Lecture 2

1. **Describe the architecture of a smart object in IoT.**
   Explain the function of each component (sensors, actuators, processing unit, radio transceiver) and how they work together to form a smart object.

2. **Differentiate between smart sensors and conventional sensors.**
   Provide examples to illustrate how smart sensors add computational and communication capabilities.

3. **Summarize the key differences between microcontroller-based boards and single-board computers used in IoT prototyping.**
   Discuss with reference to architecture, operating system support, and typical applications.

4. **Explain the internal architecture and basic functionalities of the Arduino UNO board.**
   Include discussion on analog input resolution, I/O pins, and the role of setup() and loop() functions.

5. **Illustrate the key challenges in embedded systems programming specific to IoT devices.**
   Focus on aspects such as constrained resources, memory limitations, and real-time requirements.

6. **Apply your knowledge of accelerometer sensors to explain how they are used to measure motion and orientation in IoT applications.**
   Use the principles of capacitive sensing and inertial displacement to justify your explanation.

7. **Analyze a project requirement and decide whether Arduino or Raspberry Pi is more suitable.**
   Justify your choice based on project complexity, OS needs, and interfacing requirements.

8. **Demonstrate the working of the I2C protocol with a block diagram showing master-slave communication.**
   Explain how addressing, clock synchronization, and data transfer occur in a two-wire setup.

9. **Use Pulse Width Modulation (PWM) principles to simulate analog output using digital pins in an Arduino system.**
   Explain the relationship between duty cycle and effective voltage output.

10. **Identify various domains (e.g., wearable tech, underwater monitoring, smart homes) where smart sensors are applied, and relate the choice of sensor type to environmental and functional requirements.**
    Provide real-world examples in each case.

Lecture 3

1. **Explain the functional components of an IoT architecture with examples.**
   Discuss each component's role in enabling communication, processing, and integration.
2. **Describe how RFID technology works and highlight its advantages over traditional barcode systems.**
   Include the structure of RFID tags and the role of readers.
3. **Compare and contrast the features of Bluetooth Low Energy (BLE) and ZigBee protocols used in IoT.**
   Focus on range, power consumption, and typical use cases.
4. **Discuss the key challenges in IoT network architecture design.**
   Explain aspects such as interoperability, scalability, energy efficiency, and bandwidth management.
5. **Illustrate the structure and function of MQTT in IoT communication.**
   Describe the publish-subscribe mechanism, roles of brokers, and topic hierarchy.
6. **Given a smart home scenario, select and justify the most appropriate communication protocol (BLE, ZigBee, or Wi-Fi) for sensor communication.**
   Include reasoning based on range, power, and data rate requirements.
7. **Apply the concept of Secure MQTT (SMQTT) to explain how encryption is achieved in message broadcasting.**
   Describe the setup, encryption, publish, and decryption phases.
8. **Design a lightweight IoT application using the MQTT protocol.**
   Specify components such as publishers, subscribers, topics, and a real-world use case (e.g., temperature monitoring).

Lecture 4

1. **Explain the core principles of REST architecture used in IoT systems.**
   Include discussion on statelessness, resource representation, and standard HTTP methods.

2. **Describe the role of HTTP status codes, methods, and headers in RESTful APIs.**
   Illustrate how these elements help ensure effective communication and error handling.
3. **Explain the structure and message flow in the CoAP protocol.**
   Include message types (CON, NON, ACK, RST) and how reliability is achieved.
4. **Compare REST and CoAP protocols in terms of communication model, transport layer, and suitability for constrained environments.**
5. **Discuss the publish/subscribe model in MQTT and its advantages in IoT systems.**
   Highlight the role of topics, brokers, and decoupling of message senders and receivers.
6. **Apply RESTful design principles to construct a URL structure for a smart parking API.**
   Ensure the API supports CRUD operations and includes filtering, sorting, and versioning.
7. **Given a scenario of battery-powered sensors in a smart agriculture setup, justify the use of CoAP over HTTP.**
   Consider payload size, transport reliability, and energy efficiency.
8. **Design an MQTT-based smart home system where multiple devices (e.g., lights, thermostats) communicate through topics.**
   Include examples of topic hierarchies and wildcard subscriptions.
9. **Demonstrate how different QoS levels in MQTT affect message delivery in critical vs. non-critical applications.**
   Use examples like fire alarms (QoS 2) vs. temperature logging (QoS 0).
10. **Construct a comparison table of REST, CoAP, and MQTT protocols based on parameters such as transport layer, architecture type, use case domain, and efficiency.**
    Analyze which protocol is optimal under which condition.