

**SE2205: Algorithms and Data Structures for Object-Oriented Design**  
**Lab 1: Week of Jan 23, 2019**  
(Will **not** be marked)

## 1 Objectives

The main objective of this lab exercise is for you to practise implementing recursive functions in Java. There are two problems listed below.

### Problem 1

The Newton's method can be applied recursively to compute the approximate square root  $a$  of a number  $x$  that is precise up to  $\text{eps}$ . Implement the function `public double newtonSqrt(double x, double a, double eps)` according to the algorithm outlined next. The Newton's square root algorithm is as follows. Suppose that you would like to compute the square root of  $x$ . Set the first guess of the solution to be  $a=x/2$ . Compute  $|a * a - x|$ . Check if this is less than  $\text{eps}$ . If it is, then  $a^2$  is at most  $\pm \text{eps}$  of the square of the exact root. Otherwise, modify your current guess  $a$  to  $a = (a + x/a)/2$  and repeat the above algorithm until the accuracy criteria is met. Implement your function using **recursion**.

### Problem 2

In this question, you will implement the function `public int findMinChangeCoins(int C, int n)`. This function shall use a **recursive** technique to find the *minimum* number of coins needed to make a given amount of change. Suppose, that the currency system we are considering consists of coins with values  $S=\{1, 5, 10, 20, 25\}$ . Consider the specific scenario where we need to make a change for 40 cents. A greedy algorithm may suggest that the coins needed for making this change are 25 cents, 10 cents and 5 cents (3 coins). However, we only need 2 coins (two 20 cent coins).

Implement the function that computes the minimum number of coins needed to break a change by expanding the function `public int findMinChangeCoins(int C, int n)` where  $C$  is the change needed,  $n$  is  $n^{\text{th}}$  coin in the set  $S$  containing the denominations of the coins in increasing order (you should declare this array as a global entity). Assume that an infinite number of coins in all denominations are readily available. This function will return the total least number of coins needed to make the change.