SCHEME OF EVALUATION
(Scoring Indicators)

Revision: 21 **Course Code: 6131A**

Course Title: INTERNET OF THINGS

Qst. No.	Scoring Indicator	Split up score	Sub Total	Total
	PART A			
1	• Things-refer to variety of devices. • Thing in a network can monitor/measure---Eg temperature sensor. • Things are capable of exchanging data with other connected devices in system. Data could be stored in cloud, processed there and a control action could be initiated.			1
2	Amazon webservices, adafruit, azure.	½ x 2 =1		1
3	Bluetooth low energy (BLE) is an open low energy short range radio communication technology it is also called as Bluetooth Smart.			1
4	Two protocols that enable transport are the 1) Traditional Bluetooth Low Energy 2) Light Fidelity (Li-Fi)	½ x 2 =1		1
5	Service level agreements provide details and information about the services being provided and the real value that a customer gets out of them.	Any two ½ x 2 =1		1
6	Public, private, hybrid	Any two		1
7	If, if else, while, for loop	Any two		1
8	Packages help keep your code organized, make it easier to manage and maintain, and allow you to share your code with others.			1
9	• Machinery manufacturing • Medical equipment • Automotive electronics • Communication technology • Aviation	Any two		1
II 1.	IOT can be used to build applications for 1. Vehicle, asset, pet and human tracking. 2. Agriculture and resources including water 3. Energy sector 4. Safety, vigilance, security systems. 5. Defense 6. Waste management 7. Embedded and mobile applications 8. Healthcare applications 9. Telemedicine 10. Smart city applications	Any three		3

<table>
<tr><td>II
2.</td><td>1. security/ personnel safety
2. privacy
3. Data extraction with consistency from complex environments
4. Connectivity
5. Power requirements
6. Complexity involved
7. Storage</td><td>Any three</td><td></td><td>3</td></tr>
<tr><td>II
3.</td><td>Enabling technologies/devices fall under one of the following categories
1) Technologies that help in acquiring/sensing data
2) Technologies that help in analyzing/processing data
3) Technologies that help in taking control action
4) Technologies that help in enhancing security/privacy</td><td>Any three</td><td>3 x 1</td><td>3</td></tr>
<tr><td>II 4.</td><td><table><tr><td></td><td>MQTT</td><td>COAP</td></tr><tr><td>Underlying protocol</td><td>TCP(connection oriented)</td><td>UDP (Connectionless)</td></tr><tr><td>communication</td><td>M:N (many to many)</td><td>1:1 (one to one)</td></tr><tr><td>Power</td><td>Higher than CoAP. Lesser than other protocols.</td><td>Lowest. Consumes less power than MQTT, making it the best.</td></tr><tr><td>Model</td><td>Publisher/subscriber</td><td>Request-response (RESTful and not SOAPful)</td></tr></table></td><td>Any three 1 mark for each</td><td></td><td>3</td></tr>
<tr><td>II 5.</td><td>1. Software as a service
2. Platform as a service
3. Infrastructure as a service

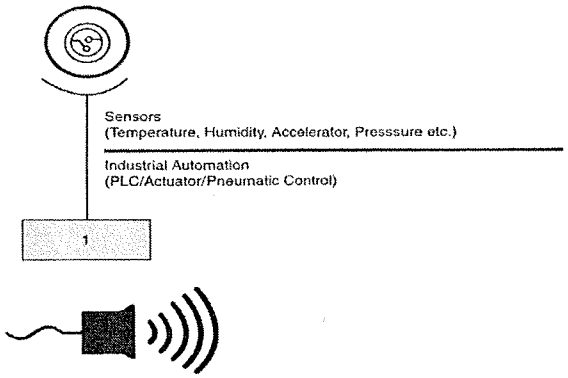
IaaS(Infrastructure as a Service)
It is a form of cloud computing that provides virtualized computing resources over the internet.
The users manage the machines, select the OS and underlying application and pay per their use.
PaaS (Platform as a Service)
The cloud service provider delivers hardware and software tools needed for application development to users over the Internet. A PaaS provider hosts the hardware and software on its own infrastructure Users have to build manage and maintain the applications as per their requirement
SaaS (Software as a Service)
A complete software application is provided to the user. This service can be availed by paying a monthly, yearly etc subscription</td><td>1 mark for each</td><td></td><td>3</td></tr>
<tr><td>II
6</td><td>1. Software security
2. Infrastructure security
3. Storage security</td><td>Any three</td><td></td><td>3</td></tr>
</table>

