CIS 36B – 9th Class / Lab Assignment – **10 Points-**
Student Name kachilau Student ID 10819338 Point Total

Section 1: Define / Answer

recursion- Recursion is a basic programming technique you can use in Java, in which a method calls itself to solve some problem. A method that uses this technique is recursive. Many programming problems can be solved only by recursion, and some problems that can be solved by other techniques are better solved by recursion.

Base Case: The case in which we end our recursion is called a base case

CIS 36B – 9th Class / Lab Assignment – **10 Points-**
Student Name kachilau Student ID 10819338 Point Total

Task 1:

The fibonacci sequence is a famous bit of mathematics, and it happens to have a recursive definition. The first two values in the sequence are 0 and 1 (essentially 2 base cases). Each subsequent value is the sum of the previous two values, so the whole sequence is: 0, 1, 1, 2, 3, 5, 8, 13, 21 and so on. Define a recursive fibonacci(n) method that returns the nth fibonacci number, with n=0 representing the start of the sequence.

Allow the user to input n.

ex. n = 8

0, 1, 1, 2, 3, 5, 8, 13

ex. n = 9

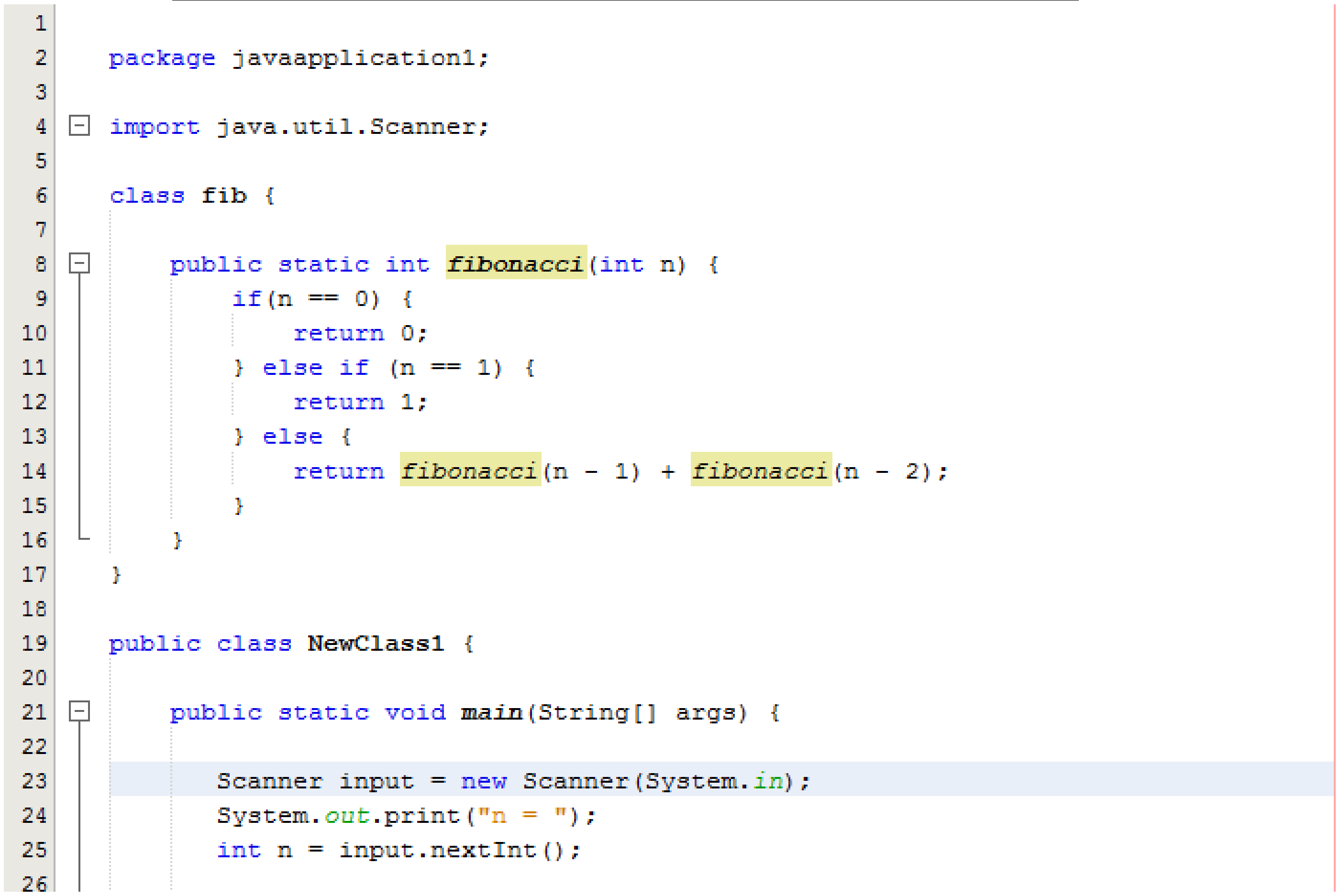
0, 1, 1, 2, 3, 5, 8, 13, 21

Use return method

Need “programmer created class structure”

Attach Snipping photos of source code and output.

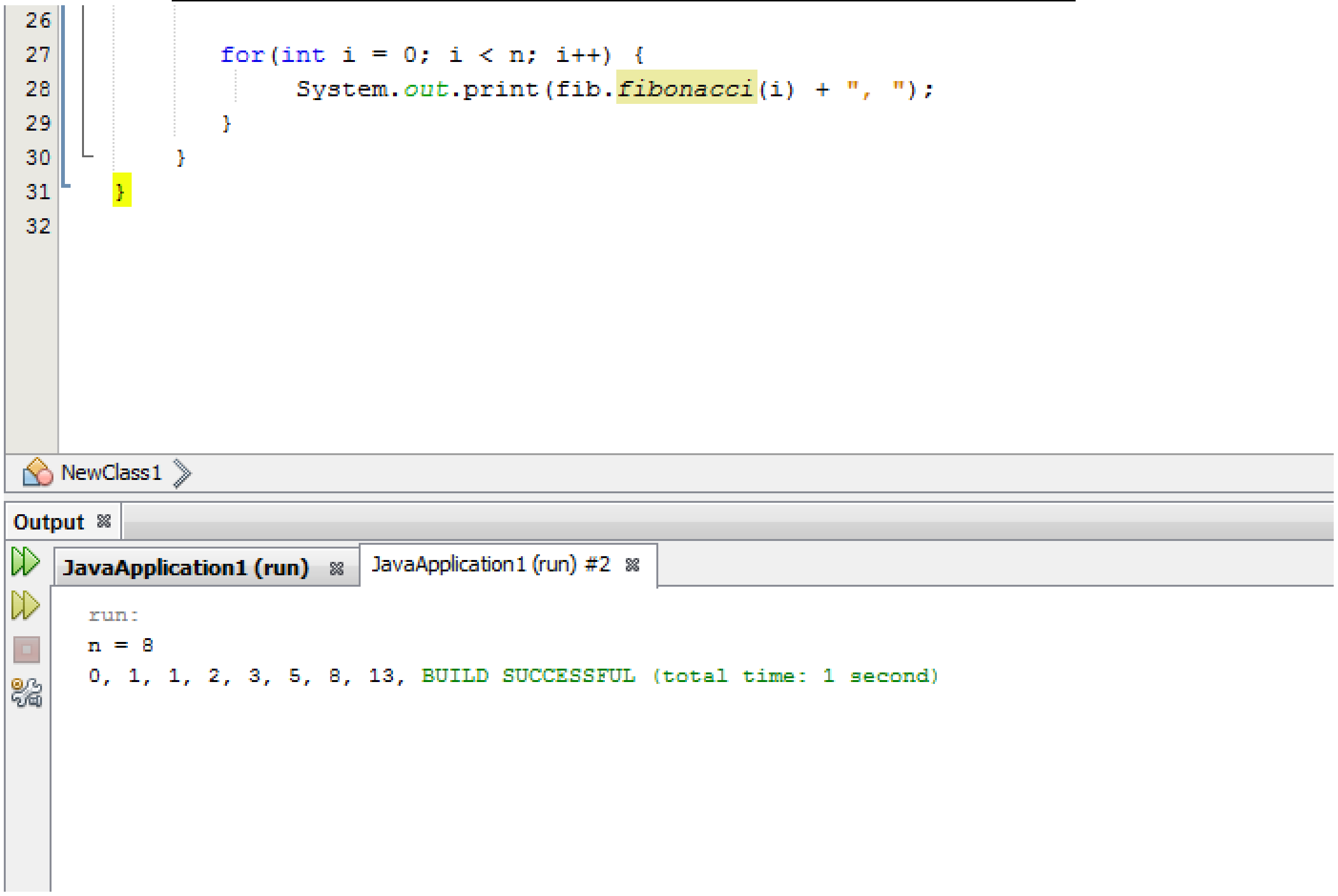
CIS 36B – 9th Class / Lab Assignment – **10 Points-**
Student Name kachilau **Student ID 10819338** **Point Total**



```java

package javaapplication1;

import java.util.Scanner;

class fib {

    public static int fibonacci(int n) {
        if(n == 0) {
            return 0;
        } else if (n == 1) {
            return 1;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }
}

public class NewClass1 {

    public static void main(String[] args) {

       Scanner input = new Scanner(System.in);
       System.out.print("n = ");
       int n = input.nextInt();

