TASK: Look at each of the phrases below and ensure you understand what each of these means. For any that you do not understand, do a little research to find a definition of each term. This research may involve looking back over these notes, or the associated lecture notes. It may also involve searching for these terms on the Internet.

● Source code: It’s written by programmers and it’s a list of commands ready to be compiled or assembled.

● Machine code: In a simple word, it can be only understood by computer and it’s a set of binary or hexadecimal instructions.

● Interpreter: An interpreter produces results from the program and translates high level programming language into machine level code line by line as the code runs.

● Compiler: A compiler converts high level programming language into machine code before the program runs and it takes a lot of time to analyze the source code.

● 2GL, 3GL, 4GL: 2GL are low level languages and they were developed in 1950s. 3GL are known as high level languages and they are much like normal text and mathematical formulas. For example, C++, Java, C, JavaScript, and Visual Basic. 4GL are known as very high-level languages and they are very easy to write and read by programmers. For example, Python, Ruby, SQL, and Perl.

● Executable: An executable refers to a file or program that can be run or executed by a computer’s operating system. Executables contain instructions and code that the computer’s processor can understand and execute, which allows them to perform specific tasks or functions.

● Expressions: In the languages of mathematics and computer programming, an expression is a set of values, variables, operators, and functions that, when evaluated, yields one value. Expressions can be as basic as a single constant value or as sophisticated as a mix of operations and variables. They are an essential idea in computer programming as well as mathematics.

Here are a few phrases with examples:

1. Expressions in Mathematics:

An expression that evaluates to the value {5} is {2 + 3}.

- The form {x \* 4} denotes a variable {x}, the value of which determines the outcome.

2. Syntactic Formulas:

- The logical statement "a > b" evaluates to either "true" or "false" depending on how "a" and "b" are compared.

3. String Concatenation: - The expression {"Hello, " + "world"} joins two strings to form the string "Hello, world."

4. Function Calls: - The expression {sin(30)} calls the sine function with a 30-degree parameter and returns a numerical value.

5. Compound Expressions: A compound expression that combines several sub-expressions with operators is - {(x + y) \* (a - b)}. What happens depends on the values of {x}, {y}, {a}, and {b}.

Expressions are widely used in computer programming to carry out calculations, reach judgements, and work with data. They are an essential component of algorithms and are vital to the operation of computer programs. An essential component of how computers interpret and manipulate data to carry out different tasks is the evaluation of expressions.

● Operators and Operands: An operator is the function that performs the operation, whereas the operand is the input to that functions. In “a+b” a and b are the operands while the ‘+’ is the operator.

● Syntax Errors: It’s an error which usually occurs if the programmer wrote the sequence of characters or tokens that is intended to be written in a particular programming language in a wrong way.

● Logical Errors: An error when the instructions given in the program do not accomplish the intended goal.