Experiment 1

Java Programming Laboratory

Manas Patil 60019210046

Aim:

Write a program to implement Fibonacci Sequence using Recursive Functions and without using Recursive Functions

Theory:

Q 1. Define the following with suitable examples:

1. Variables

Variable in Java is a data container that saves the data values during Java program execution. Every variable is assigned a data type that designates the type and quantity of value it can hold. A variable is a memory location name for the data.

2. Data Types

Data types are different sizes and values that can be stored in the variable.

Java has two categories in which data types are segregated

1. **Primitive Data Type:** such as boolean, char, int, short, byte, long, float, and double
2. **Non-Primitive Data Type or Object Data type:** such as String, Array, etc.



1. Conditional Statements
2. **if:**if statement is the simplest decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not
3. **if-else**: The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won’t. We can use the else statement with the if statement to execute a block of code when the condition is false.
4. **nested-if:** A nested if is an if statement that is the target of another if or else. Nested if statements mean an if statement inside an if statement.
5. **if-else-if ladder:** Here, a user can decide among multiple options. The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that ‘if’ is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.
6. **switch-case:** The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.
7. Loops and their types
8. while loop: A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

while (condition)

{

statements..

}

1. do-while loop: do while loop is similar to while loop with only difference that it checks for condition after executing the statements

do

{

statements..

}

while (condition);

1. for loop: for loop provides a concise way of writing the loop structure.

for (initialization condition; testing condition;increment/decrement)

{

statement(s)

}

1. Type-casting

Convert a value from one data type to another data type is known as type casting.

There are two types of type casting:

1. Widening Type Casting- Converting a lower data type into a higher one is called **widening** type casting. It is also known as **implicit conversion** or **casting down**. It is done automatically. It is safe because there is no chance to lose data.

E.g. **int** x = 7;

**long** y = x;

**float** z = y;

1. Narrowing Type Casting - Converting a higher data type into a lower one is called **narrowing** type casting. It is also known as **explicit conversion** or **casting up**. It is done manually by the programmer. If we do not perform casting, then the compiler reports a compile-time error.

E.g. **double** d = 166.66;

**long** l = (**long**)d;

**int** i = (**int**)l;

Q 2. What are recursive functions?

A recursive function is a function that calls itself. It is a function that calls itself until it reaches a base case. A base case is a condition that will stop the function from calling itself.

Q 3. Where are recursive functions used in industry?

Recursive functions are widely applicable in mathematics and computer science, in algorithms such as tree traversal, sorting algorithms. The Fibonacci numbers are an example of a sequence defined recursively.

Q4. Advantages of Recursive Functions

Code:

//This program will calculate the Fibonacci sequence of a number and takes input by BufferedReader using recursion and without recursion.

import java.io.\*;

class Main{

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter a number: ");

int num = Integer.parseInt(br.readLine());

int num2 = fibonacci(num);

System.out.println("The fibonacci sequence of the number is: " + num2);

int num3 = fibonacci2(num);

System.out.println("The fibonacci sequence of the number is: " + num3);

}

public static int fibonacci(int num) {

if(num==1 || num==2)

{

return 1; //The base case

}

else{

return fibonacci(num-1)+fibonacci(num-2); //This is the recursive call

}

}

public static int fibonacci2(int num) {

int num1 = 1;

int num2 = 1;

int num3 = 0;

for(int i=3; i<=num; i++)

{

num3 = num1+num2;

num1 = num2;

num2 = num3;

}

return num3;

}

}

Output:

