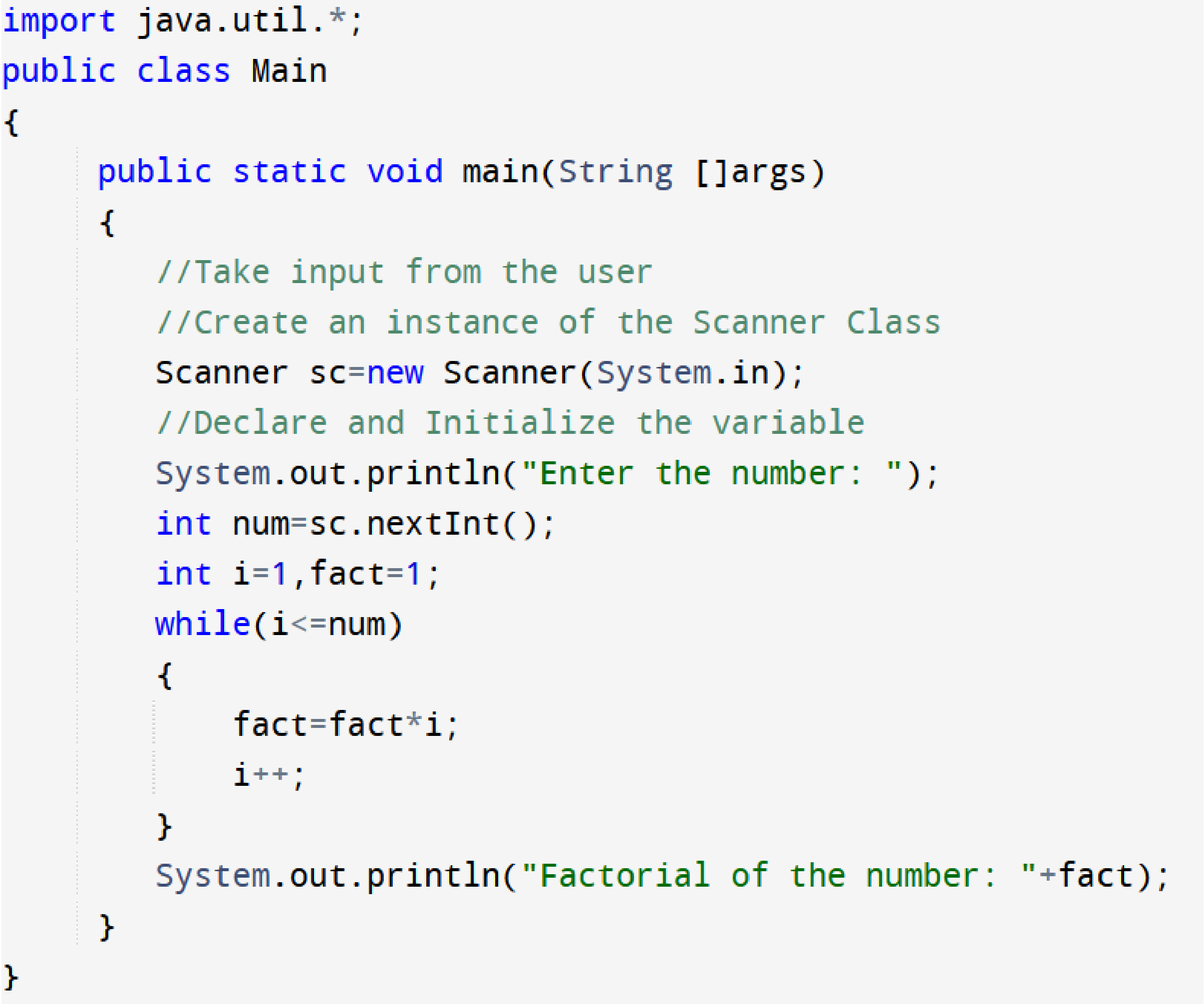
Examples of Iterative and Recursive Programs

Program 1: Write a Java Program to Find the Factorial of a Number using For Loop

//Java Program to find the Factorial of a Number using while loop



```
import java.util.*;
public class Main
{
     public static void main(String []args)
     {
        //Take input from the user
        //Create an instance of the Scanner Class
        Scanner sc=new Scanner(System.in);
        //Declare and Initialize the variable
        System.out.println("Enter the number: ");
        int num=sc.nextInt();
        int i=1,fact=1;
        while(i<=num)
        {
            fact=fact*i;
            i++;
        }
        System.out.println("Factorial of the number: "+fact);
     }
}
```

Program 2: Write a Java Program to Find the Factorial of a Number using For Loop

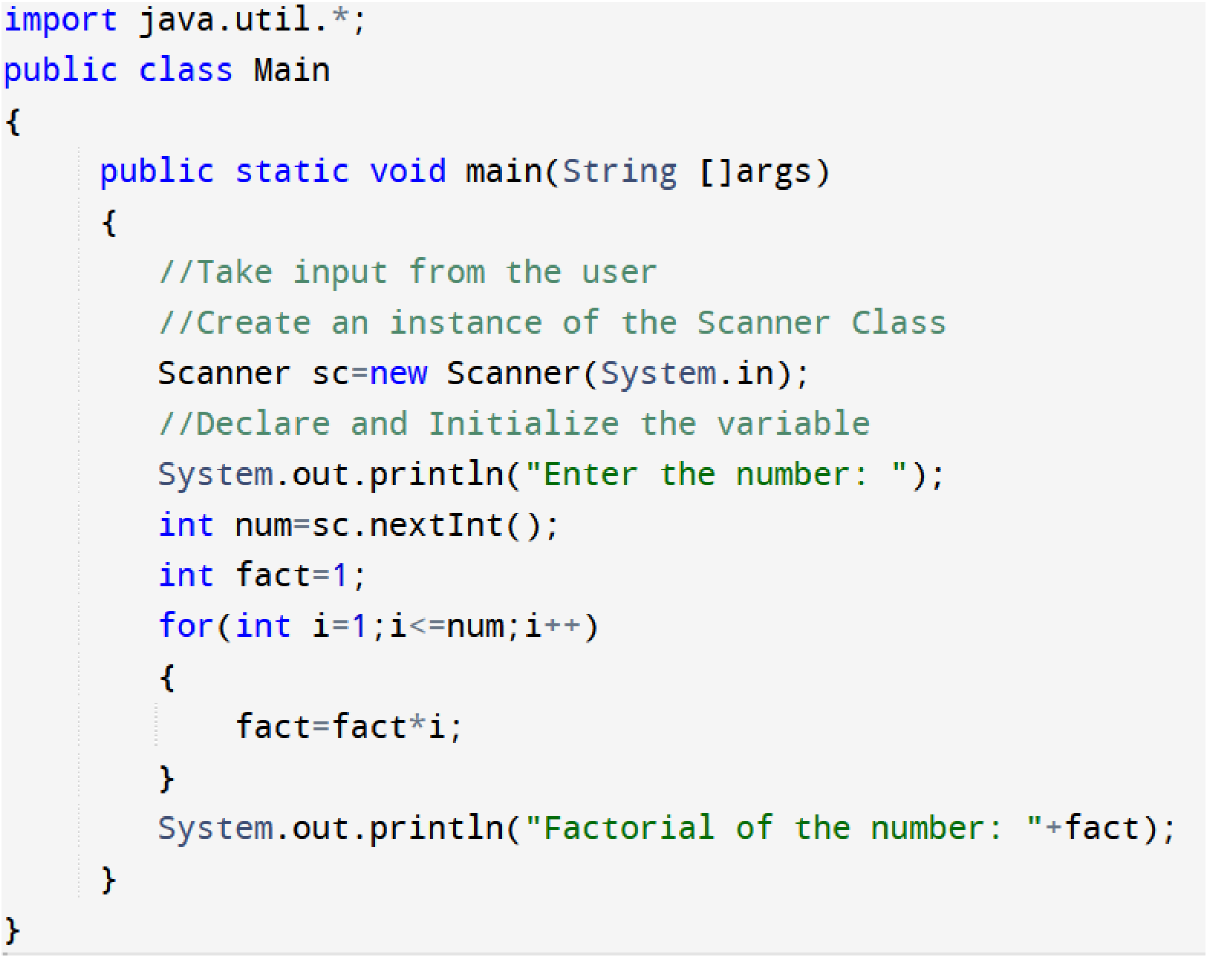
Learning Objective

In this program, we will learn how to find the factorial of a number using a for loop.

