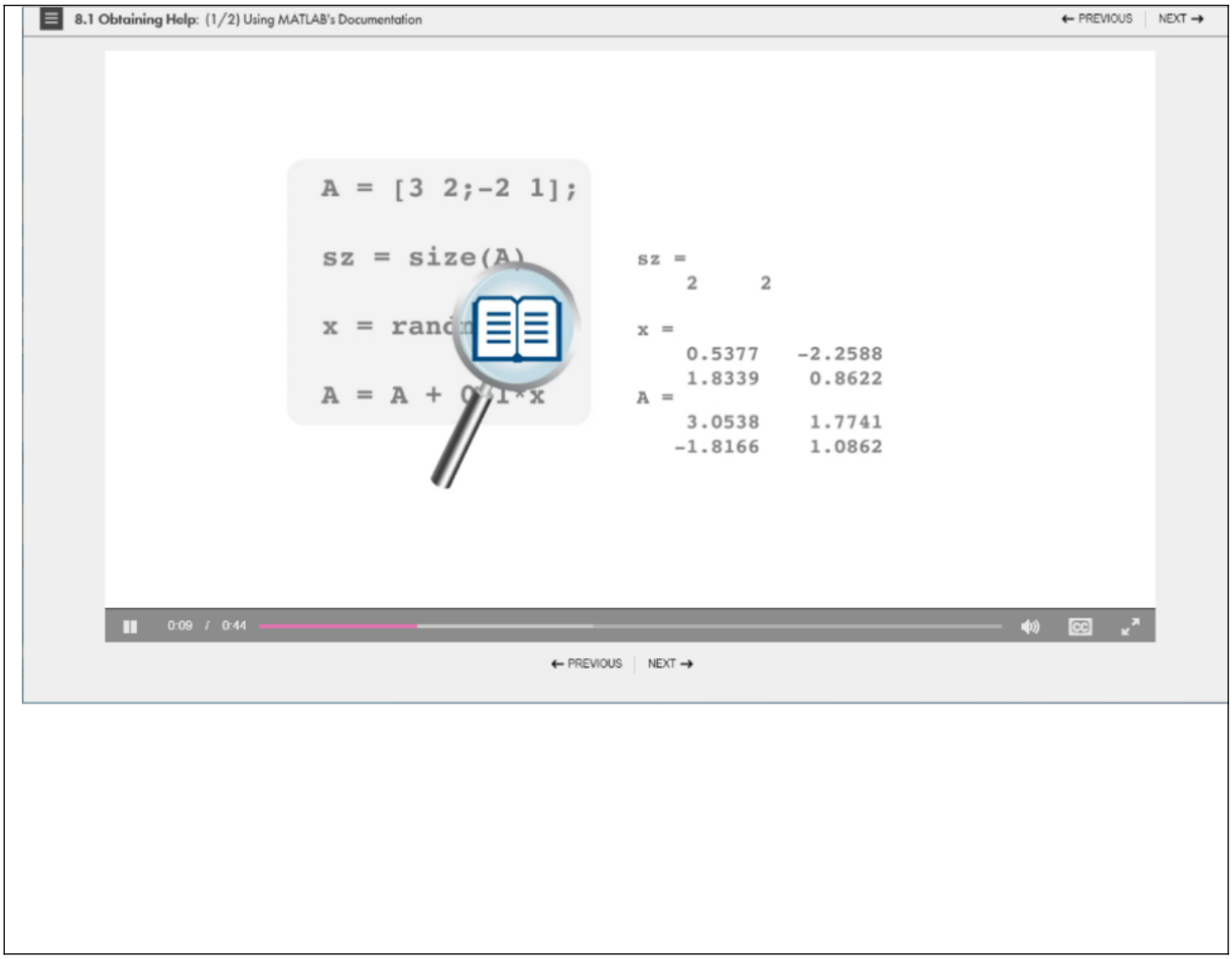
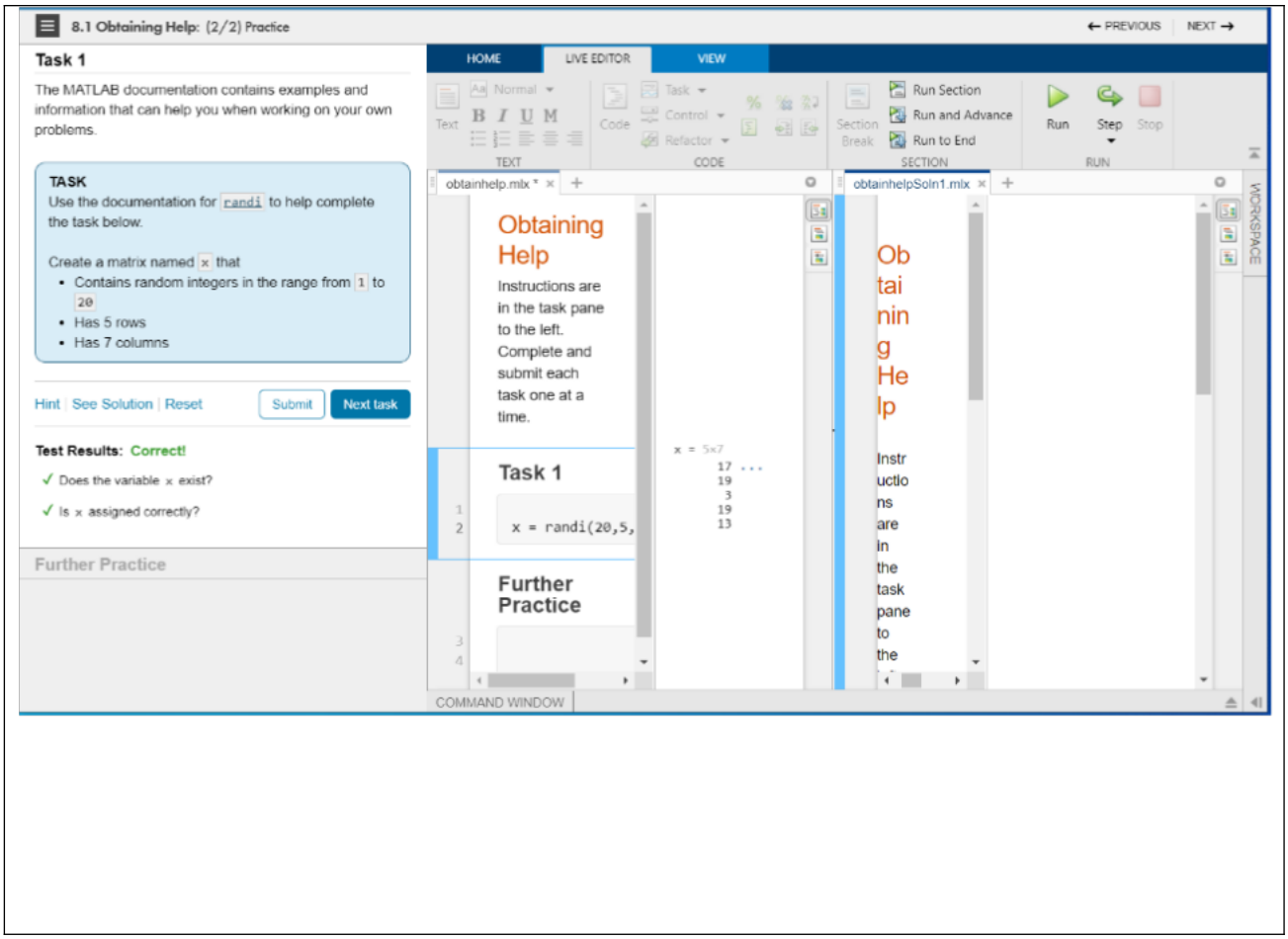
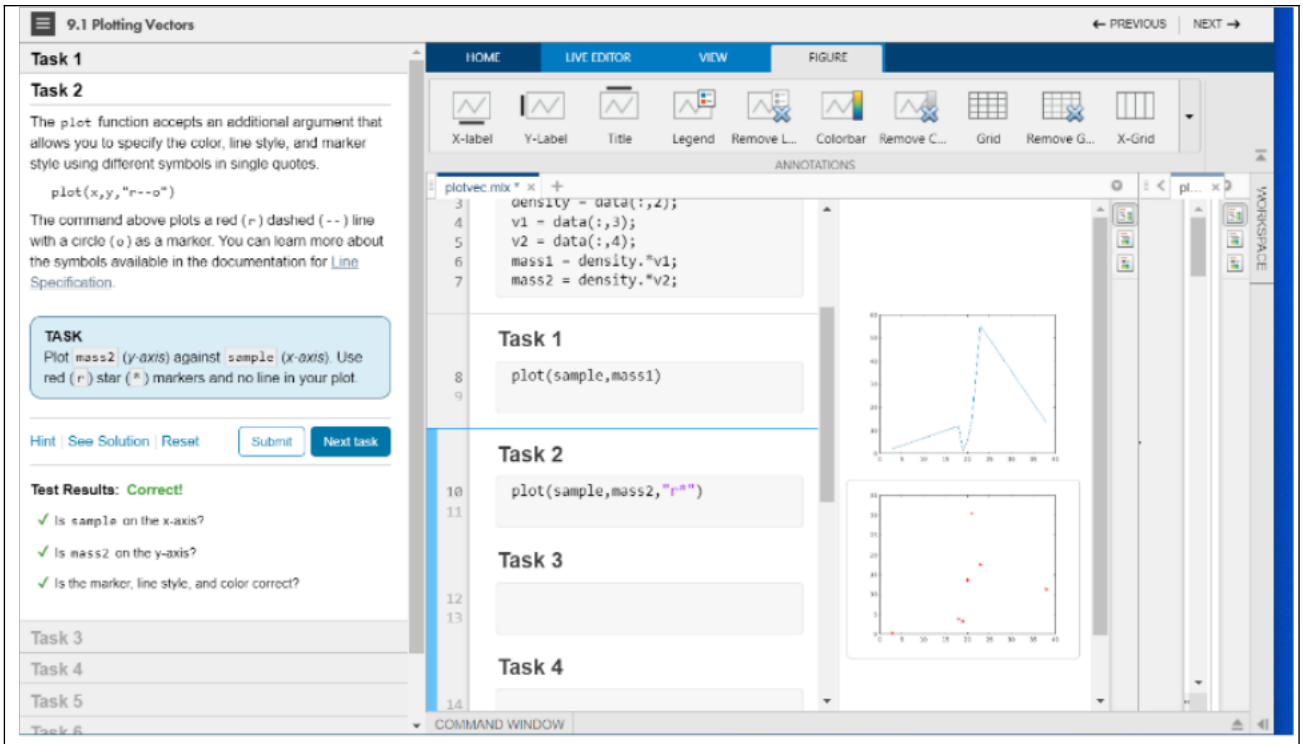
DAILY ASSESSMENT FORMAT

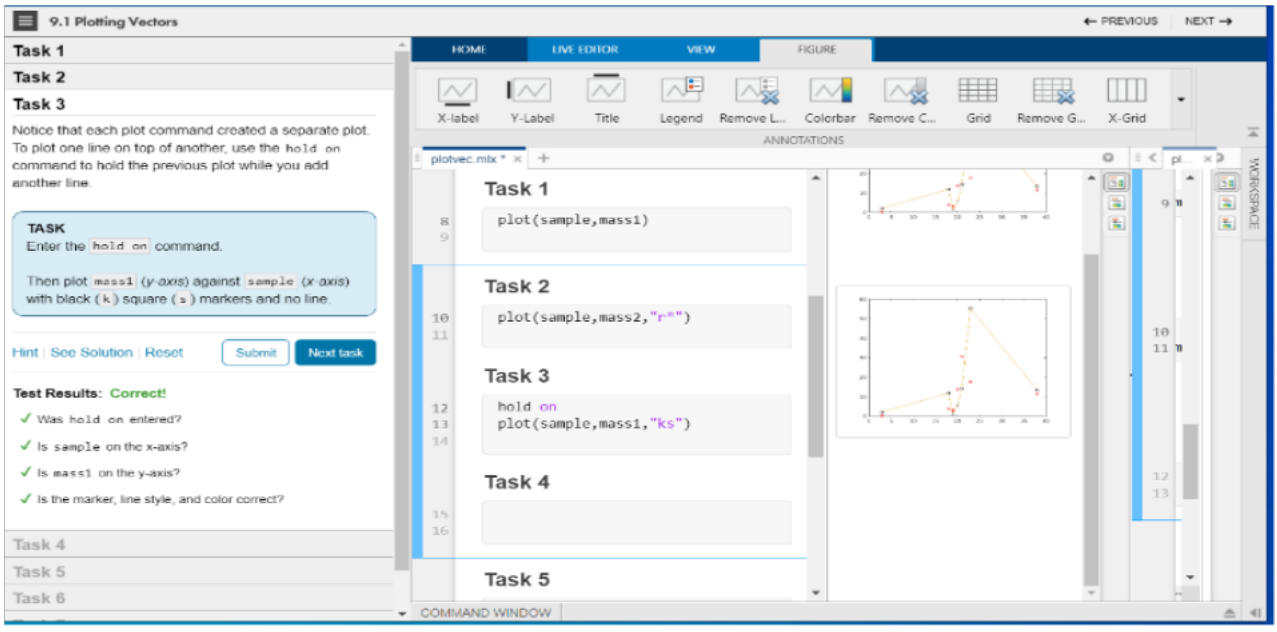
Date:	08-07-2020	Name:	POOJA K S
Course:	Mat lab Onramp	USN:	4AL17EC070
Topic:	1. Calling Functions 2. Obtaining Help 3. Plotting Data 4. Review Problems 5. Importing Data 6. Logical Arrays 7. Programming 8. Final Project 9. Conclusion	Semester & Section:	6th SEM and 'B' section
Github Repository:	pooja-shivanna		

