CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA281	INTERNET OF THINGS	ELECTIVE	3	1	0	4

Preamble: This course intends to provide insight into new innovations that will build novel type of interactions among things and humans, and enables the realization of smart cities, infrastructures, and services for enhancing the quality of life and utilization of resources. An overview of IOT and its related concepts, different IOT architectures and their components, emerging paradigms such as Fog computing, Platforms and solutions supporting development and deployment of IOT applications, message passing mechanisms such as RPC, REST, and CoAP, data and knowledge management, data confidentiality, data integrity, and operation control issues faced by IOT are included in the course.

Prerequisite: Basic concepts of Information Technology and Internet.

Course Outcomes: After completion of the course the student will be able to

CO No:	Course Outcome (CO)	Blooms Category Level
CO 1	Describe the main concepts and features of the IOT paradigm.	Level 2: Understand
CO 2	Discuss Fog computing, TinyOS - nesC and programming frameworks for IOT	Level 2: Understand
CO 3	Describe the data management techniques applied to the IOT environment.	Level 2 Understand
CO 4	Explain security, and privacy in IOT environments	Level 2 Understand
CO 5	Discuss key enablers and solutions to enable practical IoT systems	Level 2 Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3						2					
CO 2	3	1					2					
CO 3	3	1					2					
CO 4	3	1					2					
CO 5	3	1	1				2					

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	20	20	20
Understand	30	30	40

Apply			
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 8 marks
Continuous Assessment Test (2 numbers) : 20 marks
Assignment/Quiz/Course project : 12 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have a maximum of 2 subdivisions and carry 6 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare SOA-based architecture and API-oriented architecture.
2. Neatly sketch the open IOT architecture for IOT/CLOUD convergence.
3. List and explain the applications of device/cloud collaboration.

Course Outcome 2 (CO2)

1. What are the advantages associated with Fog computing?
2. Comment on the four broad requirements that motivate the design of TinyOS.
3. Summarize the communication paradigms and technologies used in resource-constrained environments.

Course Outcome 3(CO3):

1. Explain stream and stream processing in IOT.
2. Write and explain the algorithm for distributed anomaly detection by clustering ellipsoids.
3. Discuss the general architecture of a stream-processing system in IOT.

Course Outcome 4 (CO4):

1. Give an overview on the security requirements of IOT.
2. How can you nullify the impact of fault in high-availability cluster?
3. Explain the BCK with pre-shared keys for TinyTO.

Course Outcome 5 (CO5):

1. Give an overview on the Wired Gateway Interfaces.
2. List the features to select the gateway hardware.
3. List the steps to prepare Raspberry Pi for the execution.

Model Question Paper
Course Code: 20MCA281

Course Name: INTERNET OF THINGS

Max. Marks :60 **Duration: 3 Hrs**

Part A

Answer all questions. Each question carries 3 marks (10 * 3 = 30 Marks)

1. What do you mean by computation offloading?
2. Explain the framework that enables collaboration between smart mobile devices and cloud.
3. Outline the major challenges faced in the Fog paradigm.
4. Explain Polyglot Programming.
5. Which are the challenges faced by stream-processing systems?
6. Explain anomaly detection and categorize anomalies in the data.
7. List the different ways that an IOT gateway can extend connectivity to nodes.
8. Write the advantages of obfuscation and diversification techniques.
9. Explain Inter-Integrated Circuit (I^2C) or Two Wire Interface (TWI).
10. Write a short note on Zigbee.

Part B

Answer all questions. Each question carries 6 marks. (5 * 6 = 30 Marks)

11. Explain the taxonomy of Resource Management in IOT. (6 Marks)

OR

12. Draw and explain the state diagram of the open IOT services life cycle. (6 Marks)

13. a. Comment on the four broad requirements that motivate the design of TinyOS (3 Marks)

    b. Describe the design decisions for nesC. (3 Marks)

OR

14. List the features in coordination languages - Linda, eLinda, Orc, and Jolie (6 Marks)

15. Compare Stream Management System (DSMS) and Complex Event Processing (CEP). (6 Marks)

OR

16. Describe hyper ellipsoidal model for anomaly detection. (6 Marks)

17. Describe the error detection techniques which are applicable in the context of an IOT. (6 Marks)

OR

18. Explain the Station-to-Station protocol (STS) and the two main shortcomings of STS. (6 Marks)

19. Discuss the sensors required to build the environmental-sensing IoT gateway device for weather monitoring. (6 Marks)

OR

20. List and explain the six steps for the development of a sensor project. (6 Marks)

Syllabus

Module 1 (9 Hours)
Overview of Internet of Things: Open-source semantic web infrastructure for managing IOT resources in the Cloud - Device/Cloud Collaboration framework for intelligence applications.
Module 2 (11 Hours)
Introduction to Fog Computing: principles, architectures, and applications. TinyOS – NesC, Programming frameworks for Internet of Things
Module 3 (8 Hours)
Stream processing in IoT: foundations, state-of-the-art, and future directions - A framework for distributed data analysis for IoT
Module 4 (9 Hours)
Security and privacy in the Internet of Things- Internet of Things - robustness and reliability. TinyTO: two-way authentication for constrained devices in the Internet of Things - Obfuscation and diversification for securing the Internet of Things

Module 5 (8 Hours)
Creating a simple IoT project - Preparing Raspberry Pi – Interfacing the hardware - Internal representation of sensor values- Persisting data - Creating the actuator project - Creating a controller.

More detailed knowledge may be acquired through seminars, assignments and talks by eminent external experts and also by implementing a micro project.

Any one of the following or similar micro projects may be given as part of the course.