Algorithm

```
    Step 1. Start
    Step 2. Create an instance of the Scanner Class.
        Step 3. Declare a variable.
        Step 4. Ask the user to initialize the variable.
  Step 5. Declare a variable to store the factorial of the number.
Step 6.  Initialize the variable to 1.
Step 7. Use a for loop to calculate the factorial.
Step 8. Update the factorial variable by multiplying it with the loop variable in each iteration.
Step 9. Print the factorial of the number.
Step 10. Stop.
```

Below is the code example to print a factorial of a number in Java.



```java
import java.util.*;
public class Main
{
    public static void main(String []args)
    {
        //Take input from the user
        //Create an instance of the Scanner Class
        Scanner sc=new Scanner(System.in);
        //Declare and Initialize the variable
        System.out.println("Enter the number: ");
        int num=sc.nextInt();
        int fact=1;
        for(int i=1;i<=num;i++)
        {
            fact=fact*i;
        }
        System.out.println("Factorial of the number: "+fact);
    }
}
```

Recursion

-A recursive method) is a method that calls itself either directly or indirectly.

-Every recursive method has two parts:

1. Base case
   - ✓ It is also called the stopping condition
   - ✓ It is nothing more than the simplest instance of a problem
   - ✓ It consists of a condition that terminates the recursive function
2. Recursive step/ progressive case – the part that breaks downs the original problem into simpler instances of the same problem.

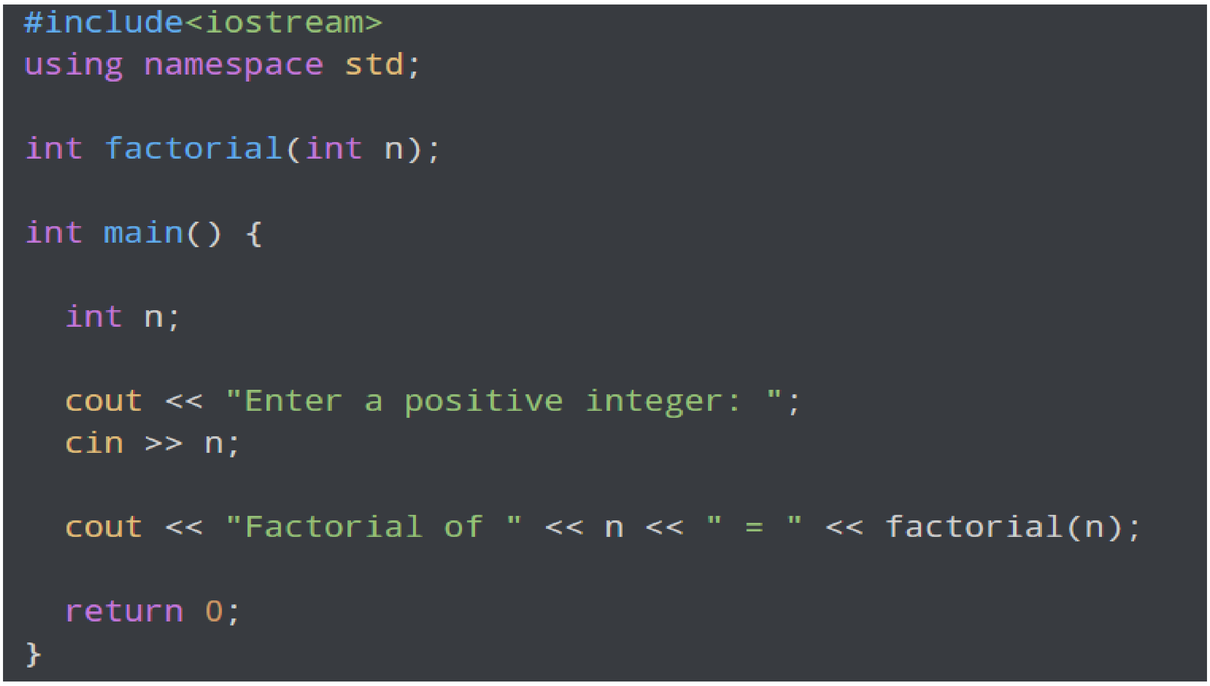
Note

-For a recursive function to work there must be an initial recursive call. This can be from a different function or from the main method/ function.

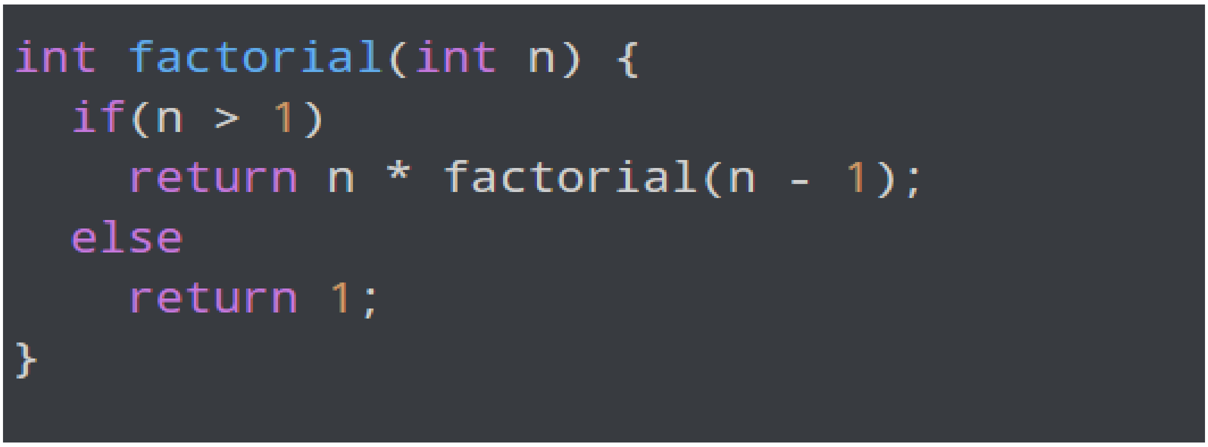
-When solving a problem using recursion we:

- divide the problem into one or more simpler or smaller parts of the problem,
- call the function (recursively) on each part, and.
- combine the solutions of the parts into a solution for the problem.

Program 3: Write a C++ Program to Find the Factorial of a Number using Recursion.