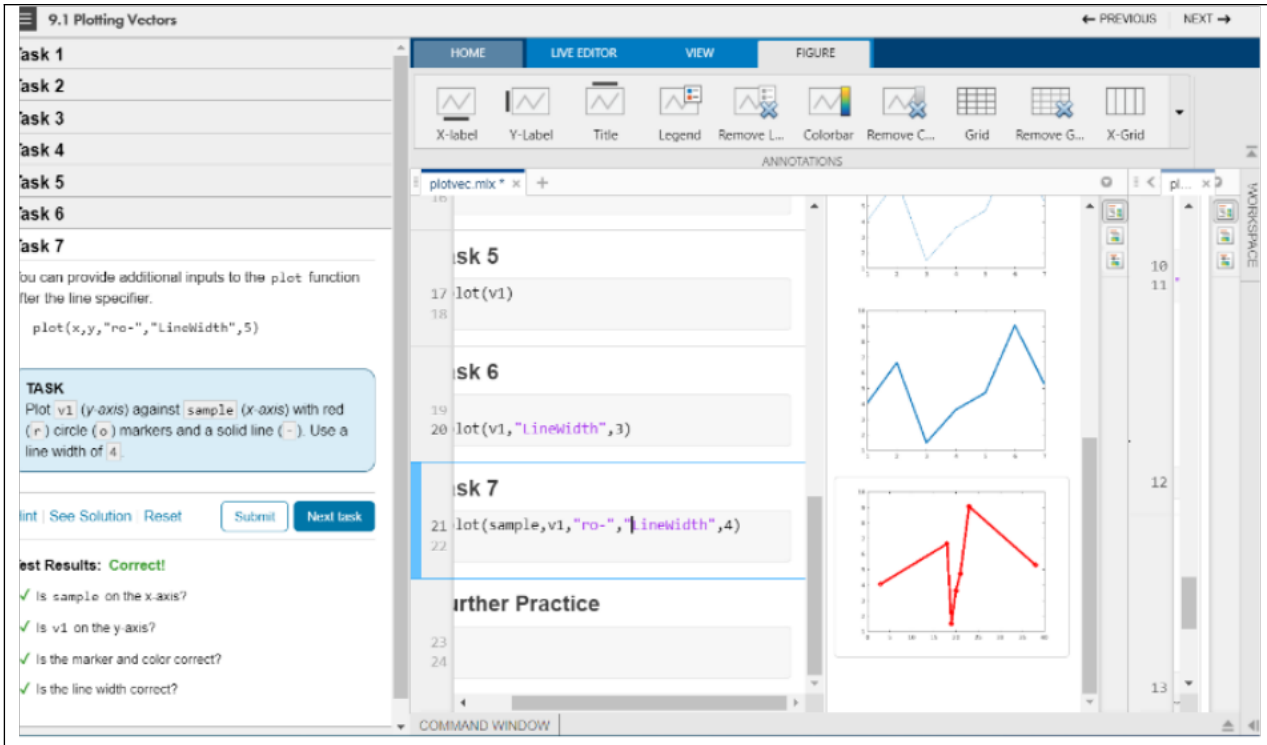
FORENOON SESSION DETAILS











MAT Lab :

MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

As of 2020, MATLAB has more than 4 million users worldwide. MATLAB users come from various backgrounds of engineering, science, and economics.

