WCC Measure – Computation Guidelines

1. Identification of the tokens begins **after** the **class declaration**.
2. In general, all the operators, keywords (except access flags such as public, static, etc.), strings, identifiers, and numerical values (including zero) are identified as separate tokens. However, they are exceptional cases.
3. All the characters inside a pair of double quotes are identified as a single token.
4. Along with the array name, the square brackets of an array are considered as one token.
5. Each comma operator that separates two program components from one another, is identified as a separate token.
6. Brackets are not identified as separate tokens.
7. In a program statement that contains a **variable declaration**, the variable name is not identified as a token. However, if a program statement contains a **variable definition**, then the variable name is identified as a token.
8. In a method declaration or invocation, the round brackets and the method name are identified as one token. However, the components inside the round brackets of **user-defined methods** are not identified as tokens. Similarly, the components inside the round brackets of the **constructors of user-defined classes** are also not identified as tokens.
9. In a decisional statement, along with the keyword that defines the decisional type, the round brackets or the colon are identified as one token. Thus, **if( )** , **if-else( )**, **else-if( )**, **for( )**, **while( )**, **do-while( )**, **switch( )** and **case:** are identified as one operator. The words **'else'** and **'do'** are not considered separately for the complexity calculation.
10. The keyword **'catch'** and the round brackets are identified as one operator. However, the word **'try'** is not considered for the complexity calculation.
11. The **'.' operator** that is used to connect classes, fields and/or methods is identified as a separate token. The **class**, **method**, or **field names** which are connected by '.' operators are also identified as separate tokens.
12. The statement terminator (;) is not identified as a token.
13. Manipulators such as **endl**, **"\n"** are considered as tokens.
14. The "∗" sign used in the declaration of pointer is not a dereference operator. It is just a notation that creates a pointer. Thus, it is not considered as a token.
15. The keyword **'return'** is not considered as a token.
16. For a program which does not have a **built-in root class**, the weight allocation of the Wi attribute begins at **1**.