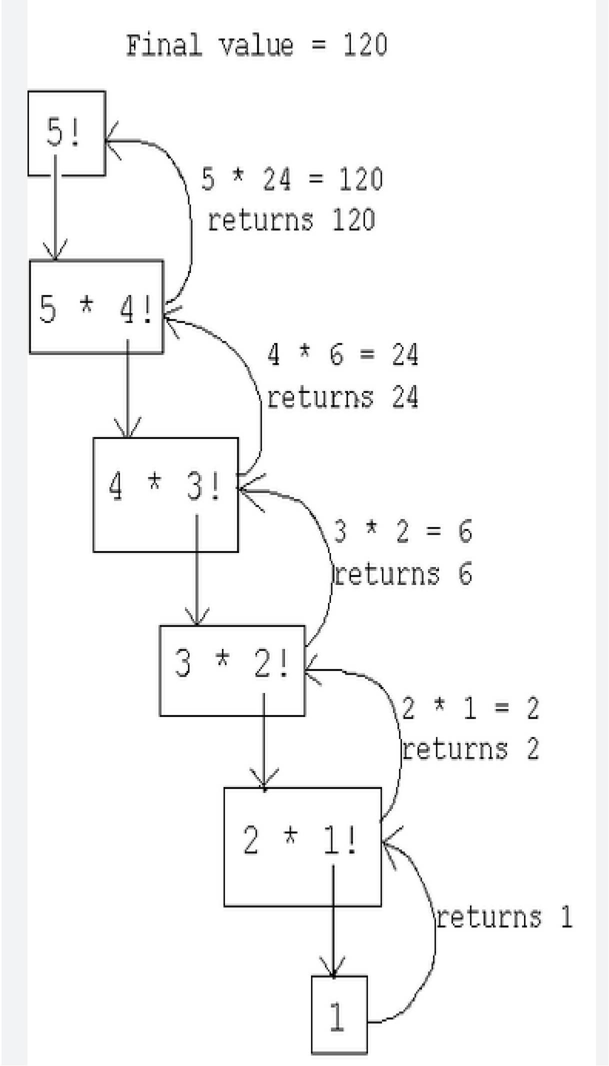
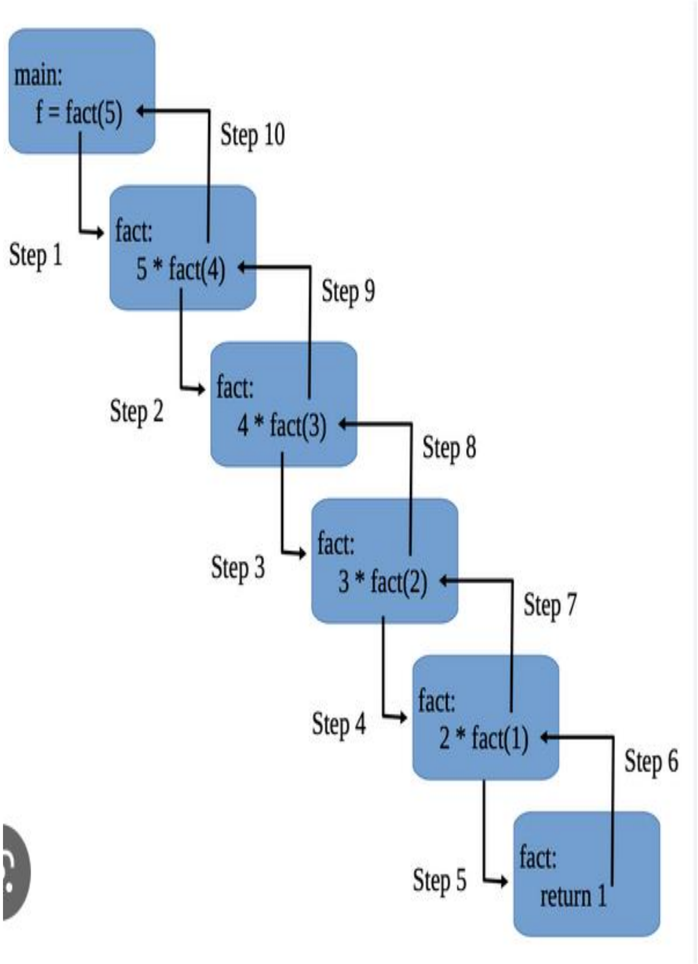


```
#include<iostream>
using namespace std;

int factorial(int n);

int main() {

  int n;

  cout << "Enter a positive integer: ";
  cin >> n;

  cout << "Factorial of " << n << " = " << factorial(n);

  return 0;
}
```



```
int factorial(int n) {
  if(n > 1)
    return n * factorial(n - 1);
  else
    return 1;
}
```

Visualisation of Recursion

 Final value = 120 5! 5 * 24 = 120 returns 120 5 * 4! 4 * 6 = 24 returns 24 4 * 3! 3 * 2 = 6 returns 6 3 * 2! 2 * 1 = 2 returns 2 2 * 1! returns 1 1	 main: f = fact(5) Step 10 Step 1 fact: 5 * fact(4) Step 9 Step 2 fact: 4 * fact(3) Step 8 Step 3 fact: 3 * fact(2) Step 7 Step 4 fact: 2 * fact(1) Step 6 Step 5 fact: return 1

Program 4: Write a Java Program to Find the Factorial of a Number using recursion

- In this program, we will find the factorial of a number using recursion with user-defined values.
- We will ask the user to enter a value and then we will calculate the factorial by calling the function recursively.

Algorithm

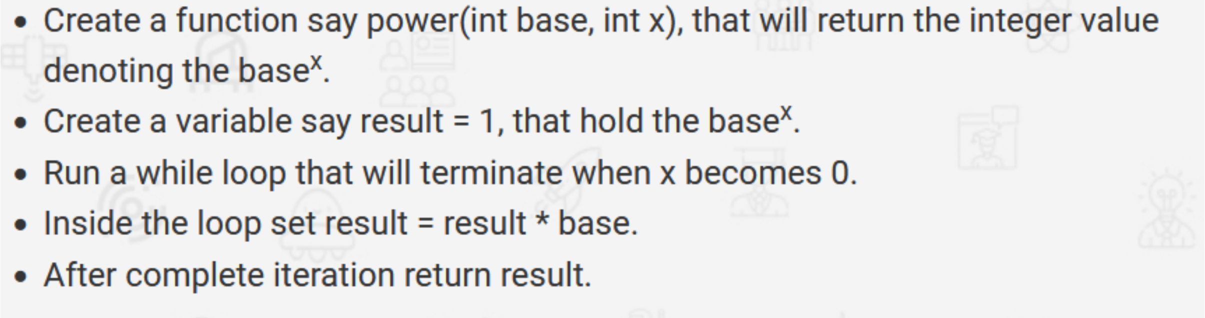
1. Start
2. Declare a variable to store a number.
3. Ask the user to initialize the number.
4. Check whether it is possible to calculate the factorial or not.
5. If the number is greater than and equal to 0, then call a recursive function to calculate the factorial of the entered number.
6. If the number is lesser than 0, print the message that it is not possible to calculate the factorial.
7. If the entered number is 0 or 1, then return 1.
8. If the entered number is other than 0 or 1, then calculate the factorial by recursively calling the same method.
9. Return the result.
10. Print the factorial of the entered number.
11. Stop

Below is the code example to print a factorial of a number in Java.

```
import java.util.Scanner;
public class Main
{
//Driver Code
public static void main(String[] args)
{
//Take input from the user
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number :");
int num = sc.nextInt(); //Input the number
if(num>=0)
{
//Call a recursive function to find the factorial
int factorial=findFactorial(num);
System.out.println("The factorial of the entered number is :"
    +factorial);
}
else
{
System.out.println("Factorial not possible.");
System.out.println("Please enter valid input.");
}
}
//Recursive Function to Find the Factorial of a Number
public static int findFactorial(int num)
{
if(num==0)
return 1;
else if(num==1)
return 1;
else
return num*findFactorial(num-1);
}
}
```

