**Experiment 2(a)**

**Aim –** To Design a program to generate n terms of Fibonacci Series using iterative and Recursive methods

**Theory–**

**Program:**

1. **Iterative Method:**

**package** govind;

**import** java.util.Scanner;

**public** **class** Fibonachi\_iterative {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of terms: ");

**int** inp = sc.nextInt();

**int** a = 0, b = 1; // First two Fibonacci numbers

System.***out***.print("Fibonacci sequence: ");

**for** (**int** i = 0; i < inp; i++) {

System.***out***.print(a + " ");

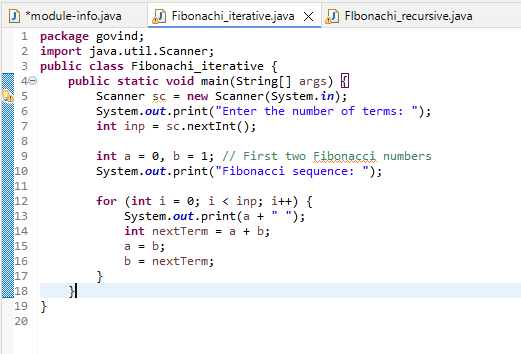
**int** nextTerm = a + b;

a = b;

b = nextTerm;

}

}

} 

1. **Recursive Method**

**package** govind;

**import** java.util.Scanner;

**public** **class** FIbonachi\_recursive{

**public** **static** **int** fibonacci(**int** n) {

// Base cases

**if** (n <= 1) {

**return** n;

}

// Recursive call

**return** *fibonacci*(n - 1) + *fibonacci*(n - 2);

}

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

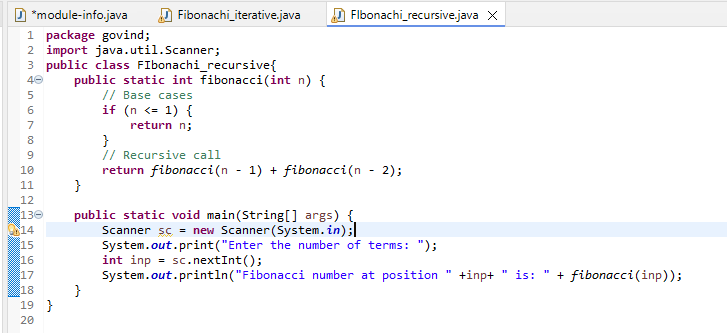
System.***out***.print("Enter the number of terms: ");

**int** inp = sc.nextInt();

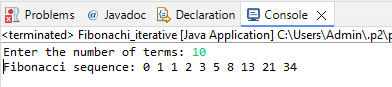
System.***out***.println("Fibonacci number at position " +inp+ " is: " + *fibonacci*(inp));

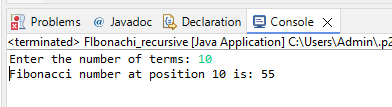
}

}



**Output:**

****

****

**Learning Outcome:**

**Experiment 2(b)**

**Aim –** To Find the Factorial of a given number using Recursion

**Theory–**

**Program:**

**package** govind;

**import** java.util.Scanner;

**public** **class** Factorial {

**public** **static** **int** factorial(**int** n) {

**if** (n == 0 || n == 1) {

**return** 1;

}

**return** n \* *factorial*(n - 1);

}

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

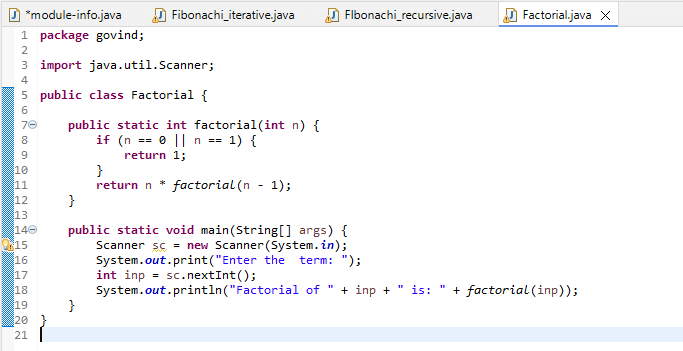
System.***out***.print("Enter the term: ");

**int** inp = sc.nextInt();

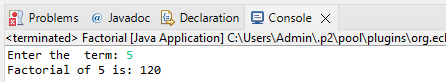
System.***out***.println("Factorial of " + inp + " is: " + *factorial*(inp));

}

}



**Output:**



**Learning Outcome:**