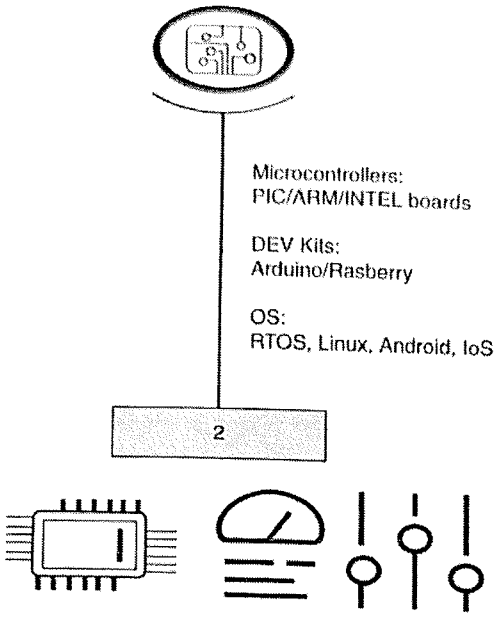
	4. Network security	1 mark for each		
II 7.	The benefits of using fog computing model are 1. Minimal amount of data sent to the cloud. 2. Reduced bandwidth consumption. 3. Reduced data latency 4. Improved data security 5. Immediate processing of data in real time.	Any three		3
II.8	Sensors convert a physical device into an electrical impulse to take the desired action. They obtain the parameters of a physical object, the output of their observation is converted into resistance, capacitance, impedance, etc. Types of Sensors 1. Temperature: Beginning with the most common type of sensor, the temperate sensor records the amount of heat in a given setting. It can be a machine, a room, a car, a lab, etc. This information can be used to take the desired action, like changing the temperature to optimal settings. The same can be automated according to some specific environmental conditions and settings. 2. Moisture: Where temperature sensors record the heat, moisture sensors record the amount of humidity. They have a wide array of applications in the environment, food supply chains, medicinal labs, agriculture, etc. Moisture sensors either have a hair tension moisture element or a psychrometer to record the moisture content. 3. Light: Light sensors record and assess the ambient light settings in a defined area and recommend actions to change the same. In your smartphone, when the brightness is adjusted according to the exposure to light, the light sensor and the electrical actuator play their part. In the modern homes that have automated light settings, these sensors are used. 4. Motion: Motion sensors are usually installed in security systems and help detect unauthorized activity. Upon sensing activity either by changes in the heat or weight, the sensor activates an alarm system sending notifications to the right people. Motion IoT sensors use radar, infrared, or ultrasonic waves to detect activity in their vicinity. 5. Noise: Noise sensors, as the name suggests, record the noise levels in the given environment. It can be an entire city, a room, a car, etc. In IoT, these sensors are used to build safe working and living environments for people. They are also used to send warning notifications to the right people when noise levels go beyond the stipulated threshold limit. , etc.	Definitio n-1 Types- 2		3

II.9	There are **three primary types of loops** that allow you to repeat a block of code multiple times:
	1. **For Loop**:
	o The for loop is used for **sequential traversal**. It iterates over a sequence of items such as a list, tuple, dictionary, or set.
	o The syntax for a for loop is as follows:
	Python
	`for variable in sequence:`
	`  # Code to execute`
	o Example:
	`fruits = ["apple", "banana", "cherry"]`
	`for fruit in fruits:`
	`  print(fruit)`
	Output:
	`apple`
	`banana`
	`cherry`
	2. **While Loop**:
	o The while loop is used to execute a block of statements **repeatedly until a given condition is satisfied**.
	o The syntax for a while loop is as follows.
	`while expression:`
	`  # Code to execute`
	o Example:
	`count = 0`
	`while count < 3:`
	`  count += 1`
	`  print("Hello")`
	Output:
	`Hello`
	`Hello`
	`Hello`
	o You can also use an else statement with a while loop. The else block is executed when the condition becomes false:
	`count = 0`
	`while count < 3:`
	`  count += 1`
		For loop-1mark
		While loop-2 mark
				3

