} is negative.")
  else:
    print(f"{num} is zero.")

# Demonstrate with examples
check_number(-5)
check_number(0)
check_number(7)
```

```
-5 is negative.
0 is zero.
7 is positive.
```

TASK-3: Translate Recursive Function (Python → C++)

Prompt:

Write a python function factorial (n) that calculates factorial of a number using recursion.

CODE WITH OUTPUT:

Python code converting in to C++ code:

Python program:

```python
def factorial(n):
  """
  Calculates the factorial of a non-negative integer using recursion.

  Args:
    n: A non-negative integer.

  Returns:
    The factorial of n.
  """
  if n == 0:
    return 1
  else:
    return n * factorial(n-1)

# Example usage
print(factorial(5))
```

```
120
```

C++ Program:

Write a C++ function factorial (n) that calculates factorial of a number using recursion.

main.cpp

```cpp
#include <iostream>

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

int main() {
    // Example usage
    int num = 5;
    std::cout << "Factorial of " << num << " is " << factorial(num) << std::endl;
    return 0;
}

```

input

```
Factorial of 5 is 120
```

TASK-4: Data Structures with Functions (JavaScript → Python).

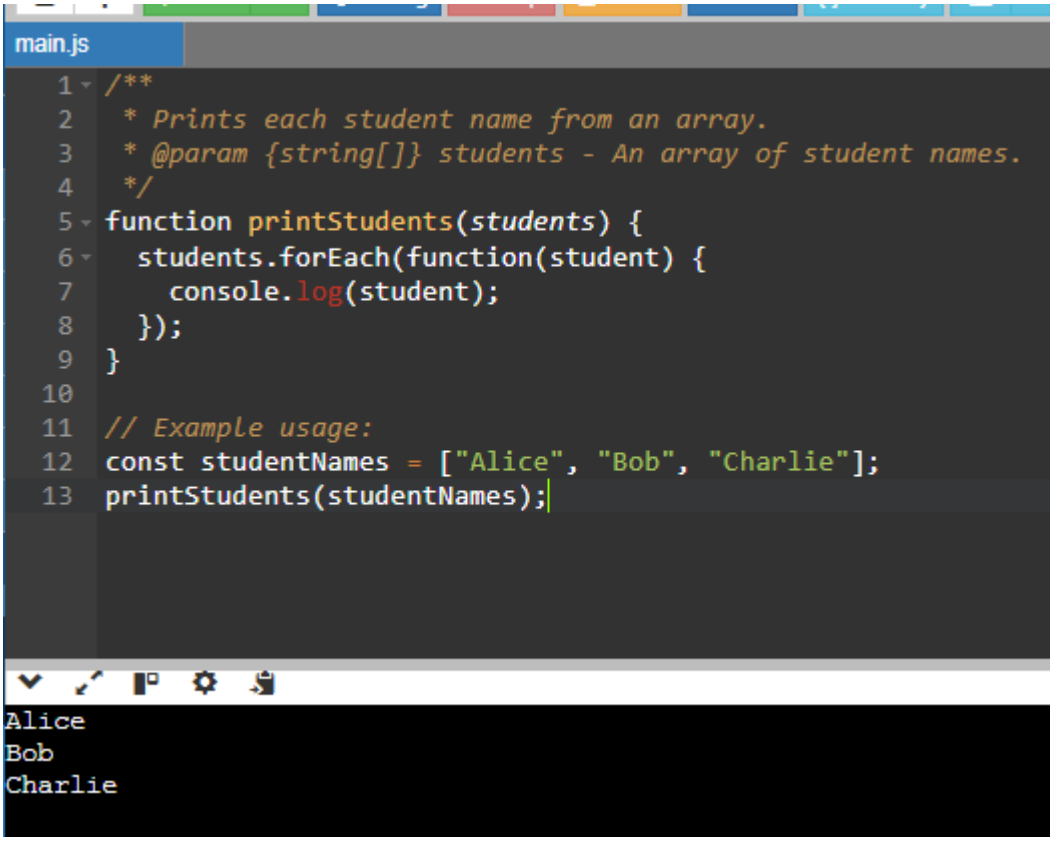
Prompt:

Write a JavaScript function print Students(students) that takes an array of student names and prints each name.

CODE WITH OUTPUT:

Java script code converting in to Python code:

Java script program:



Python program:

Write a python function print Students(students) that takes an array of student names and prints each name using list.

```
def print_students(students):
  """
  Prints each student name from a list.

  Args:
    students: A list of student names.
  """
  for student in students:
    print(student)

# Example usage:
student_names = ["Alice", "Bob", "Charlie"]
print_students(student_names)
```

```
Alice
Bob
Charlie
```

TASK-5: Class & Object Translation (Python → Java)

Prompt:

Write a Python program class Car with attributes like brand, model, year and add a method display_details() that prints car details.

CODE WITH OUTPUT:

Python code converting into java code:

Python program:

```python
class Car:
  """
  Represents a car with brand, model, and year attributes.
  """
  def __init__(self, brand, model, year):
    """
    Initializes a new Car object.

    Args:
      brand: The brand of the car.
      model: The model of the car.
      year: The manufacturing year of the car.
    """
    self.brand = brand
    self.model = model
    self.year = year

  def display_details(self):
    """
    Prints the details of the car.
    """
    print(f"Brand: {self.brand}")
    print(f"Model: {self.model}")
    print(f"Year: {self.year}")

# Example usage:
my_car = Car("Toyota", "Camry", 2022)
my_car.display_details()
```

```
Brand: Toyota
Model: Camry
Year: 2022
```

Java program:

Write a java program class Car with attributes like brand,model, year and add a method display_details() that prints car details.

Car.java

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

    /**
     * Initializes a new Car object.
     * @param brand The brand of the car.
     * @param model The model of the car.
     * @param year The manufacturing year of the car.
     */
    public Car(String brand, String model, int year) {
        this.brand = brand;
        this.model = model;
        this.year = year;
    }

    /**
     * Prints the details of the car.
     */
    public void displayDetails() {
        System.out.println("Brand: " + brand);
        System.out.println("Model: " + model);
        System.out.println("Year: " + year);
    }

    public static void main(String[] args) {
        // Example usage:
        Car myCar = new Car("Honda", "Civic", 2023);
        myCar.displayDetails();
    }
}
```

```
Brand: Honda
Model: Civic
Year: 2023
```