Input in Java

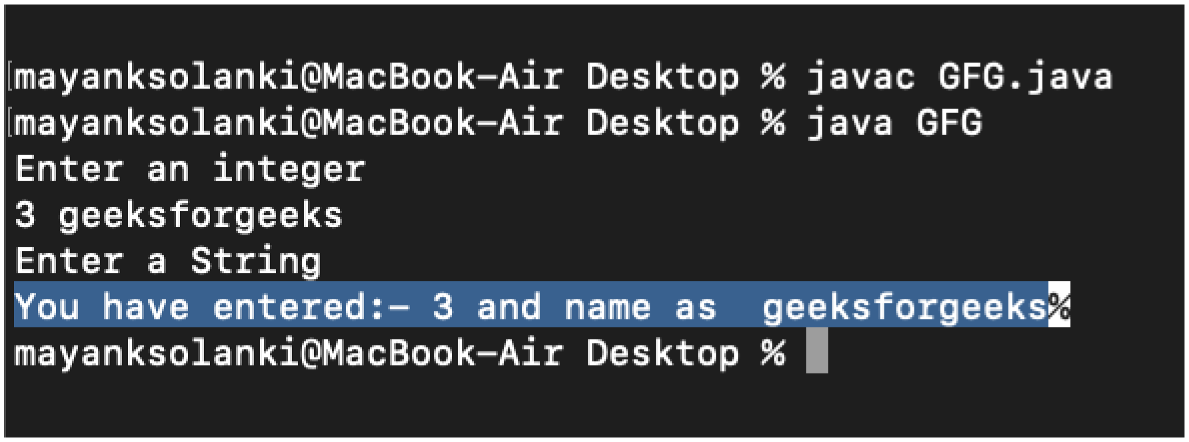
In Java, Scanner and BufferedReader class are sources that serve as ways of reading inputs. Scanner class is a simple text scanner that can parse primitive types and strings. It internally uses regular expressions to read different types while on the other hand BufferedReader class reads text from a character-input stream, buffering characters so as to provide for the efficient reading of the sequence of characters

The eccentric difference lies in reading different ways of taking input via the next() method that is justified in the below programs over the similar input set.

Example 1: Scanner Class

```java
// Java Program to Illustrate Scanner Class

// Importing Scanner class from
// java.util package
import java.util.Scanner;

// Main class
class GfG {

    // Main driver method
    public static void main(String args[]) {

        // Creating object of Scanner class to
        // read input from keyboard
        Scanner scn = new Scanner(System.in);

        System.out.println("Enter an integer");

        // Using nextInt() to parse integer values
        int a = scn.nextInt();

        System.out.println("Enter a String");

        // Using nextLine() to parse string values
        String b = scn.nextLine();

        // Display name and age entered above
        System.out.printf("You have entered:- " + a + " "
                                  + "and name as " + b);
    }
}
```

Output:

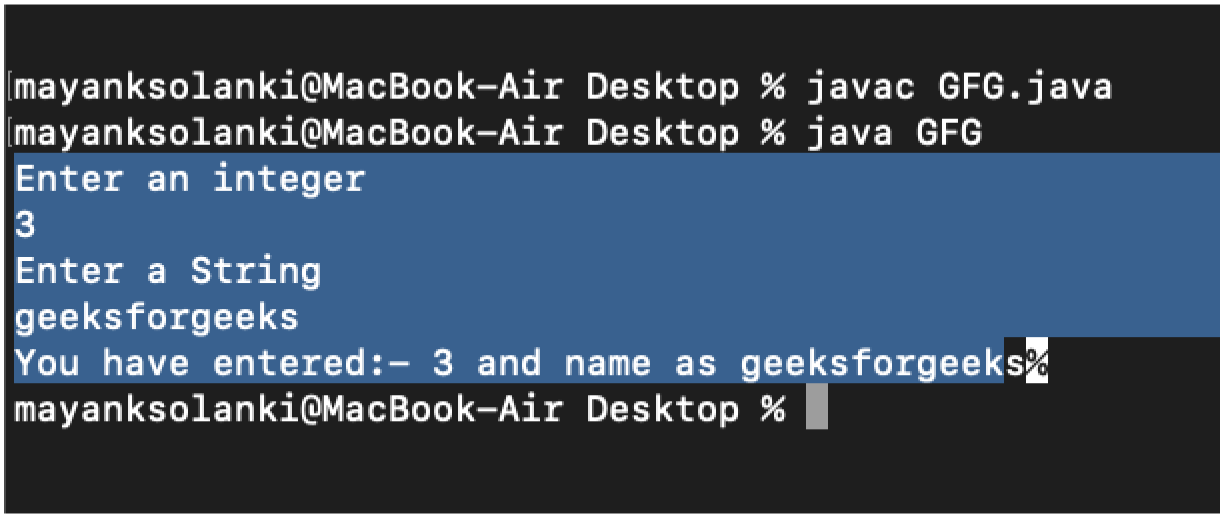


```
mayanksolanki@MacBook-Air Desktop % javac GFG.java
mayanksolanki@MacBook-Air Desktop % java GFG
Enter an integer
3 geeksforgeeks
Enter a String
You have entered:- 3 and name as  geeksforgeeks%
mayanksolanki@MacBook-Air Desktop %
```

Example 2: BufferedReader Class

```
// Java Program to Illustrate BufferedReader Class

// Importing required class
import java.io.*;

// Main class
class GfG {

    // Main driver method
    public static void main(String args[])
        throws IOException {

        // Creating object of class inside main() method
        BufferedReader br = new BufferedReader(
            new InputStreamReader(System.in));

        System.out.println("Enter an integer");

        // Taking integer input
        int a = Integer.parseInt(br.readLine());
        System.out.println("Enter a String");

        String b = br.readLine();

        // Printing input entities above
        System.out.printf("You have entered:- " + a
                          + " and name as " + b);
    }
}
```

Output:



```
mayanksolanki@MacBook-Air Desktop % javac GFG.java
mayanksolanki@MacBook-Air Desktop % java GFG
Enter an integer
3
Enter a String
geeksforgeeks
You have entered:- 3 and name as geeksforgeeks%
mayanksolanki@MacBook-Air Desktop %
```

Output explanation:

- In Scanner class if we call nextLine() method after any one of the seven nextXXX() method then the nextLine() does not read values from console and cursor will not come into console it will skip that step. The nextXXX() methods are nextInt(), nextFloat(), nextByte(), nextShort(), nextDouble(), nextLong(), next().
- In BufferReader class there is no such type of problem. This problem occurs only for the Scanner class, due to nextXXX() methods ignoring newline character and nextLine() only reads till the first newline character. If we use one more call of nextLine() method between nextXXX() and

nextLine(), then this problem will not occur because nextLine() will consume the newline character.

Tip: See this for the corrected program. This problem is same as scanf() followed by gets() in C/C++. This problem can also be solved by using next() instead of nextLine() for taking input of strings as shown here.

Major Differences between Scanner and BufferedReader Class in Java

- BufferedReader is synchronous while Scanner is not. BufferedReader should be used if we are working with multiple threads.
- BufferedReader has a significantly larger buffer memory than Scanner.
- The Scanner has a little buffer (1KB char buffer) as opposed to the BufferedReader (8KB byte buffer), but it's more than enough.
- BufferedReader is a bit faster as compared to scanner because the scanner does the parsing of input data and BufferedReader simply reads a sequence of characters.