RickiConnor.com

HOW TO UNDERSTAND COMPUTER LOGIC

...LIKE AN INFORMATION TECHNOLOGY CONSULTANT

Ricki Connor, author

I thank God for family.

Thanks to my family for their support.

#

Chapter One - Qualifications

I worked in the Information Technology field for over twenty-five years. I began as a Computer Operator and recently accepted a five year, remote assignment as an Information Technology Project Manager consultant. As you read this eBook, my hope is you will realize a clear path for you to begin a career in Information Technology.

My educational degrees include: Associate of Science in Data Processing, Bachelor of Science in Computer Information Systems, Master of Arts in Teaching Educational Technology, and Master of Business Administration. I am also Scrum Master certified. Certifications are a great way to learn many different skills; and will prepare you for more opportunities. I will share more about this.

The college courses that seemed to help me most, were: Logic and Philosophy, Algebra, Flowcharting, and the Computer Information System curriculum.

For more information, other topics can be researched by using the internet.

Information Technology positions I have held, include COBOL Programmer (Developer/Coder), Documentation Author, Metrics Owner, Requirements Lead, Project Manager, Senior Business Analyst, and any combination of these positions.

As you increase your knowledge of Information Technology, more responsibilities (positions and roles) may be assigned to you. Logic is the science of reasoning and careful thought; the one skill I always use to resolve problems. Once you understand logic, all else becomes easier. Your Information Technology reputation relies on successful delivery of solutions. The phases to provide correct solutions will be shared soon.

This eBook attempts to share my experience with working in the Information Technology field, and more specifically, logic. Also shared are tips and information to help you understand logic and how it relates to Information Technology and computer processing.

Ideas for study, improvement, internet research, and other motivational tips are also mentioned.

#

Chapter Two - Circuits

Circuits are what will be discuss in this chapter. Computers are machines, composed of electric circuits that turn on and off at very high speeds. Either a circuit's electric power is on, which is represented as a one (1), or the circuit's electric power is off, represented as a zero (0). These circuit patterns (either electric on or electric off) combine to create variables, constants, counters, rules, instructions, words, sentences, complete programs, and most importantly, logic. As an example, the On and Off states can be used to assign 'True' and 'False' values. These values can be compared in an 'IF' statement comparison. More information on these statements in Chapter Four.

Computers also have storage capabilities, a central processing unit that executes the processing, input devices (mouse/keyboard/etc.); and a monitor (or other peripherals). Output devices (printers, monitors, speaker, etc.) help users realize computer actions (from coded instructions).

Computers can do nothing without human instructions. Users are people that use computers. Computers follow instructions written for them, called code/programs/scripts. Programmers (sometimes called coders or developers) write the instructions computers follow. Computers process instructions at an extremely high speed.

Programs are what computers interpret, to perform instructions. When learning a computer language, most courses begin with teaching how to send a message to the monitor similar to: “HELLO WORLD”.

Computer languages are like foreign languages, having verbs and nouns; and are easy to learn if you avail yourself. If you enter the correct language syntax, you will have a successful program run. Languages have special programs (compilers) that stop computer processing and identifies syntax mistakes. Syntax and logic mistakes (bugs) must be eliminated so a program can run successfully. A program will be successful if it completes with no error messages; and produces the desired output.

Many times no error message is received, and the results are not what was expected. These are logic errors that compilers cannot detect. The one thing a program needs to produce the expected outcome is logic.

After studying this eBook, you will be able to understand basic computer logic, how to use logic in programming and design; and begin an entry-level career in Information Technology. Study and review all the Information Technology terms in this eBook to improve your communication and understanding.

#

Chapter Three - Initiating actions

Computers need instructions to process or run a program. These instructions (computer programs) direct the computer to a desired outcome. The goal is to logically arrive at a desired outcome, which was decided by a manager or customer. The following tip will help.

Starting at the beginning of the logical process is the INPUT of a desired outcome. Input can be a data file or it can be an initiating action. If given a data file to rearrange, do not write to the original data file, as it will erase the original data. Make a copy of the original data file and use it, so the original data is not erased. Another alternative is to write to a new file, which keeps the original file untouched.

