BACHELORS WITH INFORMATION TECHNOLOGY AS MAJOR SEMESTER 4th

BIT422J1: Information Technology: OOPS Using C++ **Credits: Theory=4, Practical=2**

UNIT-I: Introduction to OOP and C++.
Introduction to object oriented approach (OOA) and object oriented programming (OOP), concept of object and class.
Features of OOP-Encapsulation, Abstraction, Inheritance and Polymorphism. Advantages of OOP over structured programming.
Introduction to C++ with general basic features of operators and control structure (if, if-else, switch-case, while, do-while, for, etc.)

UNIT-II: Classes, Objects and Functions.
Classes-specifying class, defining member functions and member variables, scope resolution operator, access specifiers and accessing class members, friend class, static class members.
Objects-Dynamic allocation operators (New and Delete), arrays of objects, object as function argument and functions returning objects, object assignment.
Functions-Inline functions, friend functions. Default arguments, reference variables.
Constructors-parameterized, multiple constructors in a class, copy constructor, destructor. Function overloading, operator overloading (unary, binary).

UNIT-III: Inheritance, Pointers and Polymorphism.
Inheritance-Defining derived classes. Inheritance types-single, multilevel, multiple, hierarchical and hybrid inheritance. Virtual base classes.
Pointers-Pointer to objects, this pointer, pointer to derived class.
Polymorphism-Virtual functions, pure virtual functions, abstract classes.
Exception handling-Introduction, the keywords (try, catch and throw), multiple catch statements, catching all exceptions, rethrowing an exception.

Practical=2 credits

The Practical Component shall be based on the Unit-I to Unit-III

Books:

1 "Object oriented programming with C++" by E Balaguruswamy
2 "The complete reference C++" by Herbert Schildt
3 "Object oriented programming in C++" by Robert Lafore

BACHELORS WITH INFORMATION TECHNOLOGY AS MAJOR SEMESTER 4th

BIT422J2**:** Information Technology: **Operating Systems** **Credits: Theory=4, Tutorial=2**

UNIT - I

Operating Systems Overview: Introduction, Types of operating systems, Applications of operating system, process management, memory management, storage management, protection and security, distributed systems.

Operating Systems Structures: Operating system services and systems calls, system programs, operating system structure.

UNIT - II

Process and Process Management: Process concepts, process states, process control block, process scheduling, scheduling algorithms– FCFS, Shortest Job First, SRTF, Round Robin scheduling, Introduction to Inter-process communication.

Concurrency and Process Synchronization: Concept of threads, Concept of multithreading, Process synchronization, critical section problem, Peterson’s solution, semaphores, classic problems of synchronization–readers and writers’ problem, dining philosopher problem, monitors, atomic transactions.

UNIT - III

Deadlocks: System model, deadlock characterization, deadlock prevention, detection and avoidance, recovery from deadlock banker’s algorithm.

Memory Management: Swapping, contiguous memory allocation, paging, structure of the page table, segmentation, virtual memory, demand paging, page-replacement algorithms.

UNIT IV

File System: Concept of a file, access methods, directory structure, file system mounting, file sharing, protection.

Mass Storage Structure: overview of mass storage structure, disk structure,
disk scheduling algorithms.

Tutorial=2 credits

The Tutorial Component shall be based on the Unit-I to Unit-IV

References:

Abraham Silberschatz, Greg Gagne, and Peter B. Galvin, “Operating System Concepts”, Wiley Publication.

BACHELORS WITH INFORMATION TECHNOLOGY AS MAJOR SEMESTER 4th

BIT422J3: Information Technology: Software Engineering **Credits: Theory=4, Tutorial=2**

UNIT-I

Introduction & Software Development Process Modelm Definition of Software, Type of Software, Characteristic of Software, Attributes of Good Software, Definition of Software Engineering. Software Engineering Costs, Key Challenges that Software Engineering Faces. Software Process. Agile Process. Software Process Model: The Waterfall Model, Evolutionary Development, Incremental, Spiral Development.

UNIT-II

Requirements and Design SW requirements analysis, Types of requirements and steps involved in SRA, SW requirements Specifications.
Design Concept: Abstraction, Structured analysis and design, representation techniques used data modeling (ER Diagrams).Process Modeling (DFD), Behavioral modeling (State-transition Diagram)Top-down design, structural partitioning, characterization of effective modular design (functional independence, Cohesion, Coupling), SW architecture styles (data-centered, data-flow and layered architectures).

UNIT-III

Structured analysis and design and Testing Structure charts and data flow diagrams. Representation techniques used data modeling (ER Diagrams).Process Modeling (DFD), Behavioral modeling (State-transition Diagram)

Software Testing: Need for SW Testing, testing principle, approaches to the design test cases, black-box And white-box testing, Phases in testing activity: unit, integration, validation and system tests, concepts of verification and validation.

UNIT-IV

Software project management concepts Introduction to Reengineering and reverse engineering project planning and resource estimation techniques, simple Boehm model, risk analysis and management, project scheduling and tracking, software quality assurance. Introduction to CASE tools, categories of commonly used CASE Tools.