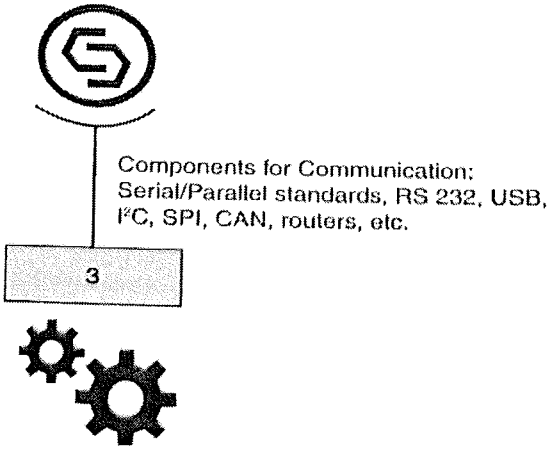
	print("Hello Geek")
	else:
	print("In Else Block")
	
	.
	
	Output:
	
	Hello
	Hello
	Hello
	In Else Block
	3. **Infinite While Loop**:
	o If you want a block of code to execute an **infinite number of times**, you can use a while loop with a condition that is always true.
	o However, it is **not recommended** to use this type of loop, as it can lead to an infinite loop. You would need to forcefully terminate the program.
	o Example:
	
	Python
	
	count = 0
	while count == 0:
	print("Hello")			

II.10	A Python module is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code.
	Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.
	Example:
	Let's create a simple calc.py in which we define two functions, one add and another subtract.
	# A simple module, calc.py
	def add(x, y):
	return (x+y)
	
	def subtract(x, y):
	return (x-y)
	Importing modules in Python Example
	Now, we are importing the calc that we created earlier to perform add operation.
	
	# importing module calc.py
	import calc
		Module-2 mark
		
		Example -1
				3

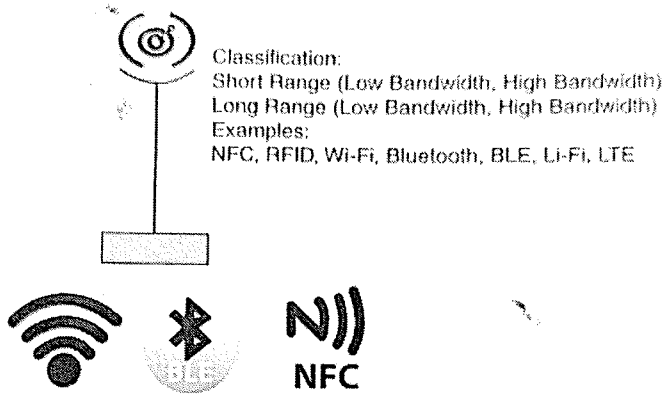
	print(calc.add(10, 2))			
III.1	IOT stack It consist of 7 layers 1. Layer 1 (Physical or sensor layer) This layer is concerned about the physical components which include sensors. Example temperature sensor, pressure sensor, humidity sensor etc. In industrial automation actuator, PLC are regarded as physical layer components. This layer is responsible for data collection.  2. Layer 2 (Processing and control action layer) This layer contains core components for IOT. Example microcontrollers, processors. Microcontrollers receive data from the sensors. Variety of development kits available are Arduino, NodeMCU. Data collected by sensors are processed in this layer.	Listing-1 Any three layers 2 marks each		7



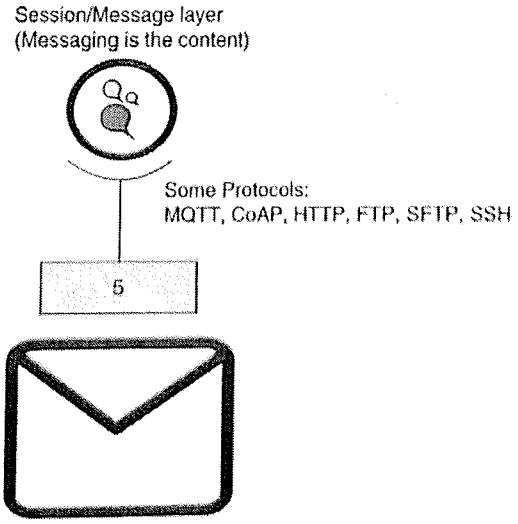
3. Layer 3 (Hardware interface layer)
   This layer contains the hardware components and communication standards such as RS232, routers etc.



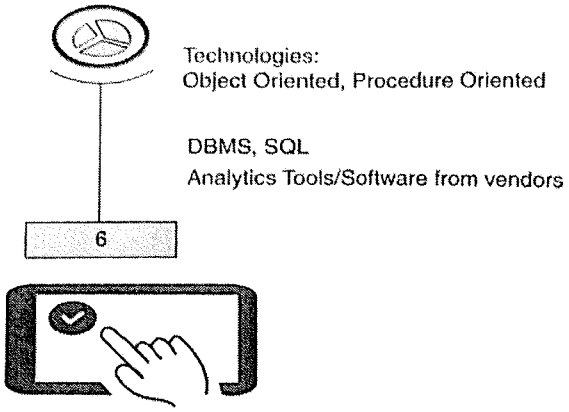
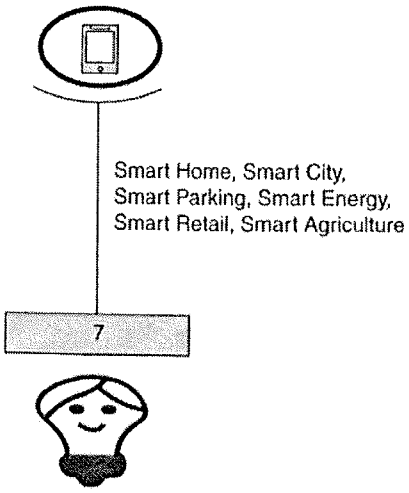
4. Layer 4 (RF Layer)
   The protocols used for communication and transport of data based on RF is listed in this layer. Example Wi-Fi, NFC, RFID, Bluetooth. This layer also include Li-Fi which are alternates for RF protocols.

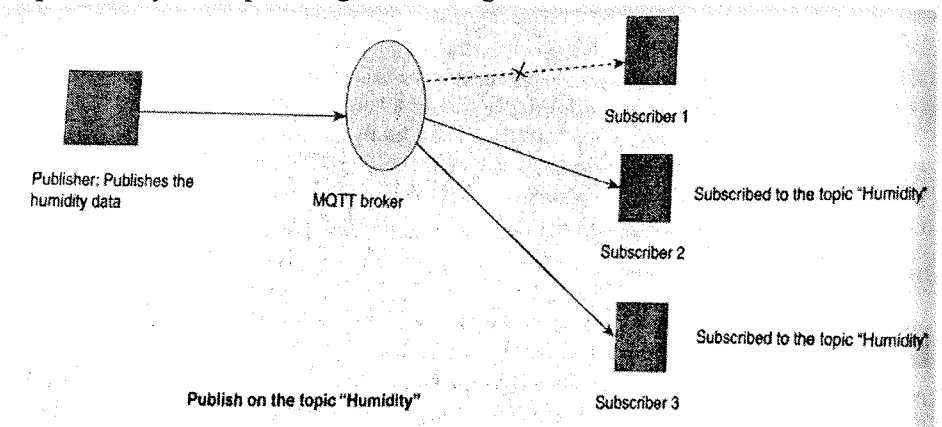


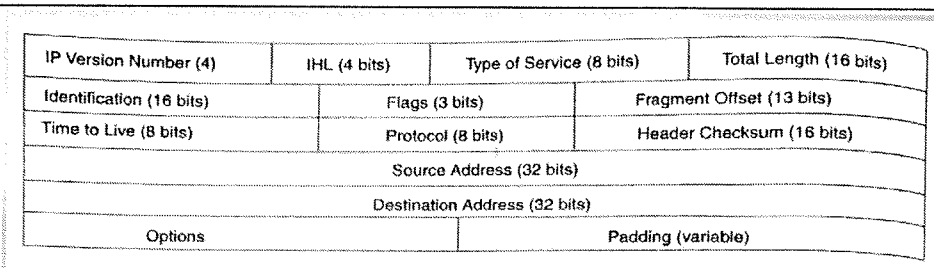
5. Layer 5 (Session/Message layer)
   Session management includes how messages are broadcasted to cloud. Messaging protocols such as MQTT,COAP are used in this layer. Other protocols are SSH and FTP.