NOTE: 'Write to a file' means to store information in a file. A file is simply a storage area used by the computer. When you want to save information, write it to a file and name it. This is referred to as digitizing your information.

The information can be text, images, sound, or video content.

Think of an initiating action as the start of the process. Clicking a SUBMIT button is an example of an initiating action. Instructions are captured in the button, and are executed after the button is pressed. The following examples initiate computer instructions:

* The cursor contacts a specified position on the computer screen, and more options appear

* A counter (accumulator) reaches a desired amount, and a blinking indicator is activated

* When receiving an email or other notification, a ding is heard

* The cursor entering into an address field, and the previous entries are shown

* The selection of a state automatically appears in the correct place

* Creation of a customer information file, and the program completes successfully

The input starts everything. Otherwise, computer programs just sit in memory. Think about how you search for a word at Dictionary.com.

INPUT a word, Dictionary.com finds its matching definition (PROCESS), and the OUTPUT is presented for you to understand. This is the same logical pattern (phases), throughout these writings: INPUT, PROCESS, OUTPUT.

Input is important to understand and identify, in order to logically transform information into the output desired. Again, input may be a physical file or request, but many times it is a click of the mouse or arrival of an email, or some other initiating action. You can begin the logical process with understanding the input. Do you have the correct input to produce the desired output? Before diving into computer instructions, input needs to be understood and identified.

NOTE: The old saying: “Garbage in, garbage out” also refers to incorrect input files to solve a problem or request.

#

Chapter Four - Process

The next phase is the Process phase. This is where logic examines the input and produces the output. You can compare anything using computers and variables. Variables hold contents for the computer to use in comparisons.

Example 1: A file of names, addresses, city, state, and zip codes, have been identified as the input variables. The output desired is in the format: Name and zip code. The instruction would read similar to: Read a record (files are made up of records that contain fields). If the field is a name or zip code, provide it for output. If the field is not a name or zip code, the value is discarded; and nothing is written to the output. We will talk about outputs in the next chapter.

Example 2: Input as an initiating action (mouse click) is tested against pre-defined rules. When the mouse button is selected (pressed), the process begins to read and execute the instructions stored within that button object.

This is evident when entering your name and other information, then selecting the submit button. The name and other information on the form/screen are the input. When the mouse is clicked (selected), the program instructions are initiated. These instructions cause the computer to enter the incoming information and store it until needed.

Someone made a decision to manipulate the incoming data in a certain way to produce the desired output. The user will not see or know this, but the data is manipulated and transformed into output (hopefully the desired output).

Example 3: The mouse button is selected, accepts the data, stores it, and provides an output message. The computer user enters the information; and the screen refreshes with the message that was created and stored in the instructions. Careful thought has to go into programming for computer s to understand next steps.

The IF/THEN/ELSE statements are very powerful, and understanding them is understanding logic. Remember, computers read instructions as zeros and ones. Zero is off/false/no, and One is on/true/yes.

A variable is an area within computer memory that holds a value. If a variable is defined with a value of zero, a test can decide the actions to take.

Example 4: IF Variable = 0, THEN do this, ELSE do that. There are several combinations involving these type of logic tests. This is a tip to coding logical instructions.

Go to YouTube.com, search for IF statements and Programming Loops. These are similar in computer languages.

#

Chapter Five - Logical Thinking

The Output phase is decided before computer instructions are requested and completed. Ending with what is asked for is very important.

Forethought has to be used to create alerts and messages for users to understand what is happening with the computer. These alerts and messages include pop-up messages and wave (sound) files. These are two methods the computer can easily send. When programmers code, user communications are also included in instructions in the form of relevant messages. Computer users must be made aware of what they can or must do next.

Thinking logically like a Computer Programmer also means understanding how to communicate with computer users. No user should be confused when they have responded to a computer’s instruction. Thinking logically will help programmers communicate with users through their instructions. Besides creating instructions to fulfill a goal, programmers must think ahead and allow for user communication.

What is the first step needed to solve this problem, and what scenarios exist? What does the user see on the screen? What should the user do? What do you want the user to do? After planning for different scenarios, identify the next logical step. The cycle repeats; with the IF statements determining the logic from comparing the input fields.

