

LAB ASSIGNMENT-19

Task-1:

Code and Output: Python

program:

Write a Python function print_numbers() that prints the first 10 natural numbers using a loop.

```
def print_first_10_numbers():
    """Prints the first 10 numbers using a loop."""
    for i in range(1, 11):
        print(i)

# Call the function
print_first_10_numbers()
```

1
2
3
4
5
6
7
8
9
10

Java program:

write a program in Java to print from 1 to 10 having class name mai

The screenshot shows a Java development environment with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Code Editor:** File named "Main.java" containing the following code:

```
1- public class Main {  
2-     public static void main(String[] args) {  
3-         for (int i = 1; i <= 10; i++) {  
4-             System.out.println(i);  
5-         }  
6-     }  
7- }
```
- Output Console:** Shows the output of the program:

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

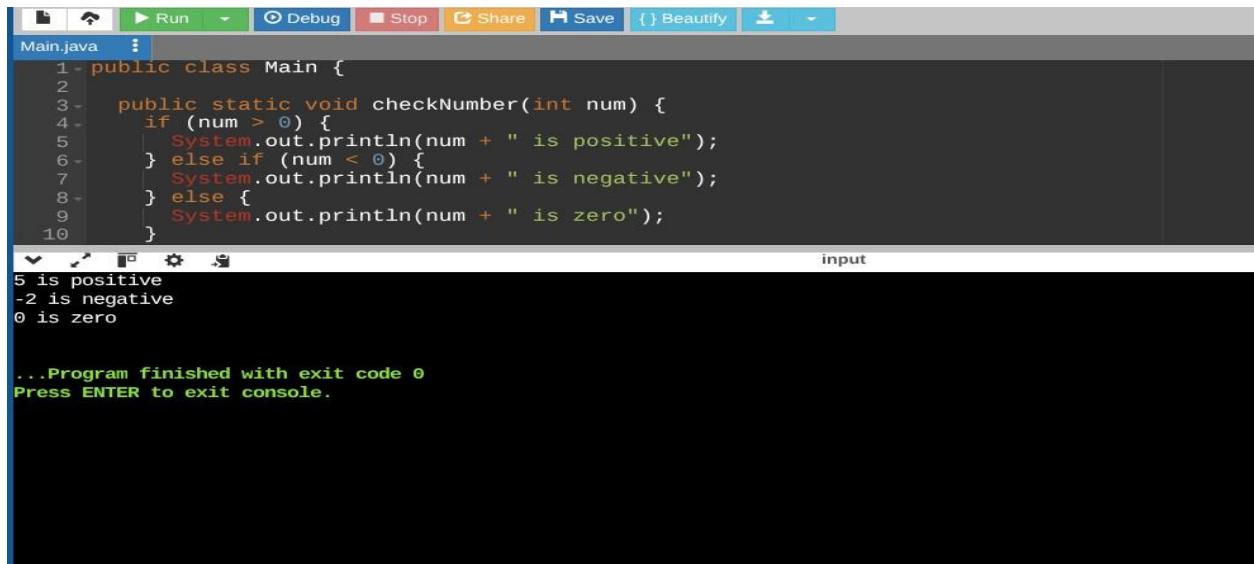
...Program finished with exit code 0
Press ENTER to exit console.

Task-2:

Code and Output:

Java program:

Write a Java method checkNumber(int num) that checks if a number is positive, negative, or zero.



The screenshot shows a Java development environment with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Code Editor:** File named "Main.java". The code defines a class Main with a static void method checkNumber that prints the sign of a given integer num.
- Output Console:** Labeled "input" at the top. It shows the execution of the program with three test cases:
 - 5 is positive
 - 2 is negative
 - 0 is zero
- Message:** "...Program finished with exit code 0" followed by "Press ENTER to exit console."