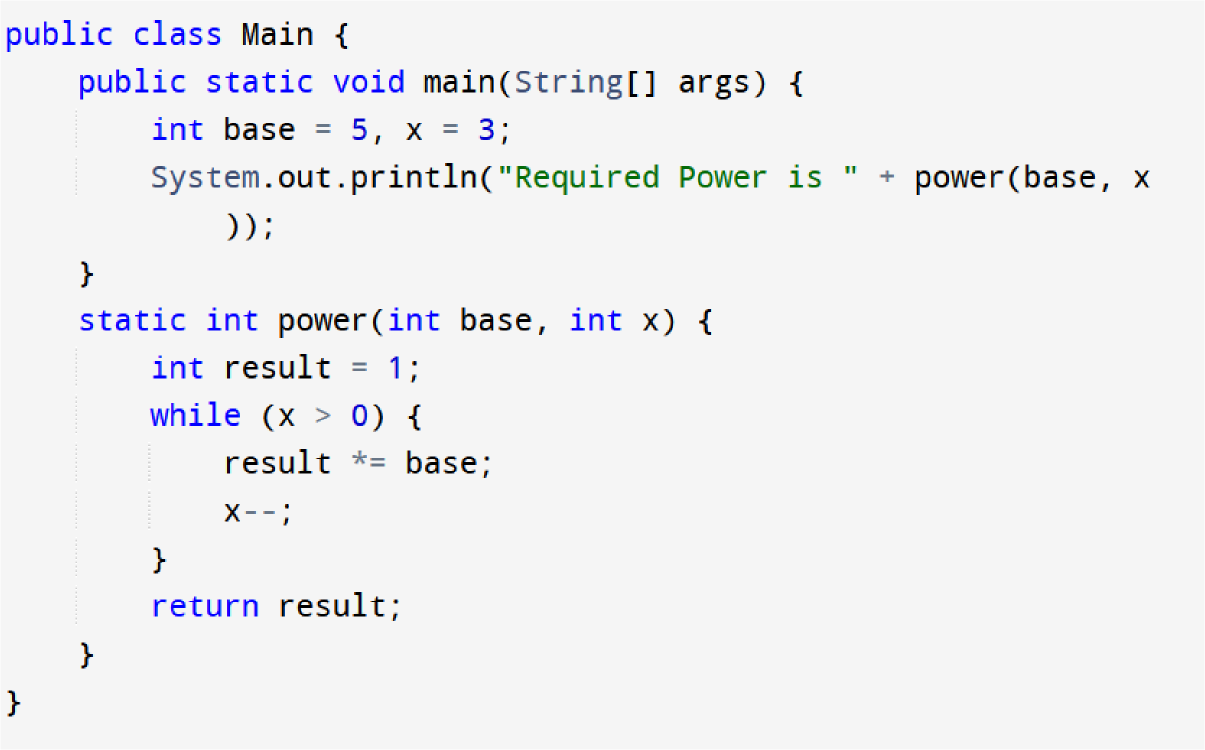
Program 4: To print power of a number using while loop

Steps



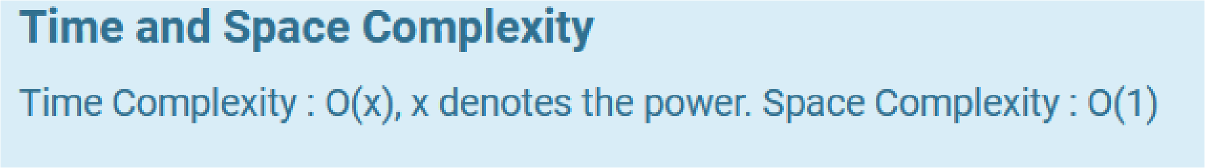
- Create a function say power(int base, int x), that will return the integer value denoting the base^x.
- Create a variable say result = 1, that hold the base^x.
- Run a while loop that will terminate when x becomes 0.
- Inside the loop set result = result * base.
- After complete iteration return result.

Java code



```java
public class Main {
    public static void main(String[] args) {
        int base = 5, x = 3;
        System.out.println("Required Power is " + power(base, x
            ));
    }
    static int power(int base, int x) {
        int result = 1;
        while (x > 0) {
            result *= base;
            x--;
        }
        return result;
    }
}
```

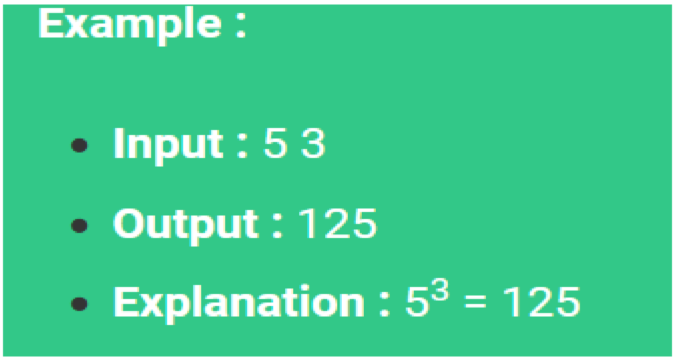
Comment on the time and space complexity of the above code.



Time and Space Complexity

Time Complexity : O(x), x denotes the power. Space Complexity : O(1)

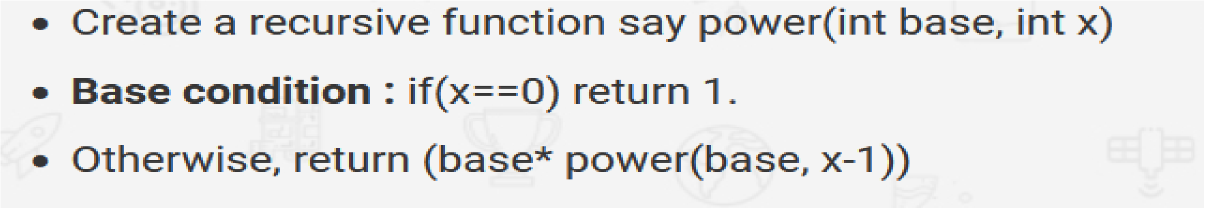
Program 5: To print power of a number using recursion



Example :