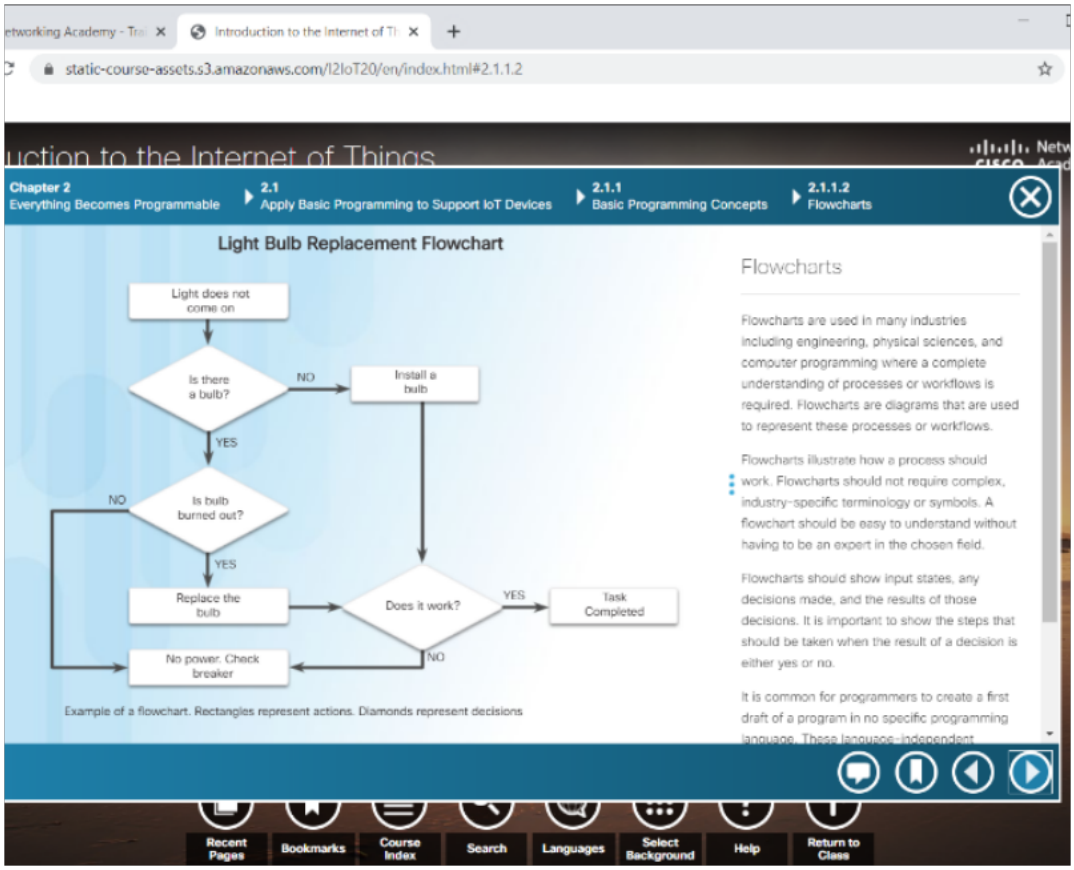
Variables

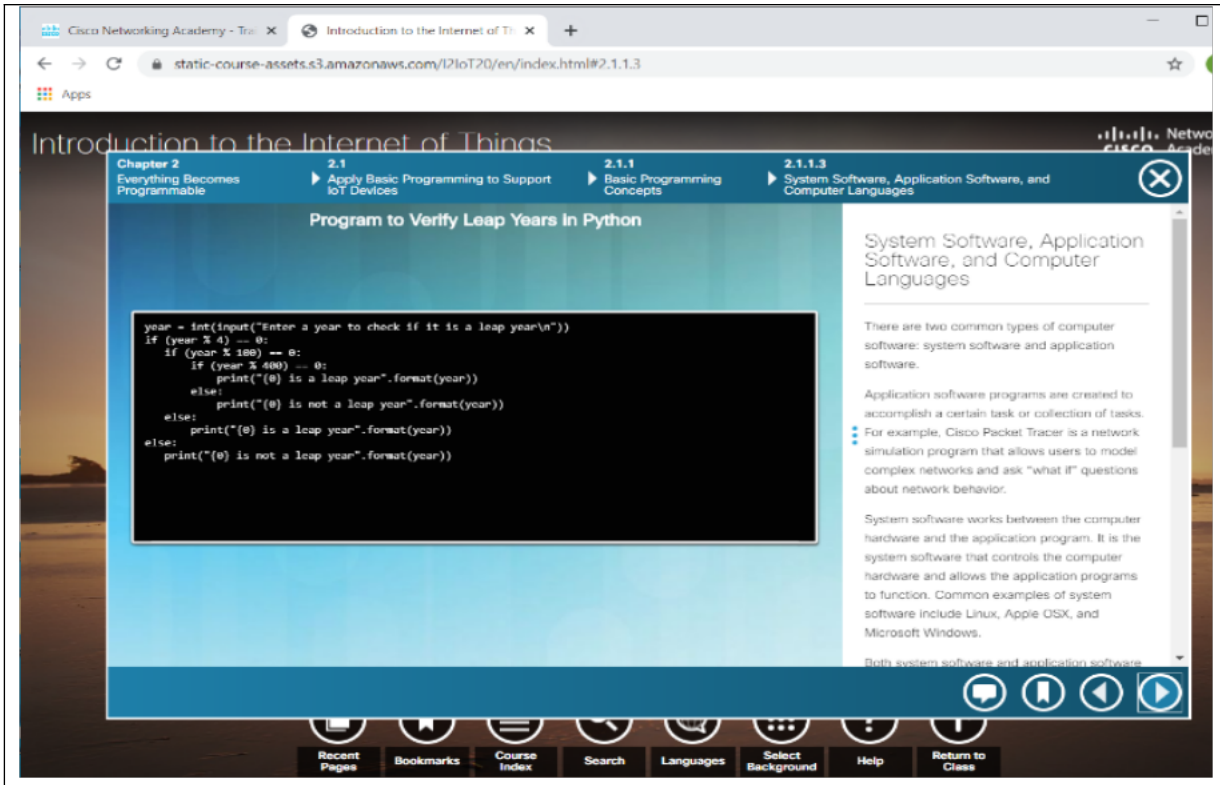
Variables are defined using the assignment operator, =. MATLAB is a weakly typed programming language because types are implicitly converted. It is an inferred

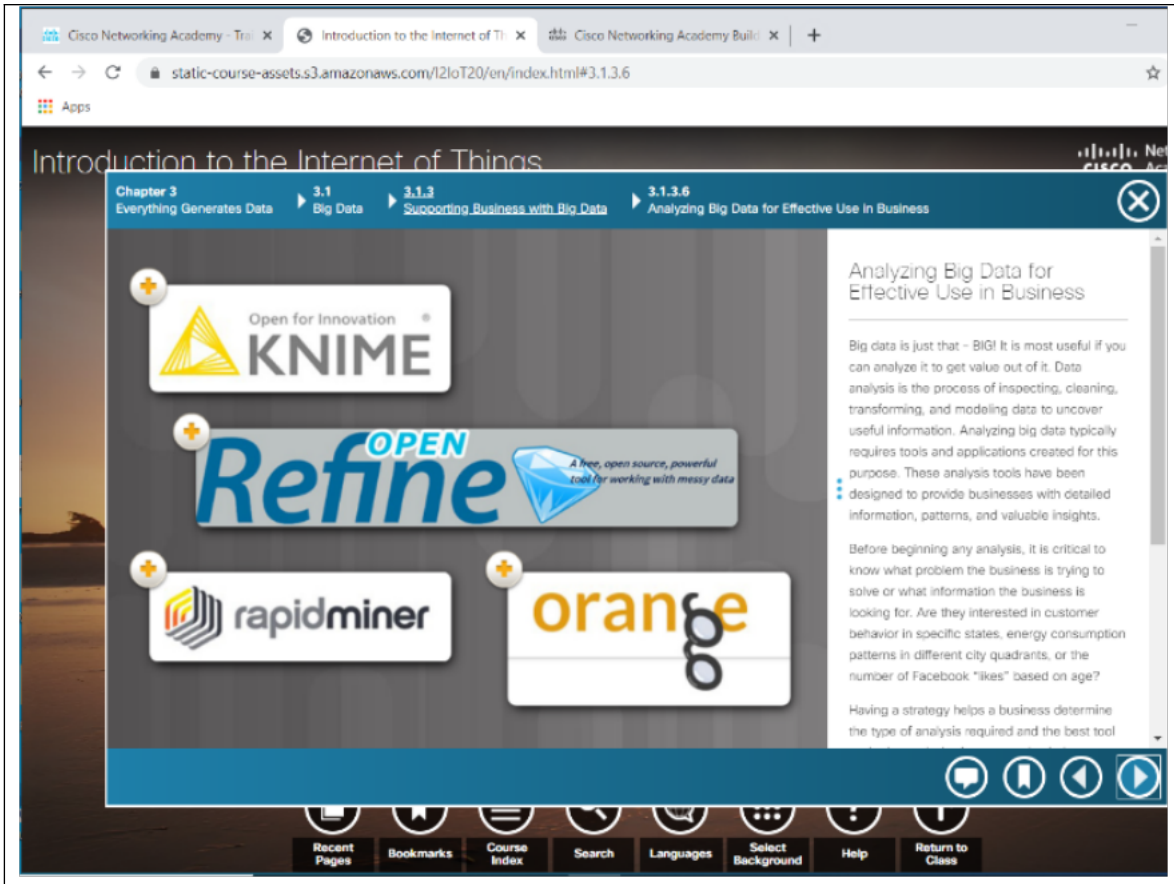
typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function. For example:

