CMPT 120

Topic: Recursion – Part 1
Last Lectures

• Developing Software that incorporates functions
  • Way 1 – the program does not exist yet
  • Way 2 – the program has already been written
• Illustrated using a Case study
  • In the process, we pointed out a few guidelines:
    • Decomposition
    • Function Interface Design
    • Generalization
    • Composition
    • Encapsulation
    • And incremental development
Learning outcomes

At the end of this course, a student is expected to:

• Describe the concept of recursion, of recursive definitions and recursive functions.

• Use recursion to solve problems:
  • Use box tracing to predict the result of simple recursive code (drawing a call stack)
  • Design recursive code
  • Design recursive functions that recurse, for example, on lists and strings
Today’s Menu

• Recursion
  • Examples of recursion occurring in the real world
  • Examples of recursion occurring in the mathematical world
  • What is recursion
  • Iterative code versus Recursive code
  • Demo of recursion
Recursion in the real world

• Russian dolls
Recursion in the real world

- Searching for the word “guffaw” in a dictionary

Source: http://www.eslstation.net/ESL310L/310L_dict.htm
Recursion in the real world

• **Droste Effect**
  • Posted on our course web site
Recursion in the mathematical world

• Multiply two numbers
Recursion in the mathematical world

- Compute factorials
Recursion - Definition

- **Recursion** occurs when an object or a process is defined in terms of itself (or a version of itself)
- In mathematics and computer science, a kind of objects or processes exhibit **recursive** behavior when they can be defined by two properties:
  1. A simple **base case** (or cases)—a terminating scenario that does not use recursion to produce an answer
  2. A set of rules that reduce all other cases toward the base case

Adapted from http://en.wikipedia.org/wiki/Recursion
Definition of recursion applied to Russian dolls

1. A simple **base case** (or cases)—a terminating scenario that does not use recursion to produce an answer

2. A set of rules that reduce all other cases toward the base case
Definition of recursion applied to multiplying two #’s

1. A simple **base case** (or cases)—a terminating scenario that does not use recursion to produce an answer

2. A set of rules that reduce all other cases toward the base case
Definition of recursion applied to factorials

1. A simple **base case** (or cases)—a terminating scenario that does not use recursion to produce an answer
   - 

2. A set of rules that reduce all other cases toward the base case
   - 

Let’s now have a look at recursion in the computer/software world
So far -> iterative statements

- So far, when solving problems, we have achieved repetition (i.e., repeating statements in our code) by using **iterative** statements -> loops
  - By putting statements we wanted to repeat in a loop
Another way of achieving repetition (i.e., repeating statements in our code) is by putting statements we want to repeat in a function and calling the function itself a certain number of times.

Directly:

Indirectly:

Recursive functions
Recursion – Demo using the Visualizer
Example - recursive factorial
(slides with animation – to be seen in PowerPoint)

• Let’s apply what we have learnt to an example

5! = 5 * 4! * 3! * 2! * 1!

Base case

When the base case is reached, the execution “recurses back”
Summary

• Recursion
  • Examples of recursion occurring in the real world
  • Examples of recursion occurring in the mathematical world
  • What is recursion
  • Iterative code versus Recursive code
  • Demo of recursion
Next Lecture

• Recursion
  • How to solve a problem using recursion