**WIA1002 TUTORIAL 9**

**Recursion (Fundamental)**

1. Explain the problem that occurs when executing the recursive method f(0).

public static int f(int n) {

if (n == 1)

return n;

else

return n \* f(n-1);

}

ANSWER:

The recursive method runs infinitely because the value of n will always be lower than the base case of n == 1, so it will cause the program's stack to continuously grow until it reaches a particular height and the StackOverflowError exception is initiated which crashes the program.

2. Explain the problem that occurs when executing the recursive method f().

public static int f(int n) {

if (n == 0)

return n;

else

return f(n+1) + n;

}

ANSWER:

The problem occurs is the compile-time error. This happens because the method f() requires an integer parameter n but the function call does not pass in an integer argument. This error happens when the compiler expects the f() method call to accept exactly one integer parameter but the argument is not provided.

*When the arugment passes into the method is non-negative (n >0), the stopping condition (n==0) will never be reached. f() becomes an infinite recursive method.*

3. Write a recursive method to reverse a string.

String -> gnirts

ANSWER:

public static String reverse(String string, int charPosition) {

// Base Case

if (charPosition == string.length() - 1) return "" + string.charAt(charPosition);

// Recursion

return reverse(string, charPosition + 1) + string.charAt(charPosition);

}

*public static String reverse(String str){*

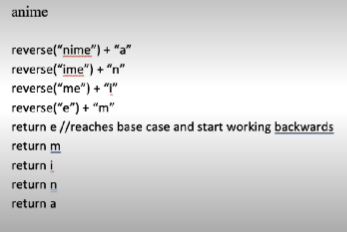
*if(null == str) || (str.length() <<= 1)){*

*return str;*

*}*

*Return reverse(str.substring(1)) + str.charAt(0);*

*}*

**

4. Write a recursive method to calculate the following :

5 + 4 + 3 + 2 + 1.

State the base case and recursive case. Trace your solution using number, N of 5.

Algorithm :

1. Base case = 1

2. Recursive case = n + sum(n-1)

ANSWER:

public static int sum(int n) {

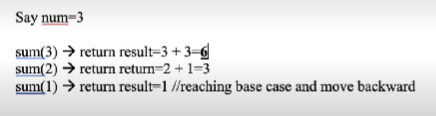
// Base Case

if (n == 1) return 1;

// Recursion

return n + sum(n - 1);

}



5. Write a recursive method printDigit that prints an integer argument as its constituent digits, with a blank space separates each digit with the next. For example, if the argument is 4567, this method will print 4 5 6 7 on the screen.

ANSWER:

public static void printDigit(int n) {

// Base Case

if (n == 0) return;

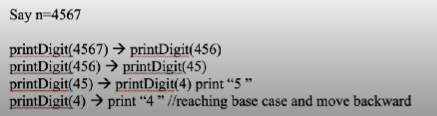
// Recursion

int divisor = (int) Math.pow(10, (int) Math.log10(n));

System.out.printf("%d ", n / divisor);

printDigit(n % divisor);

}



Print “4 5 6 7”