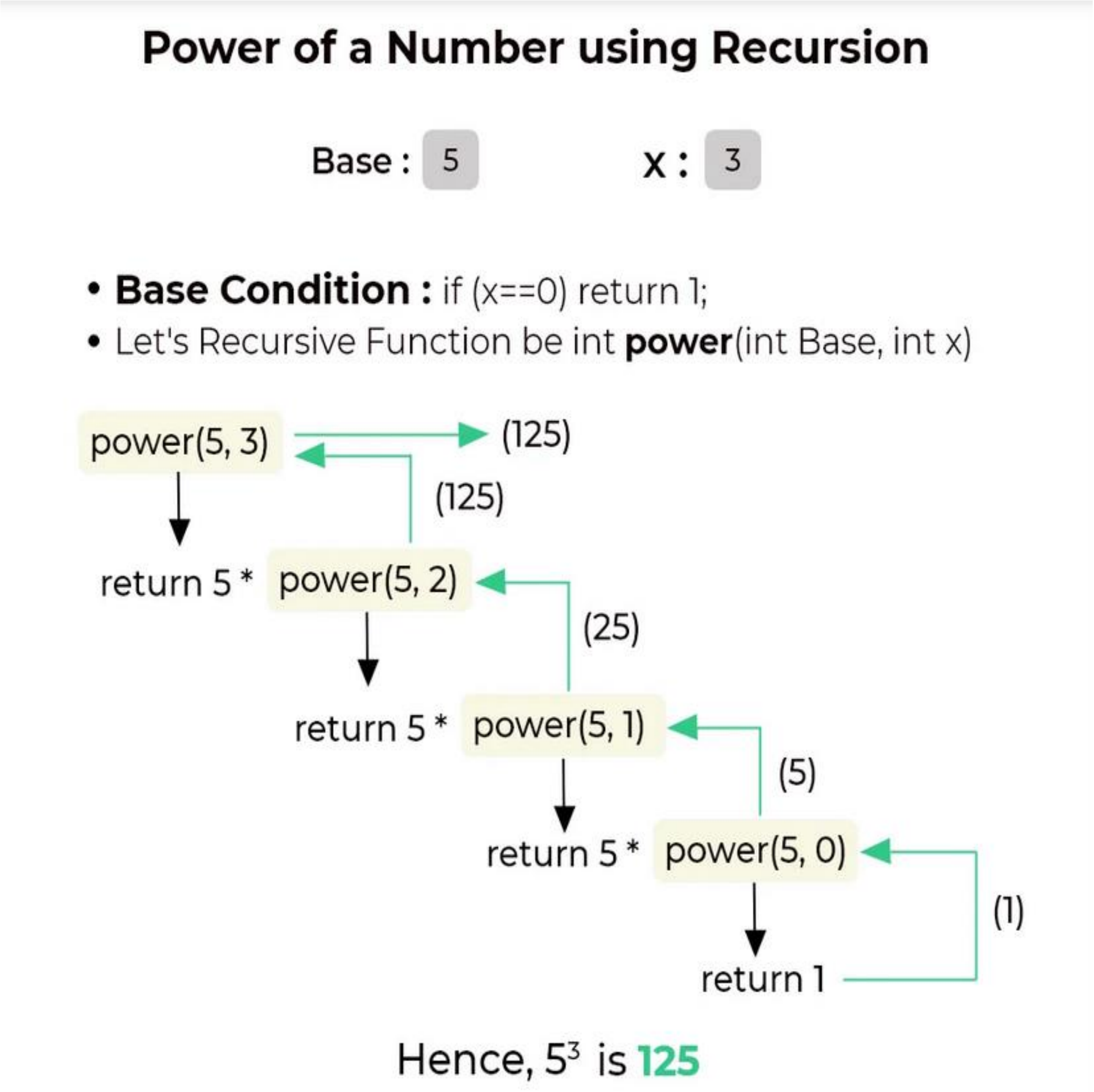
- **Input :** 5 3
- **Output :** 125
- **Explanation :** $5^3 = 125$

Steps



- Create a recursive function say power(int base, int x)
- **Base condition :** if(x==0) return 1.
- Otherwise, return (base* power(base, x-1))

Visualisation



Power of a Number using Recursion

Base : 5 x : 3

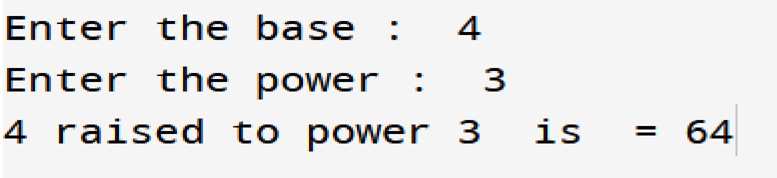
- **Base Condition :** if (x==0) return 1;
- Let's Recursive Function be int **power**(int Base, int x)

Hence, 5^3 is 125

Java code

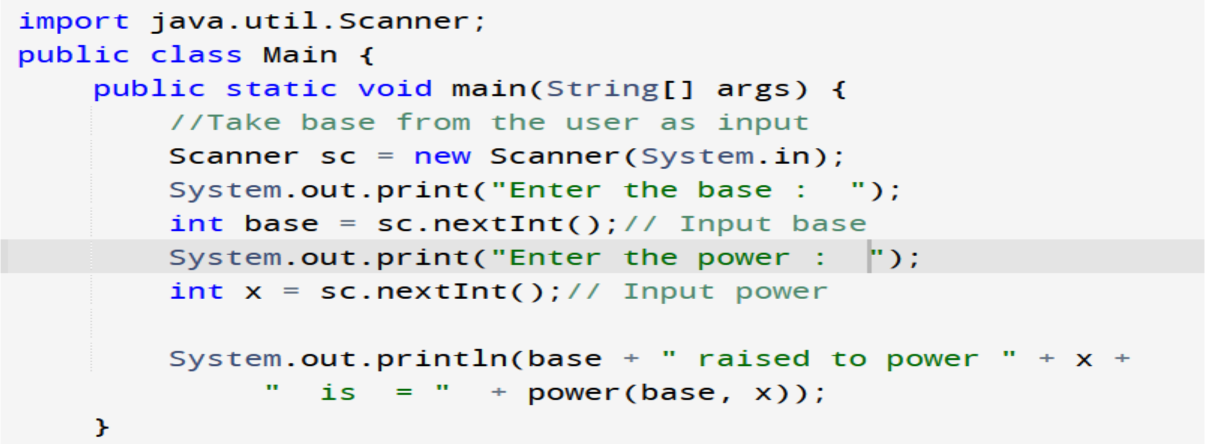
```java
public class Main
{
  public static void main (String[]args)
  {
    int base = 5, x = 3;
    System.out.println ("Required Power is " + power (base, x
        ));
  }
  //Recursive Function
  static int power (int base, int x)
  {
    if (x == 0)      //Base Condition
      return 1;
      else
    return (base * power (base, x - 1));
  }
}
```

Exercise: **Write the code snippet for the above program to allow the user to input the base and the power and print the output as follows:**



```
Enter the base :  4
Enter the power :  3
4 raised to power 3  is  = 64
```

Solution



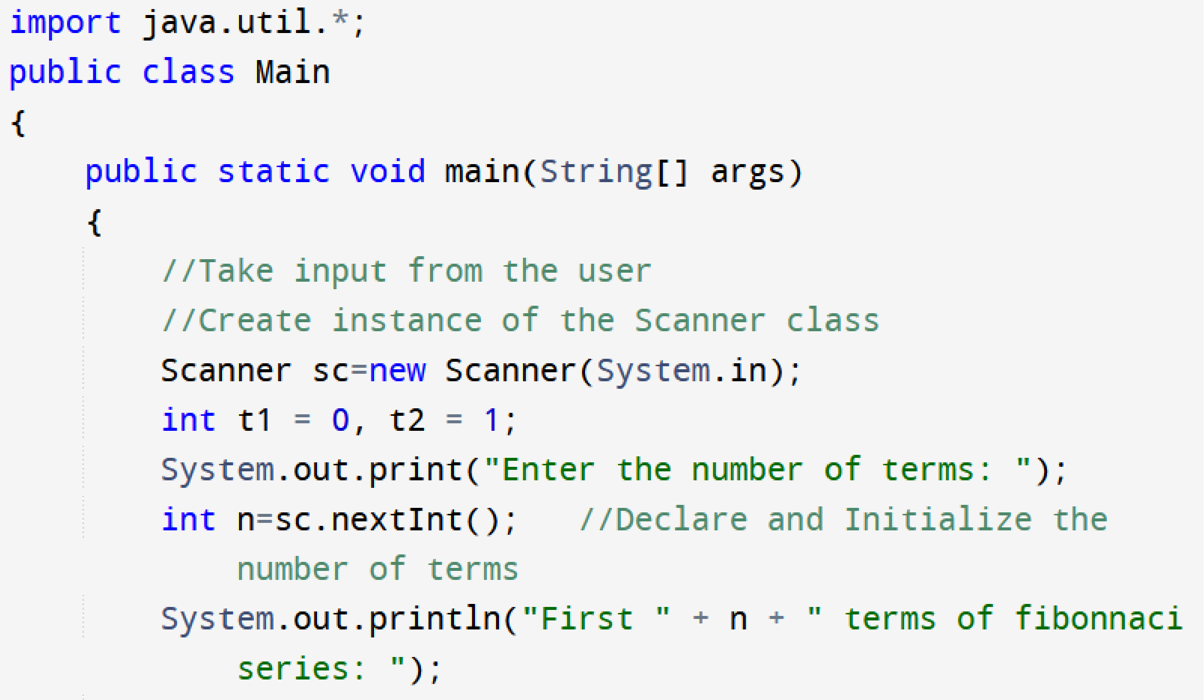
```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        //Take base from the user as input
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the base :  ");
        int base = sc.nextInt();// Input base
        System.out.print("Enter the power :  ");
        int x = sc.nextInt();// Input power

        System.out.println(base + " raised to power " + x +
            "  is  = "  + power(base, x));
    }
```