Python program:

write a program in python that checks a number is positive,negative or zero

```
[5] ✓ Os
    def check_number(num):
        """Checks if a number is positive, negative, or zero."""
        if num > 0:
            print(f"{num} is positive")
        elif num < 0:
            print(f"{num} is negative")
        else:
            print(f"{num} is zero")

    # You can call the check_number function here with different numbers to test it.
    # For example:
    check_number(5)
    check_number(-2)
    check_number(0)

→ 5 is positive
-2 is negative
0 is zero
```

Task-3:

Code and Output:

Python program:

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

```
[5] ✓ Os
    def factorial(n):
        """Calculates the factorial of a non-negative integer using recursion."""
        if n == 0:
            return 1
        else:
            return n * factorial(n-1)

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

    num = 0
    print(f"The factorial of {num} is {factorial(num)}")

    num = 7
    print(f"The factorial of {num} is {factorial(num)}")

The factorial of 5 is 120
The factorial of 0 is 1
The factorial of 7 is 5040
```

C++ program:

write a program in c++ function int factorial(int n)

The screenshot shows a code editor window with a dark theme. At the top, there is a toolbar with icons for Run, Debug, Stop, Share, Save, and Beautify. Below the toolbar, the file name "main.cpp" is visible. The code in the editor is:

```
1 #include <iostream>
2
3 int factorial(int n) {
4     if (n == 0) {
5         return 1;
6     } else {
7         return n * factorial(n - 1);
8     }
9 }
10
11 int main() {
12     int num = 5;
13     std::cout << "The factorial of " << num << " is " << factorial(num) << std::endl;
14     return 0;
15 }
16
```

Below the code editor is a terminal window with the title "input". It displays the output of the program:

```
The factorial of 5 is 120
```

At the bottom of the terminal window, the text "...Program finished with exit code 0" and "Press ENTER to exit console." is visible.

Task-4:

Code and Output:

Javascript program:

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

The screenshot shows a terminal window with the following content:

```
main.js
1 function printStudents(students) {
2   for (let i = 0; i < students.length; i++) {
3     console.log(students[i]);
4   }
5 }
6
7 // Example usage:
8 const studentNames = ["Alice", "Bob", "Charlie"];
9 printStudents(studentNames);
10
```

Below the code, the terminal shows the output:

```
Alice
Bob
Charlie
```

At the bottom, it says:

```
...Program finished with exit code 0
Press ENTER to exit console.
```

Python program:

write a python function print_students(students) using a list.

The screenshot shows a Jupyter Notebook cell with the following content:

```
[1]
✓ Os
def print_students(students_list):
    """Prints each student name from a list."""
    for student in students_list:
        print(student)

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

Below the code, the terminal shows the output:

```
Alice
Bob
Charlie
```

Task-5:

Code and Output:

Python program:

Write a Python class Car with attributes: brand, model, year. Add a method display_details() that prints car details.

```
class Car:
    """Represents a car with brand, model, and year attributes."""
    def __init__(self, brand, model, year):
        """Initializes a Car object with brand, model, and year."""
        self.brand = brand # Car brand
        self.model = model # Car model
        self.year = year # Manufacturing year of the car

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

    # Example usage:
    my_car = Car("Toyota", "Camry", 2022) # Create a Car object
    my_car.display_details() # Call the method to display car details
```

Brand: Toyota
Model: Camry
Year: 2022

Java program:

write a Java program with class car with attributes and a method displayDetails()

The screenshot shows a Java code editor and a terminal window. The code editor displays the following Java code:

```
Car.java  ::

1. public class Car {
2.     String brand;
3.     String model;
4.     int year;
5.
6.     public Car(String brand, String model, int year) {
7.         this.brand = brand;
8.         this.model = model;
9.         this.year = year;
10    }
11
12    public void displayDetails() {
13        System.out.println("Brand: " + brand);
14        System.out.println("Model: " + model);
15        System.out.println("Year: " + year);
16    }
17
18    public static void main(String[] args) {
19        Car myCar = new Car("Honda", "Civic", 2023);
20        myCar.displayDetails();
21    }
22 }
```

The terminal window below the code editor shows the output of the program:

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