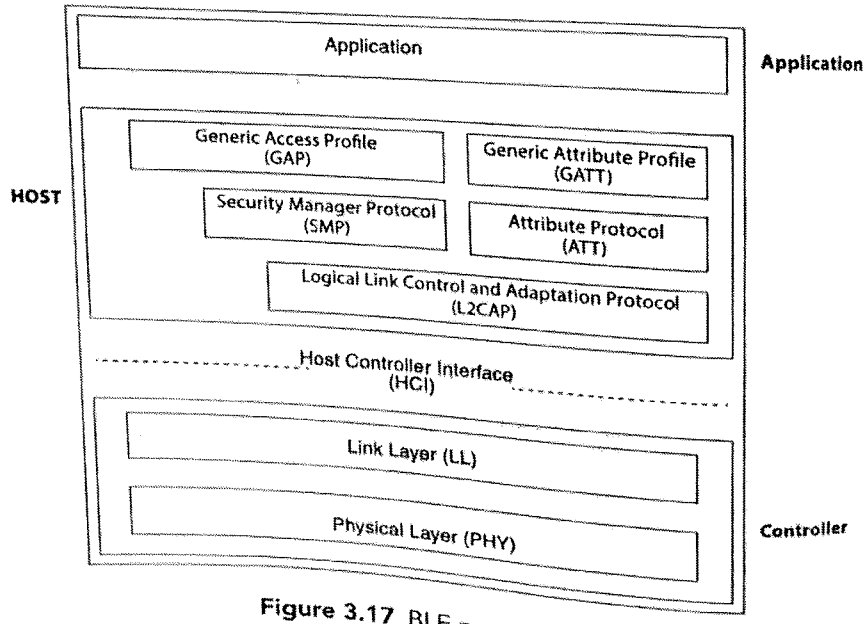
6. Layer 6 (User experience layer)
   This layer is concerned with the end user experience. Object oriented programming languages, scripting languages, analytics tools are included in this layer.

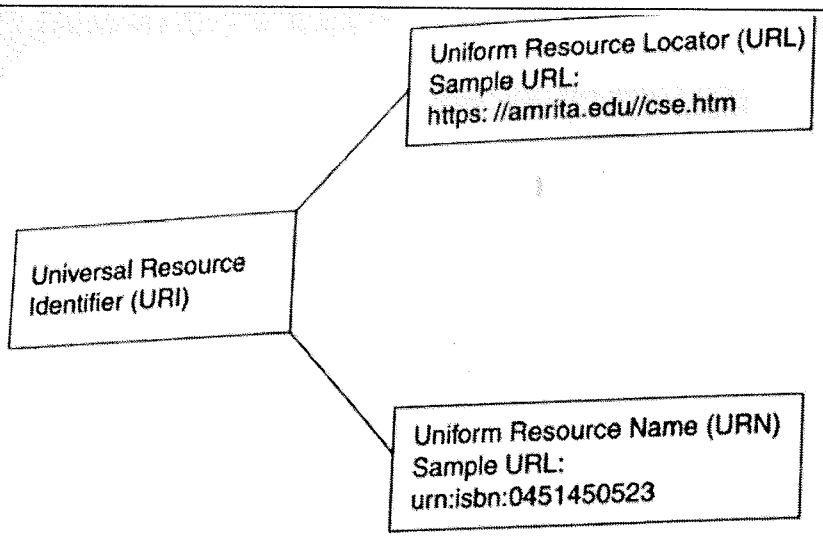
	 7. Layer 7 (Application layer) This layer includes the possible applications that can be built with the support of the rest of the layers. It ranges from a simple automation application to smart city application. 			
III.2	Various levels in IOT Level 1 Minimal complexity and easiest to build The application has one sensor- a device to sense Data sensed stored and processed locally Monitoring/control done through application Level 2 Complex than level 1 Huge data, cloud storage is preferred Frequency of sensing is fast Analysis carried locally Based on data analysis the control application can be triggered through web application/mobile application Level 3 Data is voluminous, frequency of sensing faster, data stored in cloud Here analysis also carried out in cloud.	Listing-1 mark Explanation of any three levels -6 mark		7

