ASSIGNMENT WEEK 2

COM/0037/21

OLIVER ANDAYI

1.) Discuss lexical analysis.

Lexical analysis, also known as lexing, tokenization, or scanning, is the first phase of a compiler. It is the process of breaking down a stream of characters into smaller parts, called tokens, which are the basic units of meaning in a programming language. Lexical analysis is performed by a lexical analyzer which reads the source code one character at a time, and identifies tokens based on the rules of the programming language.

Once the lexical analyzer has identified a token, it returns it to the compiler. The compiler then uses the tokens to parse the source code and generate machine code.

Lexical analysis is an important part of compilation because it provides the compiler with a well-defined and manageable set of tokens to work with. This makes the parser's job much easier, and allows the compiler to generate more efficient machine code.

Applications where lexical analysis is used include the natural language processing and the text editor. In the natural language processing its used to identify words and phrases in text. In text editor its used to identify keywords and other special tone in a source file.

2.) Discuss why lexical analysis is important in lexical analysis

Lexical analysis is important in lexical analysis since it is the first step in processing text. It breaks down text into tokens which makes it easier for subsequent steps.

a.) It makes parsing easier. The parser is responsible for analyzing the structure of the text and determining its meaning. However, the parser cannot do this effectively if the text is not first broken down into tokens. Lexical analysis provides the parser with a well-defined set of tokens to work with, which makes its job much easier.

b.) It improves code efficiency. Lexical analysis can be used to identify common patterns in the text and replace them with more efficient code. This can lead to significant improvements in the performance of the program.

c.) It helps detect errors. Lexical analysis can be used to detect errors in the text, such as misspelled words, missing semicolons, and undefined variables. This can save a lot of time in the debugging process.

d.) It is used in other applications such as natural language processing and text editors.