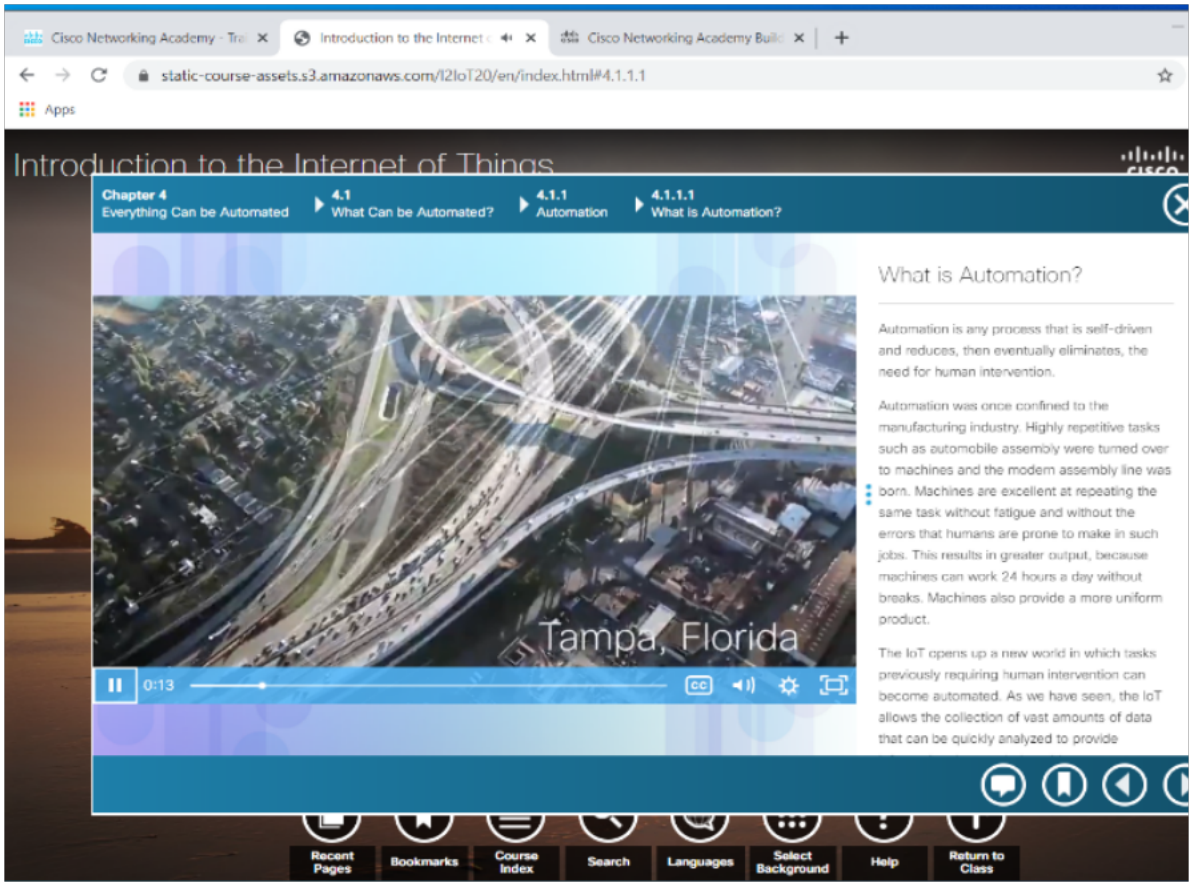
CERTIFICATE - MATLAB

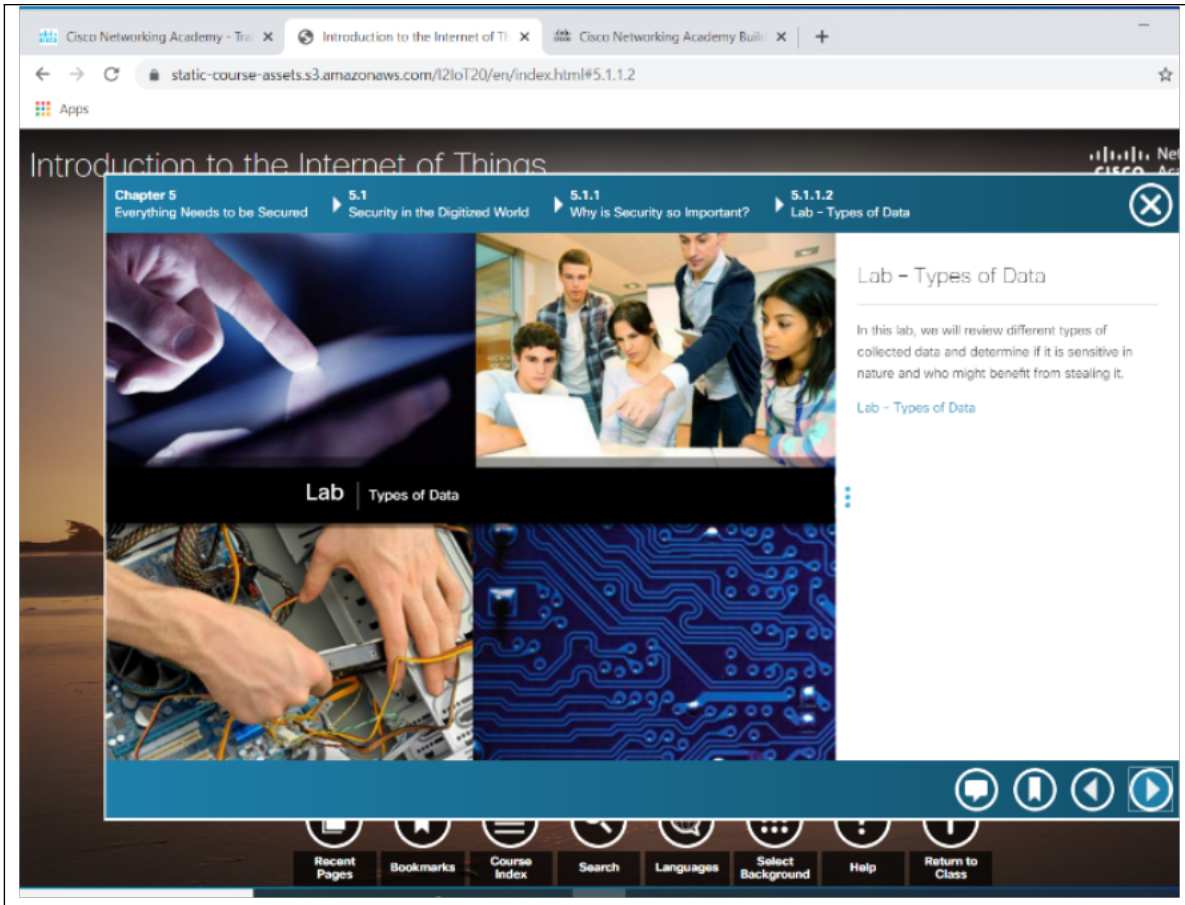
Date:	08-07-2020	Name:	POOJA K S
Course:	Introduction to Internet of Things	USN:	4AL17EC070
Topic:	1.Chapter 2 2.Chapter 3 3.Chapter 4 4.Chapter 5 5.Chapter 6	Semester & Section:	6th SEM & 'B' section