Program 6: To Print Fibonacci Series Using For Loop

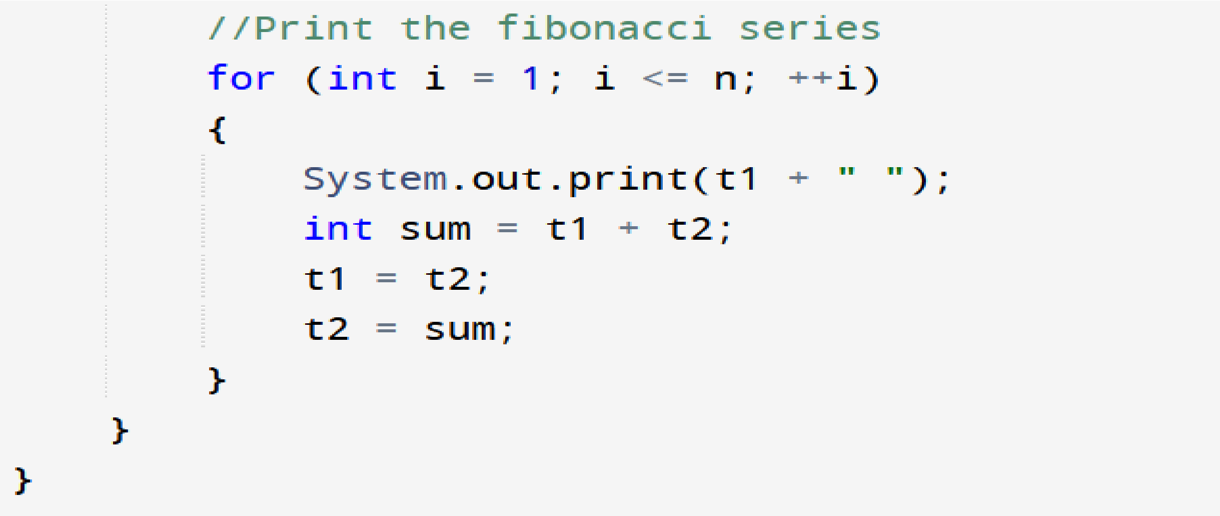
In this program, we will see how to print the Fibonacci Series in Java using for loop. Here, firstly, we will ask the user to enter the number of terms and then we will find the Fibonacci Series.

Algorithm:

1. Start
2. Declare a variable for the total number of terms.
3. Ask the user to initialize the number of terms.
4. Print the first and second numbers of the series.
5. Use a for loop to print the Fibonacci series up to that number of terms.
6. Update the series terms in each iteration.
7. Print the Fibonacci series.
8. Stop



```
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        //Take input from the user
        //Create instance of the Scanner class
        Scanner sc=new Scanner(System.in);
        int t1 = 0, t2 = 1;
        System.out.print("Enter the number of terms: ");
        int n=sc.nextInt();   //Declare and Initialize the
            number of terms
        System.out.println("First " + n + " terms of fibonnaci
            series: ");
```



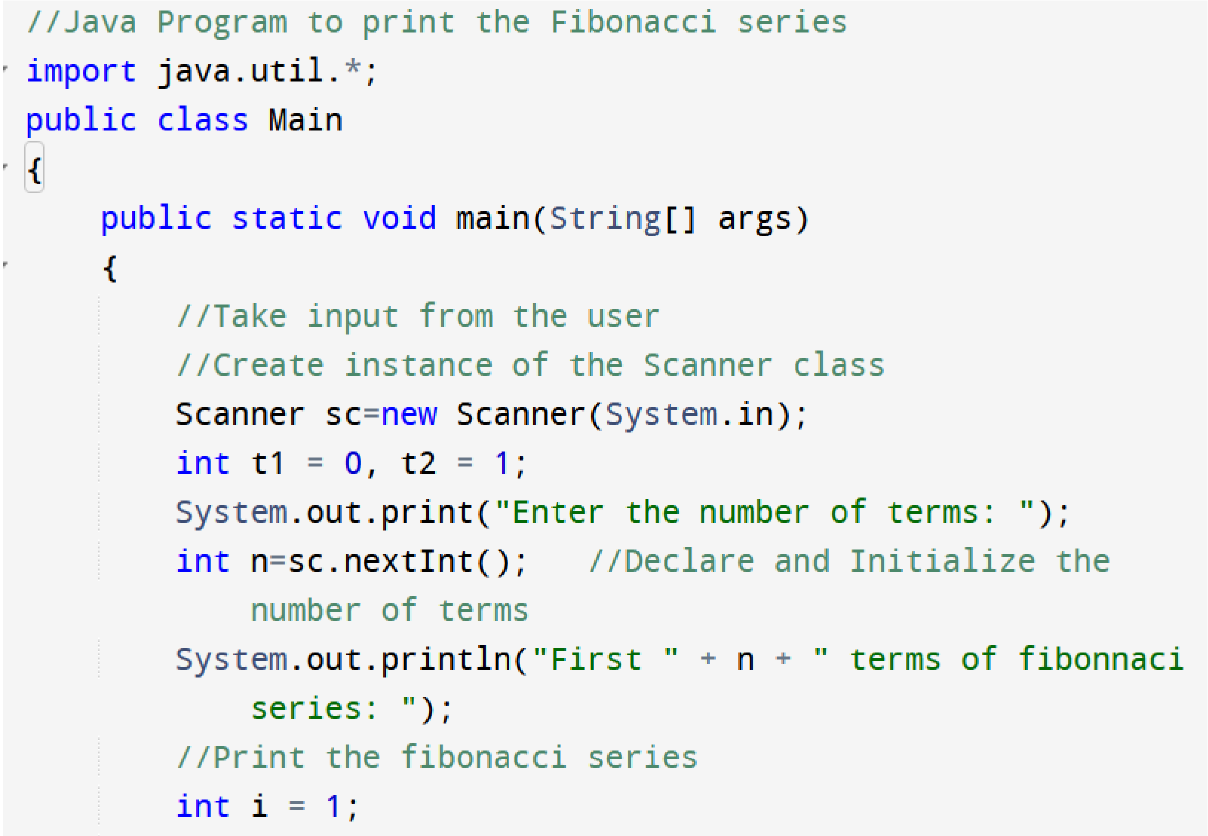
```
        //Print the fibonacci series
        for (int i = 1; i <= n; ++i)
        {
            System.out.print(t1 + " ");
            int sum = t1 + t2;
            t1 = t2;
            t2 = sum;
        }
    }
}
```