```

CIS 36B – 9th Class / Lab Assignment – **10 Points-**

Student Name kachilau **Student ID** 10819338 **Point Total**



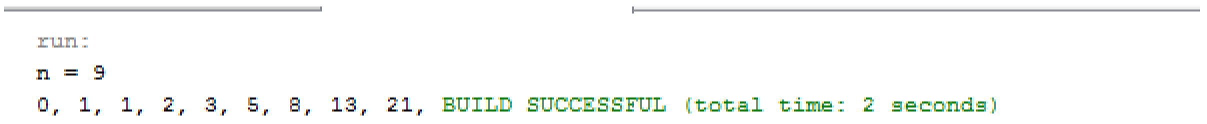
```
        for(int i = 0; i < n; i++) {
            System.out.print(fib.fibonacci(i) + ", ");
        }
    }
}
```

NewClass1

Output

JavaApplication1 (run) | JavaApplication1 (run) #2

```
run:
n = 8
0, 1, 1, 2, 3, 5, 8, 13, BUILD SUCCESSFUL (total time: 1 second)
```

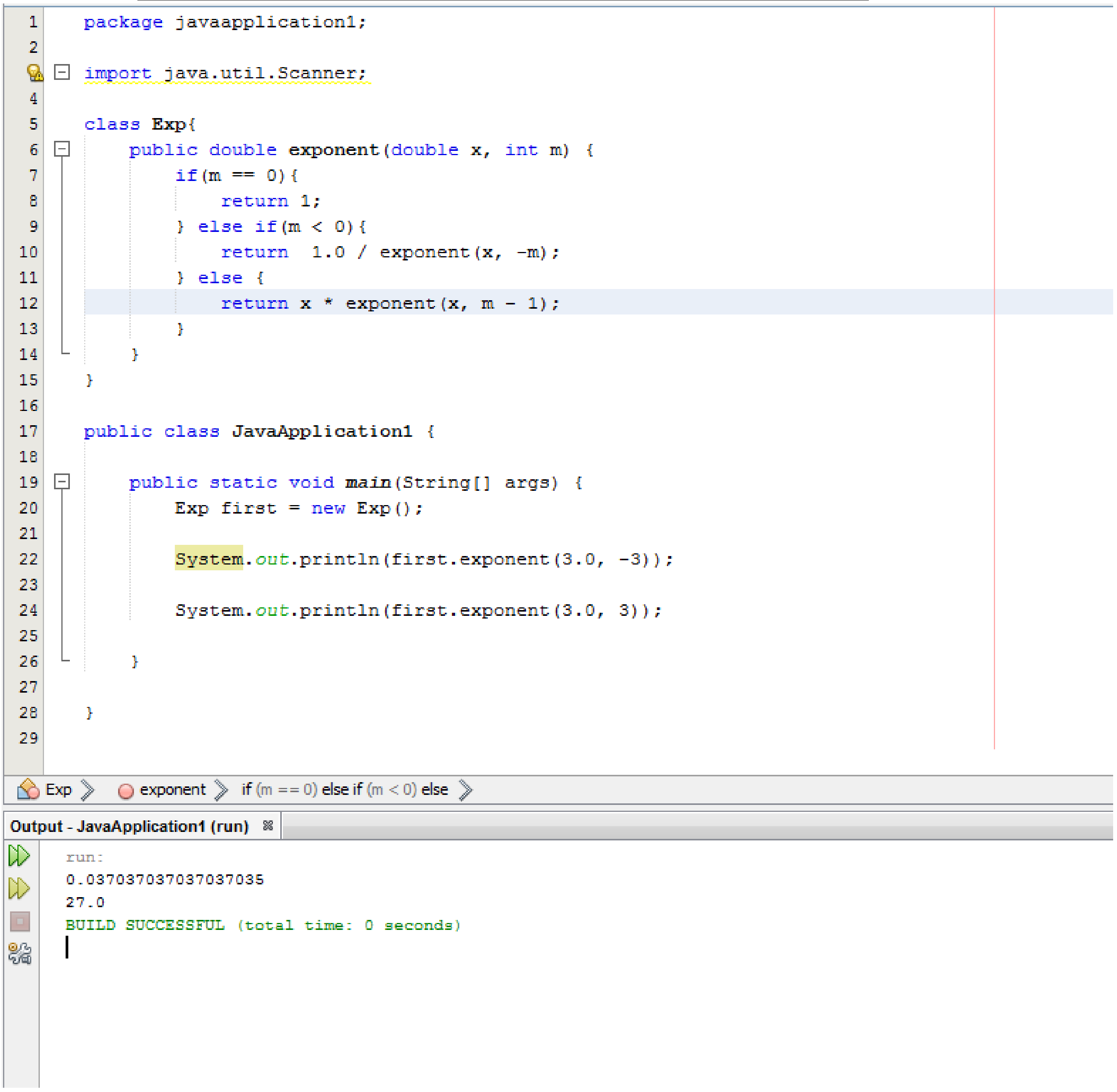


```
run:
n = 9
0, 1, 1, 2, 3, 5, 8, 13, 21, BUILD SUCCESSFUL (total time: 2 seconds)
```

Task 2: Write a recursive function to perform exponentiation.

Return X^m, assuming m >=0

CIS 36B – 9th Class / Lab Assignment – **10 Points-**
Student Name kachilau **Student ID 10819338** **Point Total**



```
package javaapplication1;

import java.util.Scanner;

class Exp{
    public double exponent(double x, int m) {
        if(m == 0){
            return 1;
        } else if(m < 0){
            return  1.0 / exponent(x, -m);
        } else {
            return x * exponent(x, m - 1);
        }
    }
}

public class JavaApplication1 {

    public static void main(String[] args) {
        Exp first = new Exp();

        System.out.println(first.exponent(3.0, -3));

        System.out.println(first.exponent(3.0, 3));

    }

}
```

Exp
exponent
if (m == 0) else if (m < 0) else
Output - JavaApplication1 (run)

```
run:
0.037037037037037035
27.0
BUILD SUCCESSFUL (total time: 0 seconds)
```