CURTIN UNIVERSITY (CRICOS number: 00301J)

Department of Computing, Faculty of Engineering and Science

Data Structures and Algorithms (DSA1002)

PRACTICAL 2 - RECURSION

AIMS

- To practice recursion
- To add validation and exception handling
- To be able to trace through a recursive algorithm
- To implement Towers of Hanoi

BEFORE THE PRACTICAL:

- Read this practical sheet fully before starting.

ACTIVITY 1: FACTORIAL AND FIBONACCI

Given the code in the lecture slides, implement functions for factorial and Fibonacci.

Investigate their performance by running them with increasing values for n. How high can you go?

ACTIVITY 2: GREATEST COMMON DEMONINATOR

Search online (or make your own) recursive algorithm for finding the greatest common denominator. Implement it in Java or Python

ACTIVITY 3: NUMBER CONVERSIONS

Search online (or make your own) recursive algorithm for converting decimal to any base (up to 16). Implement it in Java or Python

ACTIVITY 4: WRAPPERS AND EXCEPTIONS

Update you code from activities 1-3 to use Wrappers and Exceptions to validate and respond to bad input data.

ACTIVITY 5: TOWERS OF HANOI IMPLEMENTATION

- Implement the Towers of Hanoi algorithm as a java/python method.
- Now write a main method to test this. Make sure you can enter the number of disks.
- You will also need to implement the moveDisk(src, dest) method. This can be as simple as an output statement printing "Moving top disk from peg source to peg destination" where source and destination are 1, 2 or 3.
- Add indenting to your output indicate the level of recursion. Some sample output is given below (hint – you can pass a string of spaces, or a number to keep track of recursion level)

```
towers(5, 1, 3)
       n=5, src=1, dest=3, temp=2
(3)    towers(4, 1, 2)
           n=4, src=1, dest=2, temp=3
(3)        towers(3, 1, 3)
               n=3, src=1, dest=3, temp=2
(3)            towers(2, 1, 2)
                   n=2, src=1, dest=2, temp=3
 (3)               towers(1, 1, 3)
```

MARKING GUIDE

Your submission will be marked as follows:

- [2] You have implemented Factorial and Fibonacci correctly and can discuss their performance
- [2] You have implemented Greatest Common Denominator correctly and can discuss its performance
- [2] You have implemented Number Conversions correctly and can discuss its performance
- [2] You have added Wrappers and Exceptions to all of your programs.
- [2] Your Towers of Hanoi program is implemented correctly.

SUBMISSION DELIVERABLE:

Your classes (all that are required for this program) are due before the beginning of your next tutorial. Also include any other relevant classes that you may have created. **SUBMIT ELECTRONICALLY VIA MOODLE**, under the *Assessments* section.