Program 6: To Print Fibonacci Series Using While Loop

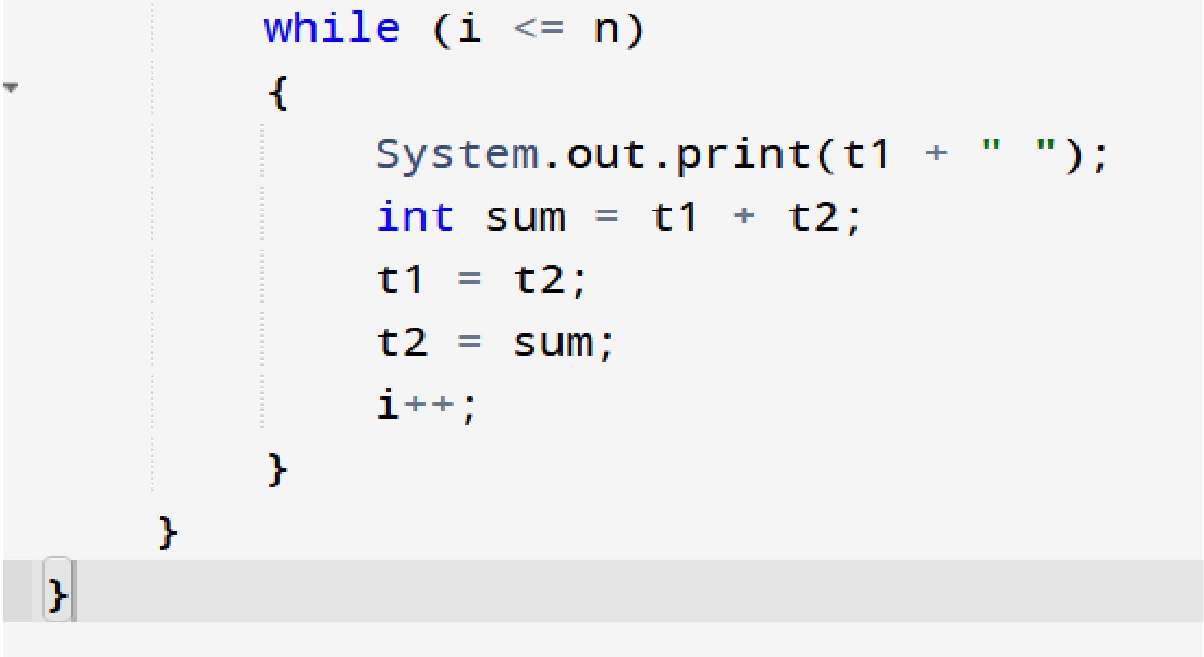
In this program, we will see how to print the Fibonacci Series in Java using a while loop. Here, firstly, we will ask the user to enter the number of terms and then we will find the Fibonacci Series.

Algorithm:

1. Start
2. Declare a variable for the total number of terms.
3. Ask the user to initialize the number of terms.
4. Print the first and second numbers of the series.
5. Use a while loop to print the Fibonacci series up to that number of terms.
6. Update the series terms in each iteration.
7. Print the Fibonacci series.
8. Stop



```
//Java Program to print the Fibonacci series
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        //Take input from the user
        //Create instance of the Scanner class
        Scanner sc=new Scanner(System.in);
        int t1 = 0, t2 = 1;
        System.out.print("Enter the number of terms: ");
        int n=sc.nextInt();   //Declare and Initialize the
            number of terms
        System.out.println("First " + n + " terms of fibonnaci
            series: ");
        //Print the fibonacci series
        int i = 1;
```



```
        while (i <= n)
        {
            System.out.print(t1 + " ");
            int sum = t1 + t2;
            t1 = t2;
            t2 = sum;
            i++;
        }
    }
}
```

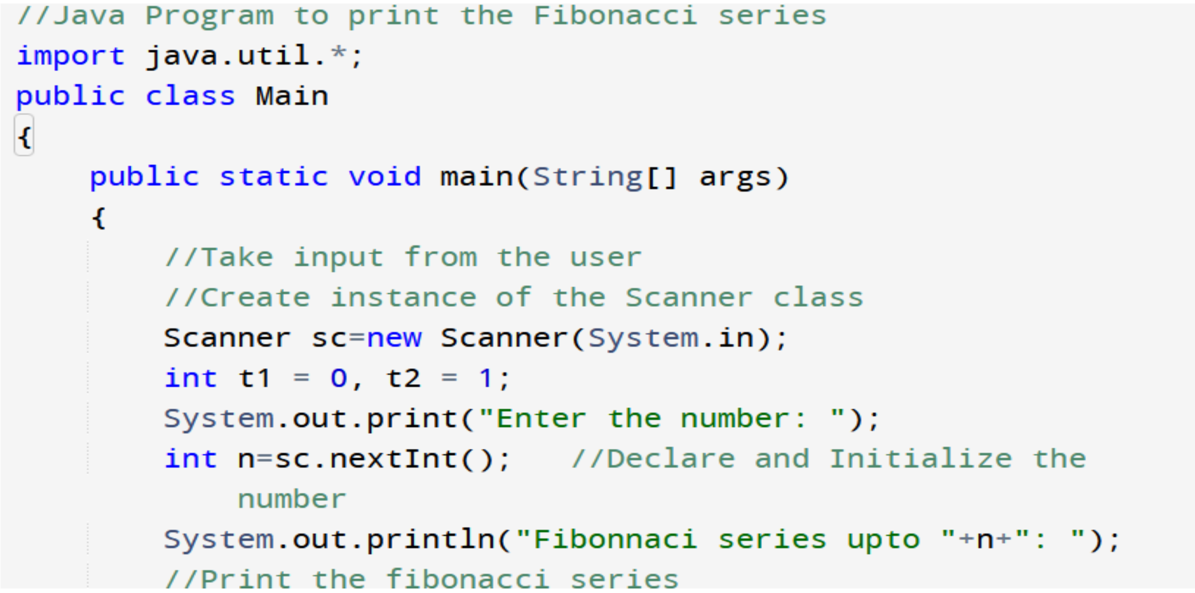
Program 7: To Print Fibonacci Series up to a Given Number

In this program, we will see how to print the Fibonacci Series in Java up to a given number. Here, firstly, we will ask the user to enter the number of terms and then we will find the Fibonacci Series up to that particular number.

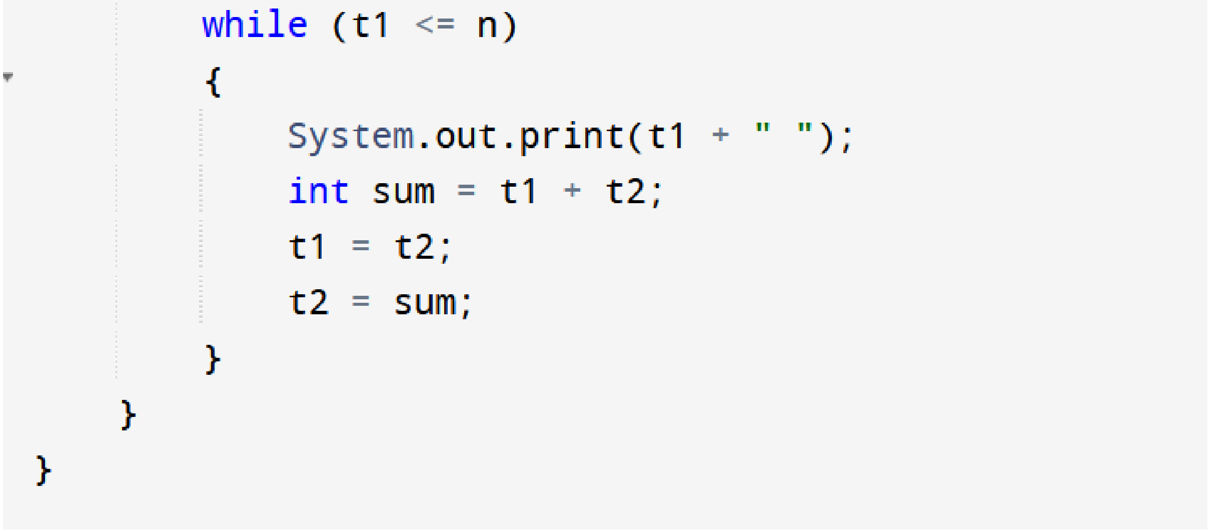
Algorithm:

