



COMP 249: Object Oriented Programming II

Tutorial 8: Recursion

Question 1

What is Recursion in Java?

- a) Recursion is a class.
- a) Recursion is a process of defining a method that calls other methods repeatedly.
- a) Recursion is a process of defining a method that calls itself repeatedly.
- b) Recursion is a process of defining a method that calls other methods which in turn call the original method.

Question 2

Which of these things will happen if a recursive method does not have a base case?

- a) An infinite loop.
- b) System stops the program after a certain time.
- c) The program automatically stops after 1000000 calls.
- d) None of the above.

Question 3

Which of these data types is used by the operating system to manage recursion in Java?

- a) Array
- b) Stack
- c) Queue
- d) Tree

Question 4

A program to find **Mod** using recursion.

```
public static void main(String args[])
{
    System.out.println(mod(25,7));
}

public static int mod( int n, int m ){
    System.out.println("n: " + n);
    if ( n < m )
        return n;
    else
        return mod( n-m, m );
}
```

OUTPUT ?

Question 5

Consider the following code:

```
public static void printtest(int n)
{
    if (n > 0 )
    {
        printtest(n-1);
        System.out.println("A: "+ n);
        printtest(n -1);
    }
    else
    {
        System.out.println("B: "+ n);
    }
}

public static void main(String[] args) {
    printtest(2);
}
```

OUTPUT ?

Coding Exercise 1

Write a recursive function named **reverse (String s)** which returns the reverse of a String s

Example:

Input: Wonderful Day
Output: yaD lufrednoW

Coding Exercise 2

Write a recursive method called `convertToBinary(int n)`, that takes an integer and returns the binary representation of the number as a String.

Example:

Input: 14

Output: 1110

Coding Exercise 3

Write a program that takes a String and print all the possible permutations of the String.

Examples:

Input: ab

Output: ab ba

Input: abc

Output: abc acb bac bca cab cba

Input: COIN

Output: COIN CONI CION CINO CNOI CNIO OCIN OCNI OICN OINC
ONCI ONIC ICON ICNO IOCN IONC INCO INOC NCOI NCIO NOCI
NOIC NICO NIOC

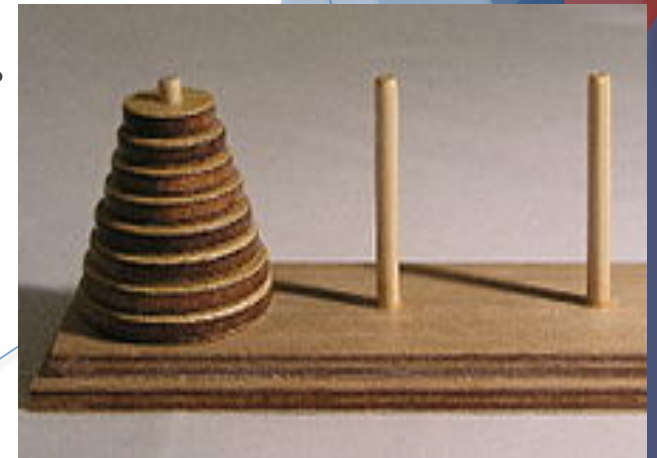
Coding Exercise 4

The Tower of Hanoi is a popular problem. There are three poles and n disks. The disks fit on the poles and are all different sizes. They are stacked from largest to smallest on pole 1, with the smallest on top.

The task is to move all the disks from pole 1 to pole 3 with following restrictions:

- ❖ Only one disk can be moved at a time
- ❖ A larger disk can not be placed on a smaller disk.

[Link to an interactive Tower of Hanoi](#)



Coding Exercise 4 (contd..)

The recursive algorithm works as follows:

- ❖ move $n-1$ disk from the starting pole to the pole which is not the target pole (intermediate).
- ❖ move disk n to the target pole.
- ❖ move $n-1$ disk from the intermediate pole to the target pole.
- ❖ The $n-1$ disks are moved recursively.

Write a recursive solution to the Tower of Hanoi problem.