**Report – Assignment 1**

**Thilina Athukorala -21038875**

**(Object Oriented Software Engineering (COMP2003/6005) Semester 1, 2023)**

**Criteria 2:**

Containers are a core use in the programme since as soon as user gives input criteria to be searched, main method in App.java saves the include criteria and exclude criteria in an arrayList named as ‘allData which contains ‘User’ objects. After user selects 3rd option in the menu, these arraylists are passed to the relevant methods to return the container of ‘Line’ objects which matches the criteria provided by the user to FileEntry Object and saved in them as an arraylist of ‘Line’ Objects.

Hashmaps are not used in the above procedure due to disarrangement of data in them unlike the input order in arraylists. But hashmaps are being used to store the indentation levels generated by recursion since it’s easy(faster) to retrieve them by providing the key and the order does not matter.

**Criteria 3:**

App.java contains of 3 methods as mentioned in the UML diagram which handles all the operation in collecting matching lines according to criteria , creating indentation levels (updateDirectoryLevels()) and generating string according to each indentation levels(getIndentation ()).

In Directory class which implments DataEntry class, it provides the basic getters and setters for the fields which contain file related to directory and creation of tree of directories per each directory to include all of them as subdirectories inside the relevant directory using find() method. It calls the other implementation of DataEntry class (FileEntry) to create fileEntry objects whenever a file is found in the directory and add it to the tree to locate file objects at the proper location in the hierarchy.

FileEntry class calls the context of Operation class which is OperationContext and initiates the search method in each file which is passed to it when find() method is called by directory class.

In each class which implements Operation class, they contain operation() method which derives the algorithm to exclude and include lines according to criteria given by user.

**Criteria 4:**

Type of errors Triggered are most likely to be InputMismatchException, FileNotFoundException, IOException and PatternSyntaxException which have been handled successfully. In each core function of the system all logs are recorded in projectLog file which all records and errors are appended to the file whenever required.

**Criteria 5:**

**Strategy   Pattern**

Strategy pattern was used in triggering algorithms related to including and excluding criteria functions given by the user, which is the core usage of this programme (find a string/regex in a file present in hierarchy of directories). Context of ‘Operation’ class handles when ever a file needs to be scanned for inclusion criteria or exclusion criteria and returns the list of Line Objects containing line number and the String of line. It is clear that Strategy pattern is used in one of the core functions to be executed in the programme and is handled via the common interface instead of separate classes containing felids and methods to handle accordingly. Since inclusion and exclusion contains a similar type of operation, it was more appropriate to use the strategy pattern in this programme.

**Criteria 6:**

**Composite   Pattern**

Composite pattern is used in one of the core functions in the programme which is determining the headachy of directories and adding the relevant files with data by maintaining a tree of directories and files according to their placement. Main interface which handles the common method find() is ’ DataEntry class’ which is implemented by FileEntry and Direactory classes. Directory class handles a core component which recurse through directories and build the hierarchy to pass the function to ‘FileEntry’ Object which handles the search function whenever a file is found.