CSE 2001: Data Structure & Algorithms

Programming Assignment-III

(Exception, Generics & Recursion)

1. Write a Java program to read your lucky number from keyboard. Treat –ve no. as *NumberFormatException*. Write appropriate Exceptional handler.
2. Assign your favorite colors in an array. Identify 2 exceptions that may be generated & write exceptional handler in Java.
3. Create a class Student & enter mark, name of the student. If mark is more than 100, create exception *MarksOutOfBoundException* & throw it using Java.
4. Write a simple main class in Java that contains an experiment that uses the generic `Box<T>` class to build boxes with different types and that verifies that this class works as advertised. Your experiment should include the following:

   - Create a boxed String object and two variables that refer to that box. Change the contents of one and determine the effect on the other.
   - Create a boxed Integer object and two variables that refer to that box. Change the contents of one and determine the effect on the other.
   - Create a boxed Object object and two variables that refer to that box. Determine what happens if you put a string in the box. Determine what happens if you put an integer in the box.

5. Write a java program to print an array of different type using a single Generic method. The signature of *printArray* method is given below.

   ```
   public static < E > void printArray( E[] inputArray)
   ```

6. Write a java method using Generics to count the occurrence of an element in an array of any type. The signature of *count* method is given below.

   ```
   public static <T> int count(T[] array, T item)
   ```

7. Write a recursive method in Java that computes the factorial of a given integer.

8 Write a recursive method in Java which, given real value *x* and a positive integer *n*, returns the value of x^n.

9. Write a recursive method in Java which, given an integer n, print it with its digits reversed. For example , given `4735`, it prints `5374`.

10. The sequence of numbers `1, 1, 2, 3, 5, 8, 13` etc are called Fibonacci numbers, each is the sum of the preceding two. Write a recursive method in Java which, given *n*, returns the n^{th} Fibonacci number.

Home Work

1. Write a recursive method in Java to return the greatest common divisor(*gcd*) of two integers m and n, given that in general, *`gcd(m,n)=gcd(n, m mod n).`*
2. Write a recursive method in Java to search an element of an array using *binary search.*
3. Write a recursive method in Java to find the binary equivalent of a positive decimal integer.
4. Write a recursive method in Java to find the product of 2 numbers.
5. Write a recursive Java method that takes a character string s and outputs its reverse. For example, the reverse of `'pots&pans'` would be `'snap&stop'`.
6. Write a recursive Java method that determines if a string s is a palindrome, that is, it is equal to its reverse. Examples of palindromes include `'racecar'` and `'gohangasalamiimalasagnahog'`.
7. Given an unsorted array, *A*, of integers and an integer *k*, write recursive program using Java for rearranging the elements in *A* so that all elements less than or equal to *k* come before any elements larger than *k*.
8. In the Towers of Hanoi puzzle, we are given a platform with three pegs, *a*, *b*, and *c*, sticking out of it. On peg *a* is a stack of *n* disks, each larger than the next, so that the smallest is on the top and the largest is on the bottom. The puzzle is to move all the disks from peg *a* to peg *c*, moving one disk at a time, so that we never place a larger disk on top of a smaller one. See Figure 1 for an example of the case *n = 4*. Write a recursive program using Java for solving the Towers of Hanoi puzzle for arbitrary *n*. (Hint: Consider first the subproblem of moving all but the *nth* disk from peg *a* to another peg using the third as “temporary storage.”)

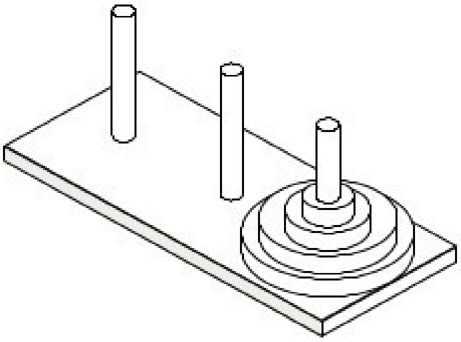

Figure 1: An illustration of the Towers of Hanoi puzzle.

****