Tutorial=2 credits

The Tutorial Component shall be based on the Unit-I to Unit-IV

Books Recommended:

1. Software Engineering—Roger Pressman
2. Software Engineering—Ian Summerville

BACHELOR WITH APPLIED COMPUTING AS MINOR
4th SEMESTER

ACP422N APPLIED COMPUTING _ FUNDAMENTALS OF IOT

CREDITS: THEORY (3) PRACTICAL (1)

COURSE LEARNING OUTCOMES:

- *Understand the fundamental characteristics of IoT, including its physical design, basic components, and the concepts of things, sensing, and actuators.*
- *Explore various application areas of IoT such as home automation, smart cities, medical, logistics, environment, analytics, and smart grids.*
- *Gain insights into IoT protocols used for communication and data exchange within IoT ecosystems.*
- *Develop hands-on skills in working with hardware platforms like Raspberry Pi and Arduino, and learn how to implement basic sensors for monitoring temperature, humidity, proximity, gas, air quality, and ultrasonic sensors.*

UNIT - 1 (15 Hours)

Introduction: Definition & Characteristics of Iot, Physical Design of Iot, Basic Components of IoT, Thing, Sensing & Actuators, Vision, Physical Parameters. Iot Protocols.

Application Areas of IoT: Home Automation, Smart Cities, Medical, Logistics, Environment, Analytics. Smart Grids.

UNIT - 2 (15 Hours)

Iot Communication Models, APIs, IoT Architecture: Basic Architecture: 3 layer and 5-layer Architecture, ITU-IoT Reference Model, Machine to Machine Communication, IoT Gateways, Wireless Sensor Networks. Technologies: Bluetooth Low Energy (BLE), ZigBee: Architecture, Comparison with other wireless standards. LoraWAN.

UNIT - 3 (15 Hours)

Electronic Product Code (EPC), Near Field Comm. (NFC), 6LoWPAN, End to End Reliability: MQTT, SCADA. Hardware and Software Platforms: Hardware: Raspberry Pi, ESP8266 Wifi Module, Arduino. Implementation of Basic Sensors (temperature, humidity, proximity, gas, air quality, Ultrasonic sensors)

Internet of Things Privacy and Security Issues, Steps towards a security platform in IoT

Text Books:

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014. (ISBN-13: 978-8173719547)
2. Internet of Things (IoT), Dr. Kamlesh Lakhwani, Dr. Hemant Kumar Gianey, Joseph Kofi Wireko, Kamal Kant Hiran

References:

1. Schwartz, Marco. “Internet of Things with Arduino Cookbook”. Packt Publishing Ltd, 2016.
2. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014. (ISBN-13: 978-0124076846)
3. Hakima Chaouchi “The Internet of Things Connecting Objects to the Web” by Wiley publications

PRACTICAL (2 CREDITS: 30 HORS)

Learning Outcome*: These practical's cover a range of fundamental concepts in IoT, including hardware interfacing, sensor integration, data communication, cloud integration, and real-world applications. Students can gain hands-on experience and a solid understanding of the core principles of IoT through these practical exercises.*

1. Study the fundamental of IOT software's and components. Install Arduino IDE development platform.
2. Connect a microcontroller chip on a breadboard, establish power connections, and verify basic functionality. Gather the necessary components: microcontroller chip (e.g., Arduino Uno), breadboard, jumper wires, USB cable for power, and an LED.
3. To practice connecting analog input and output components to the microcontroller using the breadboard.
4. Write a program to Read sensor data using analog or digital pins.
5. LED Blinking Using Arduino: To interface LED/Buzzer with Arduino/Raspberry Pi/ ESP8266
6. and write a program to turn ON & OFF.
7. To interface Push button/Digital sensor (IR/LDR) with Arduino/ ESP8266/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection. Upload the program to the board and observe the blinking.
8. To interface DHT11 sensor with Arduino/Raspberry Pi/ ESP8266 and write a program to print temperature and humidity readings and Display sensor readings on the serial monitor.
9. IoT Cloud Platform Integration: Create an account on a popular IoT cloud platform ThingSpeak.
10. Write a program to Send temperature and humidity sensor data to the cloud and visualize it on a dashboard.
11. Write a Program to interface motor using relay with Arduino/Raspberry Pi/ ESP8266 and write a program to turn ON motor when push button is pressed.
12. Build a web server using Arduino and Ethernet/Wi-Fi shield using ESP8266 Wifi module. Develop a web page to control LEDs remotely.
13. Use buttons on the webpage to toggle LED states.
14. Log data at regular intervals to an SD card or cloud platform. Create graphs to visualize the logged data over time.
15. Set up an MQTT broker (e.g., Mosquitto) on a computer or cloud server. Program an Arduino to publish sensor data and subscribe to commands via MQTT.
16. Create a soil moisture sensing system using Arduino and moisture sensors. Send data to the cloud to monitor soil conditions remotely.
17. Design a basic home automation system with Arduino and relays.