# Introduction

## Background

In software engineering, cloned code is one of the fundamental issues of productivity. The programming community is generally recognising the duplicates as a bad practice [Refactoring: Improving the Design of Existing Code by Martin Fowler], although some argue otherwise. We tend to avoid it because we are taught that repeated code is inelegant and can cause difficulties whilst trying to maintain applications, or sometimes it may involve legal risks. However, due to time pressure, or limitation of programming languages, reusing code fragments by duplicating with or without variations is a common activity evident within software development.

Unlike plagiarism detection in literature, duplicated code does not solely depend on text matching, but consideration has to be made for the syntax, semantics and patterns of the codes. For instance, two source files that functions the same may have completely different identifier names for variables – this must also be detected as a clone. Different approaches to achieve the detection exists, and will be discussed later.

## Project

In a 5 membered team, we are given the task of building a tool that analyses and reports similarity of two Java source files. This report introduces 5 different similarity detection mechanisms including Clone Detection and Plagiarism Detection, and we shall decide a suitable algorithm to implement for the tool. The tool will be tested with a given sample code and its variations to check our program’s performance whilst analysing the similarities between the pair of codes.