What is the IoT?

The Internet of Things (IoT) is the connection of millions of smart devices and sensors connected to the Internet. These connected devices and sensors collect and share data for use and evaluation by many organizations. These organizations include businesses, cities, governments, hospitals and individuals. The IoT has been possible, in part, due to the advent of cheap processors and wireless networks. Previously inanimate objects such as doorknobs or light bulbs can now be equipped with an intelligent sensor that can collect and transfer data to a network.

Lab - Create a Process Flowchart

Flowcharts are normally used to diagrammatically illustrate the process flow before a computer program is created. In this lab you will create a simple flowchart showing the process used to find a predetermined integer value.

What is Blockly?

Blockly is a visual programming tool created to help beginners understand the concepts of programming. By using a number of block types, Blockly allows a user to create a

program without entering any lines of code.

Blockly implements visual programming by assigning different programming structures to coloured blocks. The blocks also contain slots and spaces to allow programmers to enter values required by the structure. Programmers can connect programming structures together by dragging and attaching the appropriate blocks. Programming structures such as conditionals, loops, and variables are all available for use.

Creating a new variable in Blockly is a simple matter of dragging the variable block onto the work space and filling in the value slot. It is also possible to change the contents of a variable as the program is being executed.

Blockly also supports functions. Similar to the variables, Blockly has specific blocks to represent functions. Also similar to variables, programmers simply select and drag function blocks to the work space and fill in the required slots.

Notice in Figures 1 and 2 that the variable block and the print on screen block both have a bevel tab on the bottom and a slot on the top. This means that the two blocks can be snapped together to create a program sequence. Blockly will execute the block on the top first, then move on to the block below it.

Other blocks are available such as an IF THEN block, a WHILE block and a FOR block. There are also blocks specifically for sensors and actuators.

Blockly can be used to translate the block-based code into Python or JavaScript. This is very useful to beginner programmers.

Blockly Games

Google provides a series of free and open source educational games that can help you learn programming. The series is called Blockly Games.

There are a number of levels to complete to help you get started. Blockly may look like a toy, but it is a great tool to improve your logical thinking skills, which is one of the building blocks of computer programming.

Packet Tracer - Blinking an LED Using Blockly

Cisco Packet Tracer has incorporated Blockly as one of the programming languages available in its IoT functionality. In this lab you will control the blink rate of an LED using Blockly code.

What is Python?

Python is a very popular language that is designed to be easy to read and write. Python's developer community adds value to the language by creating all types of modules and making them available to other programmers.

The core philosophy of the language is summarized by the document The Zen of Python:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Despite the fact Python is designed to be easy, there is still a learning curve. To make it easier to learn Python, a beginner can use blocky to enhance his or her Python understanding.

While different programming languages have different semantics and syntax, they all share the same programming logic. Beginners can use Blackly to easily create a language-independent program, export it as Python code and use this newly created code to learn about Python syntax, structure and semantics.

The Python Interpreter

Python is an interpreted language; therefore, an interpreter is required to parse and execute Python code. The Python interpreter understands and executes Python code. Python code can be created in any text editor and Python interpreters are available for many operating systems. Python developers can create and deploy Python programs in practically any operating system. Third party tools such as Py2exe and Pyinstaller can also be used to package the Python source code into an executable file, eliminating the need for the Python interpreter when running Python code.

In Linux machines, the Python interpreter is usually installed in /usr/bin/python or /usr/bin/python3 (depending on the available Python versions on the system). With the new Windows Python installer, Python is installed by default into the user's home directory. In older Windows machines, Python is often placed in C:\PythonXX (where XX is the version of Python). After the Python interpreter has been installed, it operates somewhat like the Linux shell. This means that when called with no arguments, it reads and executes commands interactively. When called with a file name argument or with a file as standard input, it reads and executes a script from

that file.