1. Smart Gas Leakage Detector
2. Night Patrol at home

Text Books

1. RajkumarBuyya; Amir VahidDastjerdi , "Internet of Things", Morgan Kaufmann, 2016

Reference Books

1. Peter Waher, "Learning Internet of Things", Packt Publishing, 2015
2. S. SitharamaIyengar; Nandan Parameswaran; Vir V. Phoha; N. Balakrishnan; Chuka Okoye, "Fundamentals of Sensor Network Programming: Applications and Technology", Wiley, December 14, 2010
3. Robert Stackowiak, Art Licht, VenuMantha, Louis Nagode, "Big Data and The Internet of Things: Enterprise Information Architecture for A New Age", Apress, 2015

Web Resources

1. https://www.coursera.org/specializations/internet-of-things
2. http://web.mit.edu/professional/digital-programs/courses/IoT

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction (9 Hours)**	
1.1	Internet of things- definition, evolution. Applications -Smart home applications, Health care, Elder care, Traffic surveillance.	1
	SOA -Based Architecture, API oriented Architecture, Resource Management.	1
	Computational Offloading, Identification and Resource/Service Discovery, IOT Data Management and Analytics, IOT and the CLOUD	1

1.2	Open IOT architecture for IOT/Cloud convergence, Sensor middleware, Cloud computing infrastructure, Directory service, Global Scheduler, Local Scheduler component, Service delivery and utility manager	1
	Workflow of open IOT platform, Scheduling process and IOT Services lifecycle, State diagram of the Open IOT Services lifecycle within the scheduler module	1
	Scheduling and resource management, Resource optimization schemes, Caching technique	1
	Service creation flowchart, Comparison of cost - with cache server and public cloud data-score	1
1.3	Runtime adaptation engine, Device/cloud collaboration framework	1
	applications of device/cloud collaboration, Semantic QA cache	1
2	**Programming frameworks (11 Hours)**	
2.1	Introduction to Fog Computing: principles, architectures, and Applications	1
	Motivating scenario for Fog Computing, Advantages of Fog Computing, Reference architecture of Fog Computing	1
	Software-Defined Resource management layer, Services of Software-Defined Resource management layer, Applications of Fog Computing.	1
2.2	History of TinyOS, Implementation, Requirements motivating the design of TinyOS, Component Model, Interfaces. TinyOS computational concepts	1
	Overview of TinyOS Execution Model, Concurrency, TinyOS Theory of Execution: Events & Tasks, TinyOS Architecture. TinyOS-Programming Model.	1
2.3	nesC design, Component Implementation, Design Decisions for nesC, Module Components, Configuration Components	1
	Whole-Program Analysis, Detecting Race Conditions, Dealing with Race Conditions, Issues for nesC.	1
2.4	Overview of Embedded Programming Languages- nesC, Keil C, Dynamic C, B#, Message Passing in Devices-Remote Procedure Call (RPC), Lightweight RPC (LRPC)	1
	Representational state transfer (REST), Computational REST (CREST), Constrained Application Protocol(CoAP), Comparison of HTTP and CoAP, Advantages of CoAP	1

	Coordination Languages- Orchestration, Choreography, Linda and eLinda, Orc, Features of Orc, Java Orchestration Language Interpreter Engine (Jolie), Polyglot Programming, Inverse pyramid for Polyglot Programming.	1
	Features of programming frameworks for IOT, IOT programming approaches, Existing IOT frameworks	1
3	**Data management techniques (8 Hours)**	
3.1	Stream, Stream Processing, Data Stream Management System (DSMS)	1
	Complex Event Processing (CEP), differences between two use-cases of Stream Processing: DSMS and CEP	1
	The characteristics of stream data in IOT, general architecture of a stream-processing system in IOT	1
	Continuous logic processing system, challenges in stream-processing systems.	1
3.2	Anomaly detection, problem statement and definitions	1
	Hyper ellipsoidal anomaly detection	1
	Distributed anomaly detection	1
	Clustering ellipsoids, incremental local modeling	1
4	**Security and privacy (9 Hours)**	
4.1	IOT security threats, IOT security requirements, security frameworks for IOT, IOT security overview, IOT gateways and security, IOT routing attacks	1
	Security frameworks for IOT - Lightweight cryptography, asymmetric LWC algorithms, privacy in IOT networks	1
4.2	IOT characteristics and reliability issues, reliability challenges	1
	Addressing reliability, security aspects and solutions	1
4.3	TinyTO: Two-way authentication for constrained devices in the Internet of Things	1
	TinyTO protocol	1
	BCK with pre-shared keys for TinyTO, handshake implementation	1
4.4	IOT network stack and access protocols**,** Obfuscation and diversification techniques	1

	Enhancing the security in IOT using obfuscation and diversification techniques, motivations and limitations, different use-case scenarios on software diversification and obfuscation.	1
5	**IoT Implementation (8 Hours)**	
5.1	Three key components to an IOT architecture, Sensor to gateway communication - wired gateway interfaces, wireless gateway interfaces	1
	Sensors - sensors required to build the environmental-sensing IOT gateway device for weather monitoring	1
	Gateway, Gateway hardware, Gateway software	1
	Data transmission - advanced message queuing protocol, backend processing, to CLOUD or not to cloud	1
5.2	Creating a simple sensor project - Preparing Raspberry Pi – Clayster libraries	1
	Hardware, Interfacing the hardware - Internal representation of sensor values- Persisting data	1
	External representation of sensor values, Exporting sensor data	1
	Creating the actuator project – Hardware, Interfacing the hardware, Creating a controller	1