1. Start

2. Declare a variable.
3. Ask the user to initialize the number.
4. Print the first and second numbers of the series.
5. Use a while loop to print the Fibonacci series up to that number.
6. Update the series terms in each iteration.
7. Print the Fibonacci series.
8. Stop



```
//Java Program to print the Fibonacci series
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        //Take input from the user
        //Create instance of the Scanner class
        Scanner sc=new Scanner(System.in);
        int t1 = 0, t2 = 1;
        System.out.print("Enter the number: ");
        int n=sc.nextInt();    //Declare and Initialize the number
        System.out.println("Fibonnaci series upto "+n+": ");
        //Print the fibonacci series
```



```
        while (t1 <= n)
        {
            System.out.print(t1 + " ");
            int sum = t1 + t2;
            t1 = t2;
            t2 = sum;
        }
    }
}
```

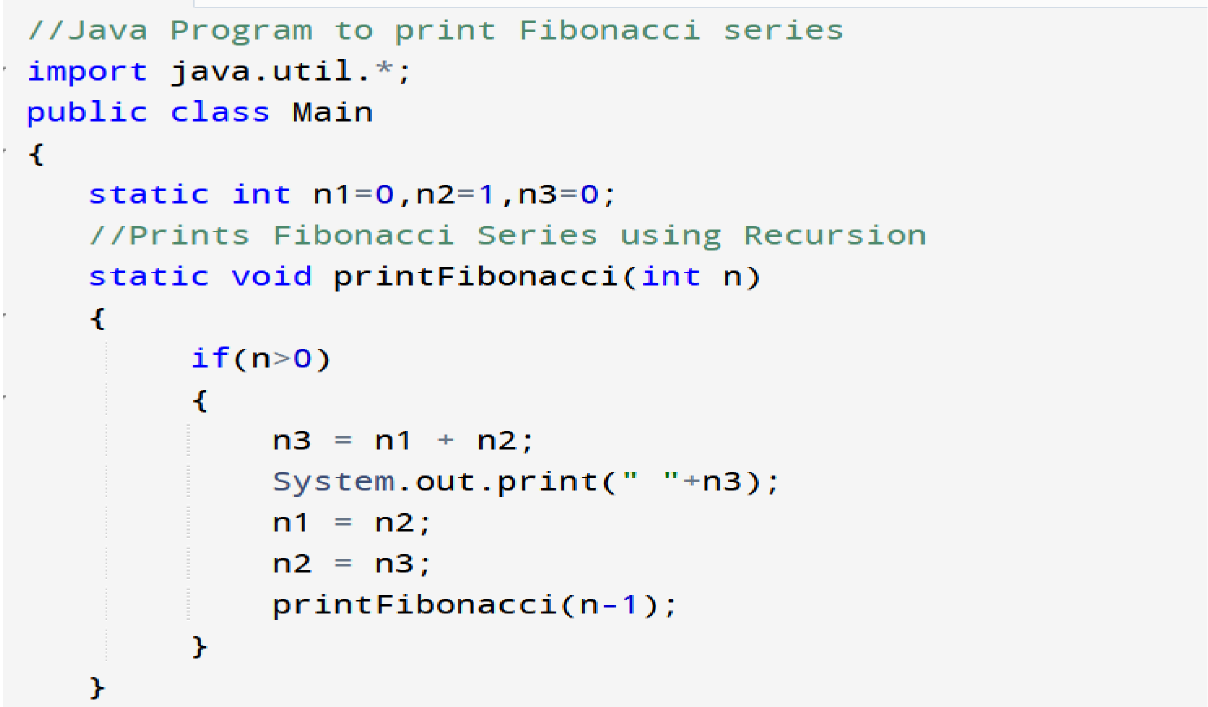
Program 8: To Print Fibonacci Series Using Recursion

In this program, we will see how to print the Fibonacci Series in Java using recursion. Here, firstly, we will ask the user to enter the number of terms and then we will find the Fibonacci Series.

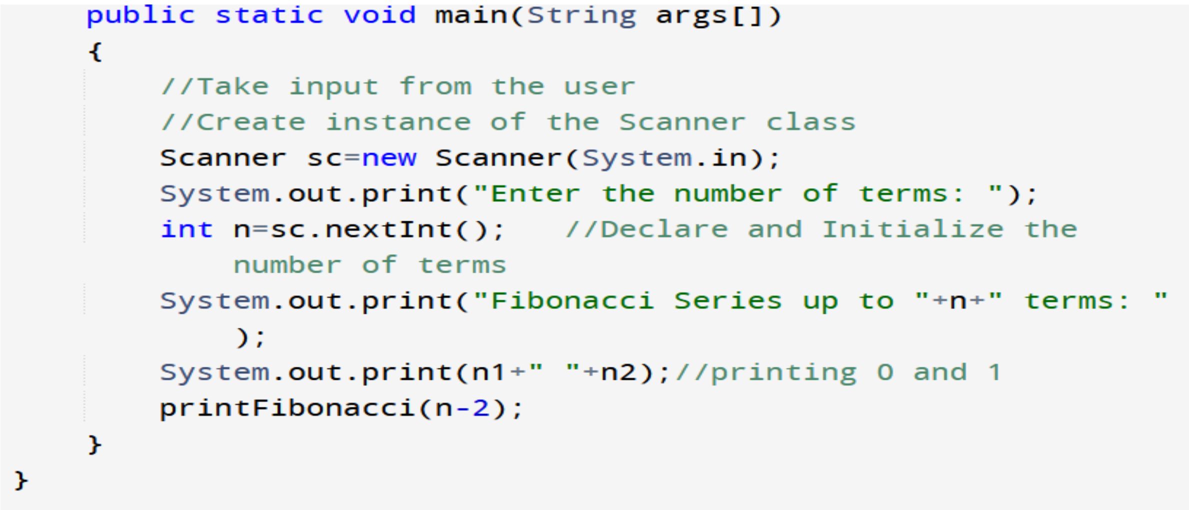
Algorithm:

1. Start

2. Declare a variable for the total number of terms.
3. Ask the user to initialize the number of terms.
4. Print the first and second numbers of the series.
5. Call a recursive function to print the Fibonacci series up to that number of terms.
6. Update the series terms recursively.
7. Print the Fibonacci series.
8. Stop



```
//Java Program to print Fibonacci series
import java.util.*;
public class Main
{
    static int n1=0,n2=1,n3=0;
    //Prints Fibonacci Series using Recursion
    static void printFibonacci(int n)
    {
        if(n>0)
        {
            n3 = n1 + n2;
            System.out.print(" "+n3);
            n1 = n2;
            n2 = n3;
            printFibonacci(n-1);
        }
    }
```



```
    public static void main(String args[])
    {
        //Take input from the user
        //Create instance of the Scanner class
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of terms: ");
        int n=sc.nextInt();    //Declare and Initialize the
            number of terms
        System.out.print("Fibonacci Series up to "+n+" terms: "
            );
        System.out.print(n1+" "+n2);//printing 0 and 1
        printFibonacci(n-2);
    }
}
```