Begin with the end, in mind. The output is the first thing to design and solve. Is it a file or is it an action? If it is a file, what format or order is desired? If it is an action, how is the action initiated?

During programming, and once the output method has been decided, a placeholder (or stub) can be created to test the output results. First, send sample output to test files to ensure the data format and your logic are correct.

Expected actions can be reviewed, changed and tested multiple times in a testing environment. After all testing is done, and the desired output is produced, programmers run the code in a production environment.

Computer users access the production environment to run programs. Remember: Begin with the end in mind. What is the desired output?

It is best to know where you are going, so you will know when you arrive. Make sure you ask questions with whomever gives you information or describes the desired output. It is very important to create exactly what is described. Document verbal instruction to ensure you followed the requirements.

#

Chapter Six - Additional help

To enhance your Information Technology studies, use Google.com and YouTube.com to research any words and terms you are unfamiliar with. A good practice to continue with is to use a dictionary or Google words you are not familiar with. Do not try to guess what a word means by the content of the sentence.

Spend time researching, reading, and understanding Information Technology. Allocate a little time to study each day; and you will realize, minutes a day can add up to mastering a new skill, if the study is consistent.

Certifications prove that a focused learning process has taken place. The one thing that certifications and online courses do not provide is experience. Certifications and classes offer focused, content learning, but are not a substitute for actual work experience.

Online classes and certifications are great ways to learn new skills.

Coursera.com, Udemy.com, and just a search on online certifications or courses will provide more information on courses anyone can purchase and attend.

#

Summary

To understand computer logic, like an Information Technology Consultant, begin with the end in mind. Step one is to completely understand what is required for the output of the assignment, program or project.

The next step is to identify the input that will correctly produce the desired output. Input can be an action (entering a value, clicking the mouse, etc.), as well as creation of a new file, that initiates another process.

Lastly, the process will convert the input into the output, by using logical comparisons of variables and pre-entered instructions (code or programs).

Logic is the study of reasoning and careful thought. You can now use reasoning to “think” logically; and provide instructions to computers. These instructions can produce a desired outcome using logic.

This is a very basic eBook on Understanding Computer Logic.

Research and review the unfamiliar words in the glossary to improve your knowledge. The next time you use your computer, think about: Input, Process, and Output. Try to identify each of these phases as you use your computer applications. Once you understand this simple lesson, all computer processing work will become easier for you to understand.

You have basic knowledge to begin learning a computer language. Search for the following on YouTube: ‘programming languages,’ ‘solidity programming,’ ‘programming tutorials,’ and ‘SWIFT.’ If you want to learn, go get it. Good luck!

#

Glossary

Alerts: Make users aware of potential threats, next instructions, or emergency situations. Alerts provide direction to users on how to respond.

Bug: A syntax or logic error in a programming language.

Code: Instructions written for computers to follow and process. Also see program.

Constants: A value defined in a computer program that does not change its value.

Consultant: A temporary knowledgeable worker assigned to a project.

Counters: Spaces defined in computer storage that count and accumulate number values.

Cursor: The marker on the computer screen that moves in relation to the mouse movements.

Digitize: Saving your information on electronic medium (computer, disk, jump drive, etc.)

Information Technology: The use of computers, storage, networking and other physical devices, with processes to create, process, store, secure and exchange all forms of electronic (digitized) data

Initiating: To begin an action.

Input: Data received and processed by a computer.

Object: any item, either physical or digital.

Output: Data, sound, or result of processing the input.

Process: an action or series of actions that a computer performs when it receives information.

Program (see code also): Instructions written for computers to follow and process.

Programmers (sometimes called coders or developers): People who write the instructions (programs) computers follow.

Programming Loops: Instructions continually repeated until a certain condition is achieved.

Project Manager: The resource responsible for day-to-day management of a project.

Read a record: Accept a record into the computer's memory for processing. See Input.

Rules: Statements informing people what they must do in order to achieve success.

Stub: Code used as a place holder for programming functionality.

Users: People that use computers.

Variables: Defined areas in computer storage that hold values. Variables are changed as new values and operations are applied.

Write to a file: Storing digitized information in a file.

#

THE END