**Comprehensibility Score (CS) Measurement Process:**

**A. For Class Level:**

In this study, we introduce the **comprehensibility** metric and show a way to measure it. ***A class is regarded as comprehensible, if the entities used in this program is readable (or understandable).*** We refer these entities as a **bag of entities** of the class. The bag of entities consists of *package name*, *class name, method name*, and *variable name* of the class. **The keywords used in Java programming language (or keyword list for other languages) are excluded in this bag of entities.** The general loop variables such as *i, j* are also excluded here. The entities in the bag of a class are then categorized into following three levels:

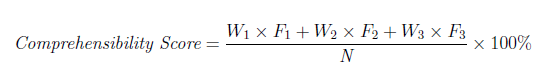
1. **Well Readable:** If an entity is completely meaningful and helps to understand the program. For example, *calculateSalary()* is a method name which is completely understandable and well readable, as after splitting the entity (*calculate, salary*), each of these are also completely meaningful. The entities of this category carry 100% weight.
2. **Moderate Readable:** If an entity is not fully meaningful but helps to understand the program, that is, the entity is partially understandable. For example, *calSalary()* is partially understandable and moderate readable, as after splitting the entity (*cal, salary*), each of these are moderately meaningful. In general, the entities of this category carry 50% weight. However, its weight can be dynamic: after splitting, each meaningful word carries 1, moderate meaningful carries 0.5 and non-readable carries 0.0 weight. In this way, the weight of ***calSalary*** is: (0.5×1+1.0×1)/2 = 0.75 = 75% (Similar to the following equation).

Another example, ***calSalaryX*** is: (0.5×1+1.0×1+0.0×1)/3 = 0.5 = 50%.

1. **Non Readable:** If an entity is neither meaningful nor understandable. For example, *cs()* is not understandable and non-readable. The entities of this category carry no (0%) weight.

[*Use a dictionary for matching the words and labeling these as well, moderate and non-readable ones.]*

After the categorization, we quantify the comprehensibility metric for a class using the following equation.



Here, **W1, W2** and **W3** represent the weights for the entities of **Well Readable, Moderate Readable** and **Non Readable** level respectively. We weights these **1, any between 0 and 1,** and **0** respectively based on the definition as discussed above. **F1, F2** and **F3** represent the frequency of the entities of the corresponding level of the class respectively. **N = F1 + F2 + F3** represents the total number of entities of the class.

For example, if a class contains 10 entities: 5 of these are well readable, 3 moderate readable, and 2 non readable. So, **Comprehensibility Score** = (5 × 1 + 3 × 0.5 + 2 × 0) / 10 = 0.65 = 65%, which means that the class is 65\% comprehensible.

**B. For Project Level:**

1. The program will show a list of programming languages for which it will calculate the **Comprehensibility Score** for the project. Like – Please select a programming language (1, 2, 3 or 4) for your project: 1. Java, 2. C#, 3. Python, 4. JavaScript
2. Then the user will provide input for the language of the project, let’s say Java, so input will be 1.
3. The program will read and calculate the **Comprehensibility Score** for each class file (e.g., .java file) of the project. [Follow the process: A. For Class Level].
4. Finally, the program will calculate the **Comprehensibility Score** for the whole project by averaging the **Comprehensibility Scores** of the classes.

Program Output: ProjectName\_ComprehensibilityScore.csv file consists of two sheets:

1. Sheet-1 Name: ClassWeiseCS: Show Class wise score and project comprehensibility score.
2. Sheet-2 Name: ClassWiseDetailsCS : Show each entity of each class in details with scores.