	Áction can be triggered through mobile application/web application Level 4 Data is voluminous, frequency of sensing faster, data stored in cloud Multiple nodes are present which are independent of each other These sensor nodes upload data to cloud Data is huge Analysis carried on cloud Áction can be triggered through mobile application/web application Level 5 Amount of data is extensive and is sensed much faster Multiple independent nodes When an application is completely cloud oriented it is computationally intensive in real time			
III.3	MQTT is a Lightweight Protocol. It demands minimal resources for its functioning. Follows public-subscribe pattern PUBLISH-SUBSCRIBE PATTERN • Central broker plays a key role • It helps to dispatch messages to the nodes that have subscribed for the same • Publisher sends the message to the broker and the broker takes the responsibility of dispatching the messages to the subscribed destinations.  **Figure 3.1** MQTT scheme: publish – subscribe pattern. Publisher (node) can send the data to the MQTT broker • Based on subscriptions from the nodes connected, the broker sends the messages accordingly • Only subscribers 1 and 3 have subscribed for the humidity data • Data captured by nodes that have expressed interest through subscription • The data reaches the node that really wants it (not all nodes in network) Messages are published as topic and the publisher publishes the messages with a topic • In the example discussed above the topic is humidity. • Clients subscribe to the topic and they get the messages based on their subscription. • In this protocol clients do not have addresses. • Messages are transferred based on topics.	Figure -2 Explanation-5	2+5	7

	• Messages are sent as a topic and brokers filter the messages based on the topics subscribed which are then circulated to respective subscribers.			
III.4	 **Figure 4.4** IPv4 protocol format. IP version number • It is a half a byte field (4bits) which indicated the version of IP being utilized • This field has value 4 in binary(0100) • Internet Header Length • Specifies internet header length and points to beginning of data • The minimum value for a header is 5(0101) • Type of service • 8 bit field which indicates quality of service • QoS can be precedence, delay and reliability • The type of service field is composed of 8 bits 1. Precedence: bits 0 to 2 represent precedence 2. 2. Delay: it's the 4th bit. Here 0 indicates normal delay and 1 indicates lower delay 3. 3. Throughput: 0 indicates normal throughput, 1 indicates high throughput 4. 4. Reliability: 0 indicates normal reliability and 1 indicates high reliability Total Length: This is a 16 bit field which defines the length of the IPv4 datagram. • This length includes header and data • The minimum length of the IP datagram is 20 bytes and the maximum can be 65,535 bytes • Identification • 16 bit field added by the sender to help in assembling the fragments • Flags: • Composed of three bits • The first bit is always zero and is kept unused. • Second bit is Don't Fragment (DF). • If DF is set to 0 the IP datagram can be fragmented, and if set to 1 it cannot be fragmented • Next bit is called More Fragments (MF) which if set will indicate that more fragments are on the way. Fragment Offset: When a fragmentation of a message occurs, this field specifies the offset or the position in the overall message where the data in this fragment is present. • It is specified in units of 8 bytes • The first	Format -4 marks Explanation-3		7

	fragment has an offset of 0 • Time to Live: An 8-bit field which indicates the time that IP datagram will survive • There will be a time set that will be decremented by 1. • When it reaches 0, the datagram will be discarded by router • Protocol: • This serves as an identifier for the higher layer protocol carried in the IP datagram • The protocols can be ICMP, IGMP, TCP or UDP • Header Checksum: • This is a 16-bit checksum computed over the header to provide basic protection against corruption in transmission • Source address: • This is the 32-bit IP address of the source of the datagram Destination Address • This is the 32-bit IP address of the destination which receives the IP datagram • Options • There are a lot of optional header settings available and they are used for the debugging/testing and security purposes • Padding • IP options field may vary in length. The padding field provides additional bits so that the total header length is an exact multiple of 32 bits.			
III.5	BLE PROTOCOL STACK CONTROLLER • Consists of Host Controller Interface (HCI), link layer and physical layer 1) Host Controller Interface is used to enable interoperability between hosts and controllers assembled by different manufactures 2) Link Layer (LL) defines packet structure 3) Physical Layer(PHY) takes care of transmission and reception, modulation and demodulation. It also responsible for analog to digital conversion HOST • Generic Access Profile (GAP) takes care of device discovery, connection establishment, connection management and security. This layer mandatory for all devices • Generic Attribute Profile (GATT) oversees the data exchange process. Whenever there is a need to push data and to read or write data, the generic guidelines with respect to process are governed by GATT • Attribute Protocol (ATT) is the protocol for accessing data • Logical Link Control and Adaption Protocol (L2CAP) is responsible for fragmentation and defragmentation of the application data. It administers the multiplexing and demultiplexing of channels over the shared logical link • Security Manager (SM) handles pairing, authentication, and encryption. • Host Controller Interface (HCI) The functionality remains the same as that in the controller. It is used to enable interoperability between hosts and controllers assembled by different manufacturers. APPLICATION LAYER • Similar to application layer of OSI reference model • Top most layer in BLE • Responsible for user interface(UI), logic and data handling	Figure- 3 mark Explanation – 4 marks	3+4	7

