



Assessment Item 3:

Application of Programming in Java

Bryce Sandilands: 11649401
ITC206-201860





Task One: Logic Explanation

The program consists of two files, `Subject.java` & `TestSubject.java`. The `Subject` class contains all the properties and methods to create, validate, sort subjects. The `TestSubject` class holds the main logic of the program, where the user can create a list of subjects with all of the validation and error prevention logic. Due to the word count and amount of methods this is an overview of the structure and flow of the code. Each method is commented with its function and state.

- For the `Subject` class, I chose to replace all `array's` with `ArrayLists` (since some of the logic has an unknown amount of items in list before processing) to keep the class as a whole consistent.
- The `Subject` class constructor assigns the encapsulated properties and calls the `isValidCode` method, this method validates every possible error for the subject code and leaves the constructor neat and readable. The object will not be created if an error occurs.
- The rest of the `static` methods provide what is asked in the subject outline by taking arguments of `ArrayList<Subject>` and `String` and processing them as explained in commented code.
- The `TestSubject` class is broken up into 3 main parts. The `main` method, the processing methods & the `run` method. The `main` method instantiates a `File`, `FileWriter` and two `Scanners` for creating, reading & writing to a text file, and getting input from user.
- An `ArrayList` is created and `readFromFileToList` is called. This uses the `Scanner` to get the two `substring` from each line in the text file, use them to create a new `Subject` and add it to the `ArrayList`.
- After getting the user input, a `while` loop keeps iterating until the user chooses "n" to adding another subject. Every time this iteration happens, if the user chooses "y" the `run` method is called which calls all processing methods in order to `displayDisciplines` get input for `discipline`, `disciplineCode`, `subjectName`, from user. The method then collects this information, creates a new `Subject` and adds it to the list by calling `addSubject`. Following this `writeToFile` is called to write all of the subjects from the list to the file.
- If the user inputs anything other than "y" or n the loop iterates again. This continues until the user inputs "n". This is when the program exists the loop, and `endProgramAndDisplayList` is called.
- `endProgramAndDisplayList` lists contents of list while.
- `fileIn` and `userInput` are then closed.





Task One: Sample Output

```
Would you like to add subjects to your list? (y/n): y

There are no subjects in your list
Please enter the three alphabetical characters for the discipline you wish to add: ITCC
Please re-enter discipline as three alphabetical characters: ITC

There are no existing subject codes in your list for that discipline:
Please enter the three numerical characters for the subject code: 2066
Please re-enter a valid three numerical subject code: 206

Please enter the the name of the subject: Programming in Java

Would you like to add another subject? (y/n): n

Thank you!, your list is as follows:
ITC206 Programming in Java
```

✓ Run Succeeded Time 592 ms

Sample Output One Showing:
Validation for each of the user inputs

Sample Output Two Showing:
Validation for existing subject code.
Listing of existing disciplines.
Adding multiple subjects testing while loop
Exiting while loop and displaying newly added and already saved items from the file.

```
Would you like to add subjects to your list? (y/n): y

Your current disciplines are as follows:
ITC
Please enter the three alphabetical characters for the discipline you wish to add: ITC

The existing subject codes for your chosen discipline are as follows:
206
Please enter the three numerical characters for the subject code: 206
Code already exists, please re-enter the subject code: 204

Please enter the the name of the subject: Human Computer Interaction

Would you like to add another subject? (y/n): y

Your current disciplines are as follows:
ITC
Please enter the three alphabetical characters for the discipline you wish to add: MTH

There are no existing subject codes in your list for that discipline:
Please enter the three numerical characters for the subject code: 105

Please enter the the name of the subject: Intro to Math

Would you like to add another subject? (y/n): n

Thank you!, your list is as follows:
ITC206 Programming in Java
ITC204 Human Computer Interaction
MTH105 Intro to Math
```

✓ Run Succeeded Time 612 ms





Task Two: Logic Explanation

The `Triangle` class inherits from `GeometricObject`. It contains 3 `private` side variables to create a `Triangle`, get methods for each side and two overridden `abstract` methods of `getArea` and `getPerimeter` which both return their respective values

- The `Triangle` class is very self explanatory, comments have been kept simple for this reason. The `class` enables you to create a default `Triangle` with sides of 1.0, and also supplies a `constructor` allowing user to select the length of an individual `side`.
- The accessor methods return their respective values.
- The `toString` method makes use of the abstract methods from the `super class` by calling them to display the values from `getArea` and `getPerimeter`. This method also uses the accessors inherited from the super class to return the color & filled status of the current triangle.





Task Two: Sample Output & UML Diagram

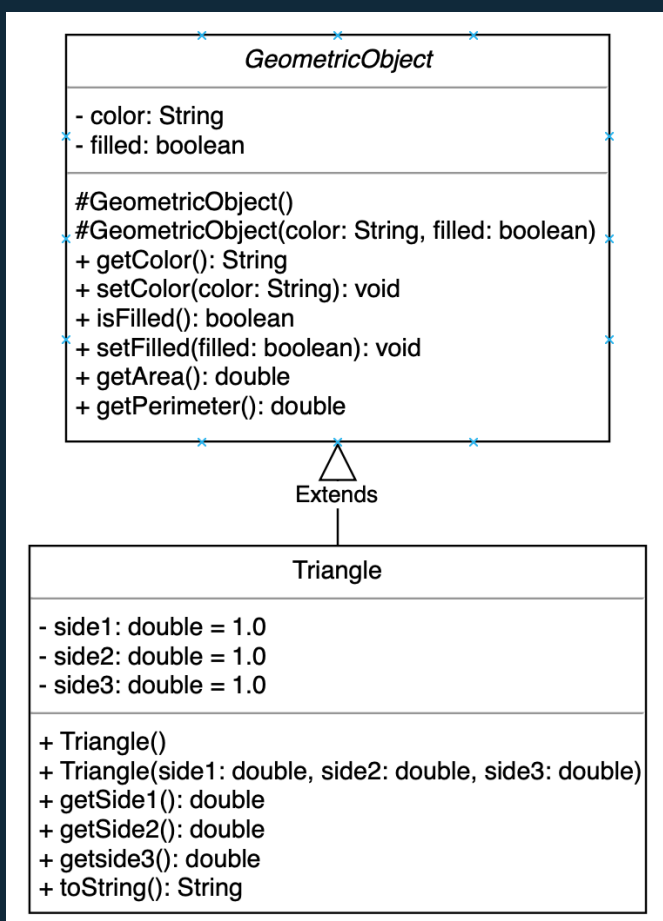
Sample Output One

```
The area is: 36.0  
The perimeter is: 12.0  
The color is: yellow  
The triangle is filled: true
```



Run Succeeded

Time 379 ms



UML Diagram

