Assignment #4

Check Your Understanding of CS Concepts!!!

Part 1:

Computation: Informational processing when it comes to computers.

Central Processing Unit (CPU): The brain of the computer where all the operations of the computer are ran and executed.

Arithmetic Logic Unit (ALU): A piece of hardware of the motherboard that computes the arithmetic and logical operations of a program of a computer. Such as an instruction containing operand and operations.

Registers: They are a part of a computer processor that stores certain amounts of data. Registers must be at least large enough to store an instruction.

Memory: Are instructions and data are stored electronically. It is ran through the computer's microprocessor, and there are different types, such as random access memory or read-only memory.

Cache: A temporary storing place for data. Some uses that are popular for cache are cache servers, disk caches, cache memory, and flash cache.

Disk Drive: Stores large amounts of data for the computer by reading/writing memory, data, information, etc. onto a disk.

Input: Information that goes into a system, program, etc.

Output: Information that get exported out of a system, program, etc.

Compiler: A program that takes statements and converts it in order for the computer to understand such of information.

Interpreter: A program that has logically sequenced series of operating commands that converts it into something that the computer processor can work on.

High-level Language: Language of a program that can be interpreted or compiled.

Intermediate Language: A form of complied high-level language, which is faster and is translated prior/ahead of time to runtime.

Machine Language: A form of complied high-level language, which is faster and is translated prior/ahead of time to runtime.

Integer: A whole number.

Float: A real number.

Double: A larger value of numbers of a comparison of multiple values.

String: A method in inputting numbers/equations into a printed sentence of a program.

Variable: A place to assign information to.

Constant: A set value or phrase of a variable.

Data Type: A programming language that has a valued data with certain characteristics.

Expression: A certain phrase or equation that contributes a meaning.

Assignment Statement: An action of setting or resetting a value or phrase to a variable.

Mathematical Operator: Symbols that calculate operations that output a given value.

Logical Operator: Phrases that compare the statements of True and False.

Relational Operator: Symbols that compare values of certain assigned variables.

Precedence: Regards to the order of statements written in a program.

Function: A preset tool that can compute a given task to a value/phrase.

Part 2:

Step 1: Problem Analysis.

a. The program needs to be able to take an input of a positive whole number and out the square root of that number, but it has to by using the Babylonian algorithm. Then at the end, the resulting output is compared the actual calculated square root of that inputted number by using the square root function by Python.
b. One should know how to solve the square root of a positive whole number by using the Babylonian algorithm.

Step 2: Program Design.

1. Prompt the user for a positive whole number.
2. Read the inputted number.
3. Prompt the user for a desired number of loops wanted to compute their number by using the Babylonian algorithm.
4. Read the desired amount of loops.
5. Calculate a guess of the answer of the inputted number’s square root, by calculating half of the number.
6. Calculate the inputted number divided by the guess.
7. Set the guess to the guess plus the previous calculation all divided by two.
8. Loop steps 5 and 6 depending on the number of desired amount of loops determined by the user.
9. Print the result of the guess.
10. Compare the resulting guess to the actual square root of the initial inputted number.
11. Print the calculated comparison.

Step 4: Program Testing. Create a Test Plan with several test cases including the average and extreme cases.

Average Case: When the user inputs positive whole numbers, the program will run as it was written.

Extreme Case: The larger the inputted number becomes, the more required loops increases also. So the ending comparison of the two numbers may not match identically.