

Pg. 228 – 230 Java Programming *A comprehensive Introduction*

Section 1: Define

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

Task1:

Pg. 252 Java Programming *A comprehensive Introduction*

Write a recursive method `countDown()` that takes an integer **n** as its parameter. It prints the integers from **n** down to 0, one per line, and then prints “Blast off”

```
1
2  package javaapplication1;
3
4  import java.util.Scanner;
5
6  public class NewClass1 {
7
8      public static int countdown(int n) {
9
10         if (n == 0) {
11             System.out.println(n);
12             System.out.println("Blast Off");
13             return 0;
14         } else {
15             System.out.println(n);
16             return countdown(n - 1);
17         }
18     }
19
20     public static void main(String[] args) {
21
22         Scanner input = new Scanner(System.in);
23
24         System.out.print("Enter an integer: ");
25         int a = input.nextInt();
26
27         countdown(a);
28     }
29 }
30
```

```
run:
Enter an integer: 17
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1|
0
Blast Off
BUILD SUCCESSFUL (total time: 5 seconds)
```

Task 2:

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.

```
1
2 package javaapplication1;
3
4 ☐ import java.util.Scanner;
5
6 class fib {
7
8 ☐     public static int fibonacci(int n) {
9         if(n == 0) {
10             return 0;
11         } else if (n == 1) {
12             return 1;
13         } else {
14             return fibonacci(n - 1) + fibonacci(n - 2);
15         }
16     }
17 }
18
19 public class NewClass1 {
20
21 ☐     public static void main(String[] args) {
22
23         Scanner input = new Scanner(System.in);
24         System.out.print("n = ");
25         int n = input.nextInt();
26     }
```

```
26
27     for(int i = 0; i < n; i++) {
28         System.out.print(fib.fibonacci(i) + ", ");
29     }
30 }
31 }
32
```

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)