**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **30-05-2020** | **Name:** | **Karthik J** |
| **Course:** | **Logic Design** | **USN:** | **4AL16EC030** |
| **Topic:** | Application of PLC’s | **Semester & Section:** | **8TH A** |
| **GitHub Repository:** | Karthik-J |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image Section**         **What is a PLC?** PLC stands for “Programmable Logic Controller”. A PLC is a computer specially designed to operate reliably under harsh industrial environments – such as extreme temperatures, wet, dry, and/or dusty conditions. It is used to automate industrial processes such as a manufacturing plant’s assembly line, an ore processing plant, or a wastewater treatment plant.  PLCs share many features of the personal computer you have at home. They both have a power supply, a CPU (Central Processing Unit), inputs and outputs (I/O), memory, and operating software (although it’s a different operating software). The biggest differences are that a PLC can perform discrete and continuous functions that a PC cannot do, and a PLC is much better suited to rough industrial environments. A PLC can be thought of as a ‘ruggedized’ digital computer that manages the electromechanical processes of an industrial environment.  PLCs play a crucial role in the field of automation, using forming part of a larger SCADA system. A PLC can be programmed according to the operational requirement of the process. In the manufacturing industry, there will be a need for reprogramming due to the change in the nature of production. To overcome this difficulty, PLC-based [control systems](https://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/) were introduced. We’ll first discuss PLC basics before looking at various applications of PLCs. **PLC Basics** PLCs were invented by Dick Morley in 1964. Since then PLC has revolutionized the industrial and manufacturing sectors. There is a wide range of PLC functions like timing, counting, calculating, comparing, and processing various analog signals.  The main advantage of PLC over a “hard-wired” control system is that you can go back and change a PLC after you’ve programmed it, at little cost (just the cost of the programmer’s time). In a hard-wired control system, you’re essentially having to rip out wires and start from scratch (which is more expensive and takes longer). Let’s look at an example to better understand this advantage.  Imagine you have a light connected to a switch. In general, the light operates under two conditions – ON and OFF. Now you are given a task that when you turn ON the switch, the light should glow only after 30 seconds. With this hard-wired setup – we’re stuck. The only way to achieve this is to completely rewire our circuit to add a timing relay. That’s a lot of hassle for a minor change. **How Does a PLC work?**  The working of a PLC can be easily understood as a cyclic scanning method known as the scan cycle.  A PLC Scan Process includes the following steps   * The operating system starts cycling and monitoring of time. * The CPU starts reading the data from the input module and checks the status of all the inputs. * The CPU starts executing the user or application program written in relay-ladder logic or any other PLC-programming language. * Next, the CPU performs all the internal diagnosis and communication tasks. * According to the program results, it writes the data into the output module so that all outputs are updated. * This process continues as long as the PLC is in run mode. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | | **30-05-2020** | **Name:** | **Karthik J** |  |
| **Course:** | | [Programming with Python: Hands-on Introduction for Beginners](https://www.udemy.com/course/python-programming-beginners/) | **USN:** | **4AL16EC030** |  |
| **Topic:** | |  | **Semester & Section:** | **8th A** |  |
|  | **AFTERNOON SESSION DETAILS** | | | | |
|  | **Image of session** | | | | |
|  | Dictionary A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values. Example Create and print a dictionary:  thisdict = {   "brand": "Ford",   "model": "Mustang",   "year": 1964 } print(thisdict) Python - Tuples A tuple is an immutable sequence of Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.  Creating a tuple is as simple as putting different comma-separated values. Optionally you can put these comma-separated values between parentheses also.  For example −  tup1 = ('physics', 'chemistry', 1997, 2000);  tup2 = (1, 2, 3, 4, 5 );  tup3 = "a", "b", "c", "d"; Accessing Values in Tuples To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index.  For example −  tup1 = ('physics', 'chemistry', 1997, 2000);  tup2 = (1, 2, 3, 4, 5, 6, 7 );  print "tup1[0]: ", tup1[0];  print "tup2[1:5]: ", tup2[1:5]; Updating Tuples Tuples are immutable which means you cannot update or change the values of tuple elements. You are able to take portions of existing tuples to create new tuples  Example  tup1 = (12, 34.56);  tup2 = ('abc', 'xyz');  # Following action is not valid for tuples  # tup1[0] = 100;  # So, let's create a new tuple as follows  tup3 = tup1 + tup2;  print tup3; Delete Tuple Elements Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.  To explicitly remove an entire tuple, just use the **del** statement.  example −  tup = ('physics', 'chemistry', 1997, 2000);  print tup;  del tup;  print "After deleting tup : ";  print tup;  This produces the following result. Note an exception raised, this is because after **del tup** tuple does not exist any-more − Basic Tuples Operations Tuples respond to the + and \* operators much like strings; they mean concatenation and repetition here too, except that the result is a new tuple, not a string. Indexing, Slicing, and Matrixes Because tuples are sequences, indexing and slicing work the same way for tuples as they do for strings. Assuming following input −  L = ('spam', 'Spam', 'SPAM!') | | | | |