	 **Figure 3.17** BLE protocol stack.			
III.6	PROTOCOL FOR IDENTIFICATION: UNIFORM RESOURCE IDENTIFIER • Uniform Resource Identifier(URI) is a sequence of characters used to identify logical resources. • All guidelines with respect to URI are issued by the IETF • URI has Uniform Resource Name(URN)/ Uniform Resource Locator(URL) in it • URIs try to identify a resource and location by accessing its address and name through a primary access mechanism • URN defines an items identity whereas URL renders a method for finding it • URL contains information on how to fetch or acquire a resource from its location • URL always begin with a protocol(http) • It will have details of host name • In addition it will contain a path • A URL is used when a client raises a request to the server for the service • Uniform Resource Name (URN) gives the name of a resource and its identification.	Figure-2 marks Explanation- 5 marks		7

	 **Figure 4.9** Uniform Resource Identifier.			
III.7	1. Privacy and security 2. bandwidth cost 3. migration and portability 4. Availability, Reliability and Robustness 5. costing 6. data ownership 7. expertise	Listing-1 mark Explanation – 6 marks		7
III.8	Criteria 1: Certification and standards compliance Criteria 2: Financial health of the service provider Criteria 3: Business and technology strength Criteria 4: compliance audit Criteria 5 : service level agreements Criteria 6: reporting/tracking Criteria 7: costing and billing Criteria 8: maintenance monitoring and upgrade Criteria 9: support Criteria 10: security	Listing-1 Explanation of any three – 6marks		7
III.9	Raspberry Pi is known as a single board computer. It is a computer constructed from a single printed circuit board. Raspberry pi is small, about the size of a credit card. Raspberry pi boards come in various models with different specifications. They all have GPIO (General Purpose Input/Output) pins that allow users to connect and control external devices such as sensors, motors and cameras. They can run various operating systems, including the Linux based Raspberry pi OS. They can be used for a wide range of projects, from retro gaming consoles to home automation systems.			7

	1. Central processing unit 2. HDMI port 3. Graphic processing unit 4. Memory (RAM) 5. Ethernet port 6. SD card slot 7. General purpose input and output (GPIO)pins 8. LEDs 9. USB ports. 10. Power source.			
III.10	1. **Smart Agriculture** Food is an integral part of life without which we cannot survive. However, it is an unfortunate fact that a lot of food is wasted in developed countries like America while people starve in poorer countries like Chad, Sudan, etc. One way to feed everyone is through better agricultural practices which can be enhanced using IoT applications. This can be done by first collecting data for a farm such as soil quality, sunlight levels, seed type, and rainfall density from various sources like farm sensors, satellites, local weather stations, etc. and then using this data with Machine Learning and IoT to create custom recommendations for each farm that will optimize the planting procedure, irrigation levels required, fertilizer amount, etc. All this will result in better yield or crops with a focus on reducing world hunger in the future. This is done very efficiently by SunCulture, a top IoT application, which is an initiative by Microsoft AI for Earth. 2. **Smart Vehicles** Smart vehicles or self-driving cars are Iot applications as they can be called are pretty dependent on IoT. These cars have a lot of features that are integrated with each other and need to communicate such as the sensors that handle navigation, various antennas, controls for speeding or slowing down, etc. Here the Internet of Things technology is critical, especially in the sense that self-driving cars need to be extremely accurate and all the parts need to communicate with each other in milliseconds on the road. Tesla Cars are quite popular and working on their self-driving cars. Tesla Motors' cars use the latest advancements in Artificial Intelligence and the Internet of Things. And they are quite popular as well!!! Tesla Model 3 was the most sold plug-in electric car in the U.S. in 2018 with a total yearly sales of around 140,000 cars. This top IoT application has gained a lot of advancement in recent years 3. **Smart Home** Maybe one of the most famous applications of IoT is in Smart Homes. After all, who hasn't heard about connecting all the home applications like lighting, air conditioners, locks, thermostat, etc. into a single system that can be controlled from your smartphone? These IoT devices are applications of IoT and becoming more and more popular these days because they allow you complete freedom to personalize your home as you want. In fact, these IoT devices are so popular that every second there are			7

