# ­­Quadrus Developer Practicum

Thank you for your interest in becoming a Quadrus Developer. To ensure we hire the most skilled professionals we ask that you complete this practicum so that we can assess your development skills. We recommend that you read through the description thoroughly before starting and set aside a minimum of 4 hours to complete it.

## Intro

This practicum contains several coding problems. Please provide your code as well as instructions on how to execute the code. Please use one of the following languages to create your solution: C#, Java, JavaScript. You are free to use any built-in libraries that are part of the language you choose. You can make any reasonable assumptions, as long as these are documented in the comments or elsewhere.

## Problem #1 – Pip/Pop

Print out the numbers 1 to 100. For numbers divisible by 3, print out “Pip” instead of the number. For numbers divisible by 5, print out “Pop” instead of the number. Finally for numbers divisible by both 3 and 5, print out “PipPop” instead of the number.

## Problem #2 – Optimization

Implement a stack class that does not allow duplicate values to be added. This class will be used with millions of records and should therefore be efficient in how it checks for duplicates. The stack class should have 3 operations:

* Peek: should return the top value of the stack without removing it
* Push: should add an item to the top of the stack following the constraints outlined above
* Pop: should remove the top item from the stack

## Problem #3 – Data Structures

Create a tree class that has the following requirements:

* Each node has an integer value that is greater than 0
* The maximum number of children a node can have is equal to its value
* All nodes could have at least 1 child
* If a node’s value is updated such that the number of children it has is greater than the node’s new value, the node should move excess children lower in the tree hierarchy, creating new children if necessary. E.g. The root node’s value is changed from 2 to 1 causing one of its children to move lower on the hierarchy.

Chan

* In the above requirement, you can use any selection criteria when deciding which children are moved lower
* No other functionality is needed at this time

## Evaluation

You will be evaluated on how well you fulfill the requirements in each of the 3 problems. Consideration will be given for elegant or efficient solutions. Please provide a way to execute and validate your code e.g. console application, shell script, unit tests, etc. If you get stumped or are unable to complete a particular problem, submit what you have; in addition feel free to attach any pseudo code or anything else to explain how you might approach the problem.

We genuinely appreciate the effort that you put into this assignment and every care will be taken to evaluate it for the purposes of recruiting. However due to the volume of applications we receive, we are unable to provide individuals with direct feedback.