To start the interpreter, simply type python or python3 at the shell prompt.

Some legacy systems are still running on an older version of Python 2, but many new systems are moving to use the new Python version 3. Python's version is printed on the first line when the interpreter is launched (Figure 1). This course is built on Python 3 code.

When the Python interpreter is called with no arguments, and commands are entered via the keyboard, the interpreter is said to be in interactive mode. In this mode, the interpreter waits for commands. The primary prompt is represented by three greater-than signs (>>>). Continuation lines are represented by three dots (...). Continuation is the default secondary prompt.

The >>> prompt indicates the interpreter is ready and waiting commands.

Continuation lines are needed when entering multi-line code.

Another way of using the interpreter is python -c command [arg] ... which executes the statement(s) in the command. Because Python statements often contain spaces or other characters that are particular to the shell, it is suggested to enclose the entire command between single quotes.

Useful Functions and Data Types in Python

Python supports many useful functions and datatypes. Some of the more important ones are as follows:

Range()

The range () function generates a list of numbers usually used to iterate with FOR loops. Figure 1 shows examples of the range () function.

- Range (stop) - This is the number of integers (whole numbers) to generate, starting from zero.
- Range ([start], stop [, step] – This is the starting number of the sequence, the ending number in the sequence, and the difference between each number in the sequence.

Tuples

A tuple is a sequence of unchangeable Python objects. Tuples are sequences, separated by parentheses. Figure 2 shows examples of tuples.

Lists

Lists are a sequence of changeable Python objects. Lists can be created by putting different comma-separated values between square brackets. Figure 3 shows examples of lists and how they can be updated.

Sets

Sets are unordered collections of unique elements. Common uses include membership testing, removing duplicates from a sequence, and computing standard math operations on sets such as intersection, union, difference, and symmetric difference. Figure 4 shows examples of sets.

Dictionary

A dictionary is a list of elements that are separated by commas. Each element is a combination of a value and a unique key. Each key is separated from its value by a colon. The entire dictionary is written within braces. Dictionary elements can be accessed, updated, and deleted. There are also many built-in dictionary functions such as a function that compares elements within different dictionaries and another that provides a count of the total number of elements within a dictionary. Figure 5 shows examples of dictionaries.

What is Big Data?

Data is information that comes from a variety of sources, such as people, pictures, text, sensors, and web sites. Data also comes from technology devices like cell phones, computers, kiosks, tablets, and cash registers. Most recently, there has been a spike in the volume of data generated by sensors. Sensors are now installed in an ever growing number of locations and objects. These include security cameras, traffic lights, intelligent cars, thermometers, and even grape vines!

Big Data is a lot of data, but what is a lot? No one has an exact number that says when data from an organization is considered “Big Data.” Here are three characteristics that indicate an organization may be dealing with Big Data:

- They have a large amount of data that increasingly requires more storage space (volume).
- They have an amount of data that is growing exponentially fast (velocity).
- They have data that is generated in different formats (variety).

How much data do sensors collect? Here are some estimated examples:

- Sensors in one autonomous car can generate 4,000 gigabits (Gb) of data per day.
- An Airbus A380 Engine generates 1 petabyte (PB) of data on a flight from London to Singapore.
- Safety sensors in mining operations can generate up to 2,4 terabits (TB) of data every minute.
- Sensors in one smart connected home can produce as much as 1 gigabyte (GB) of information a week.

While Big Data does create challenges for organizations in terms of storage and analytics, it can also provide invaluable information to fine-tune operations and improve customer satisfaction.

What is Automation?

Automation is any process that is self-driven and reduces, then eventually eliminates, the need for human intervention.

Automation was once confined to the manufacturing industry. Highly repetitive tasks such as automobile assembly were turned over to machines and the modern assembly line was born. Machines are excellent at repeating the same task without fatigue and without the errors that humans are prone to make in such jobs. This results in greater output, because machines can work 24 hours a day without breaks. Machines also provide a more uniform product.

The IoT opens up a new world in which tasks previously requiring human intervention can become automated. As we have seen, the IoT allows the collection of vast amounts of data that can be quickly analyzed to provide information that can help guide an event or process.

As we continue to embrace the benefits of the IoT, automation becomes increasingly important. Access to huge amounts of quickly processed sensor data started people thinking about how to apply the concepts of machine learning and automation to everyday tasks. Many routine tasks are being automated to improve their accuracy and efficiency.

Automation is often tied to the field of robotics. Robots are used in dangerous conditions such as mining, firefighting, and cleaning up industrial accidents, reducing the risk to humans. They are also used in such tasks as automated assembly lines.

We now see automation everywhere, from self-serve checkouts at stores and automatic building environmental controls, to autonomous cars and planes. How many

automated systems do you encounter in a single day?

Become an Informed Consumer

The last few years have given us improvements in the speed and availability of Internet services, as well as advances in cloud computing and sensor technology. These technical gains, together with recent developments in automation and artificial intelligence, have created a highly digitized world. Digitization currently impacts every aspect of our daily lives. Digitization continues to provide new opportunities for professionals who are trained to develop and support the technology that is used to deliver the IoT.

The IoT provides an immeasurable amount of information that is readily available for consumption. This information can be quickly analysed and used to automate many processes that were previously considered impossible to turn over to machines. For example, just a few years ago self-driving cars existed only in our imaginations and now they are a reality. Think about what else has changed in your life because of the IoT.