TASK 1

PMD provides several insights on some of the poorly implemented code snippets as discussed below.

CouplingBetweenObjects: High amount of different objects as members denotes a high coupling

ExcessiveClassLength: Avoid really long classes.

ExcessiveMethodLength: Avoid really long methods.

CyclomaticComplexity: The method 'Amount\_saved()' has a cyclomatic complexity of 11.

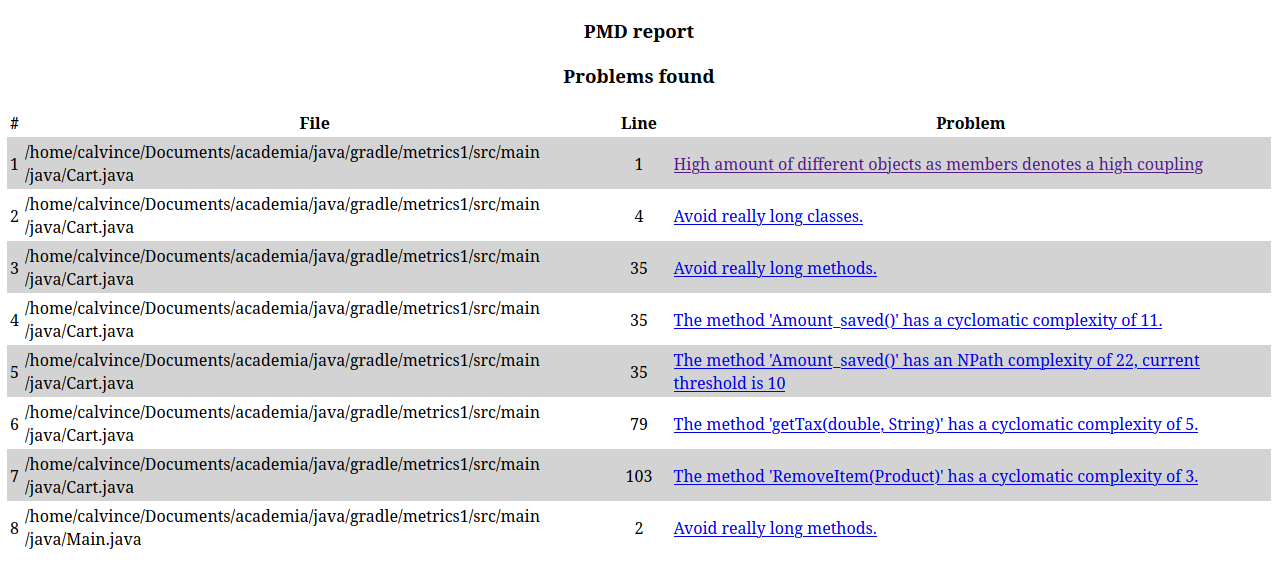
NPathComplexity: The method 'Amount\_saved()' has an NPath complexity of 22, current threshold is 10

CyclomaticComplexity: The method 'getTax(double, String)' has a cyclomatic complexity of 5.

CyclomaticComplexity: The method 'RemoveItem(Product)' has a cyclomatic complexity of 3.

ExcessiveMethodLength: Avoid really long methods.

The complexity of methods directly affects maintenance costs and readability. Concentrating too much decisional logic in a single method makes its behaviour hard to read and change. Cyclomatic complexity assesses the complexity of a method by counting the number of decision points in a method, plus one for the method entry. Decision points are places where the control flow jumps to another place in the program.



TASK 2