	127 new devices connected to the internet. Some popular ones that you might have heard have, or even have in your home, include Google Home, Amazon Echo Plus, Philips Hue Lighting System, etc. There are also all sorts of other inventions that you can install in your home including Nest Smoke Alarm and Thermostat, Foobot Air Quality Monitor, August Smart Lock, etc. These applications of IoT are getting famous nowadays. 4. **Smart Pollution Control** Pollution is one of the biggest problems in most of the cities in the world. Sometimes it's not clear if we are inhaling oxygen or smog! In such a situation, IoT applications can be a big help in controlling pollution levels to more breathable standards. This can be done by collecting data related to city pollution like emissions from vehicles, pollen levels, airflow direction, weather, traffic levels, etc using various sensors in combination with IoT. Using this data, Machine Learning algorithms can calculate pollution forecasts in different areas of the city that inform city officials beforehand where the problems are going to occur. Then they can try to control the pollution levels till it's much safer. An example of this is the Green Horizons project created by IBM's China Research Lab. 5. **Smart Healthcare** There are many applications of IoT in the Healthcare Industry where doctors can monitor patients remotely through a web of interconnected devices and machines without needing to be in direct contact with them. This is very useful if the patients don't have any serious problems or if they have any infectious diseases like COVID-19 these days. One of the most common uses of IoT applications in healthcare is using robots. These include surgical robots that can help doctors in performing surgeries more efficiently with higher precision and control. There are also disinfectant robots that can clean surfaces quickly and thoroughly using high-intensity ultraviolet light (which is pretty useful these days!) Other types of robots also include nursing robots that can handle the monotonous tasks that nurses have to perform for many patients day in and day out where there is little risk to the patients.			
III.11	Features and characteristics of Arduino. Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. Key components of an Arduino board are • Microcontroller - this is the brain of an Arduino, responsible for executing programmed instructions. • USB port -Facilitates connection to a computer for programming the board and powering it simultaneously. • USB to Serial chip - Helps translating data that comes from e.g. a computer to the on board microcontroller. This is what makes it possible to program the Arduino board from the computer.			7

	• Digital pins - pins that use digital logic (0,1 or LOW/HIGH). Commonly used for switches and to turn on/off an LED. • Analog pins - pins that can read analog values in a 10 bit resolution (0-1023). • 5V / 3.3V pins- these pins are used to power external components. Object 1 • GND - also known as ground, is used to complete a circuit, where the electrical level is at 0 volt. • VIN - stands for Voltage In, where you can connect external power supplies			
III.12	The following are the standard or built-in data types in Python: • Numeric • Sequence Type • Boolean • Set • Dictionary • Binary Types(memoryview, bytearray, bytes) Numeric Data Types in Python The numeric data type in Python represents the data that has a numeric value. A numeric value can be an integer, a floating number, or even a complex number. These values are defined as Python int, Python float, and Python complex classes in Python. • **Integers** – This value is represented by int class. It contains positive or negative whole numbers (without fractions or decimals). In Python, there is no limit to how long an integer value can be. • **Float** – This value is represented by the float class. It is a real number with a floating-point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation. • **Complex Numbers** – A complex number is represented by a complex class. It is specified as *(real part) + (imaginary part)j*. For example – 2+3j Sequence Data Type in Python The sequence Data Type in Python is the ordered collection of similar or different data types. Sequences allow storing of multiple values in an organized and efficient fashion. There are several sequence types in Python – • Python String • Python List • Python Tuple List Data Type Lists are just like arrays, declared in other languages which is an ordered collection of data. It is very flexible as the items in a list do not need to be of the same type. **Creating a List in Python**	Listing-1 mark Any three datatypes 6 marks		7

	Lists in Python can be created by just placing the sequence inside the square brackets[]			