MC4102 **OBJECT ORIENTED SOFTWARE ENGINEERING** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To understand the phases in object-oriented software development
- To gain fundamental concepts of requirements engineering and analysis.
- To know about the different approach for object-oriented design and its methods
- To learn about how to perform object-oriented testing and how to maintain software
- To provide various quality metrics and to ensure risk management.

UNIT I **SOFTWARE DEVELOPMENT AND PROCESS MODELS** **9**

Introduction to Software Development – Challenges – An Engineering Perspective – Object Orientation – Software Development Process – Iterative Development Process – Process Models – Life Cycle Models – Unified Process – Iterative and Incremental – Agile Processes.

UNIT II **MODELING OO SYSTEMS** **9**

Object Oriented Analysis (OOA / Coad-Yourdon), Object Oriented Design (OOD/Booch), Hierarchical Object-Oriented Design (HOOD), Object Modeling Technique (OMT) – Requirement Elicitation – Use Cases – SRS Document – OOA - Identification of Classes and Relationships, Identifying State and Behavior – OOD - Interaction Diagrams – Sequence Diagram – Collaboration Diagrams - Unified Modeling Language and Tools.

UNIT III **DESIGN PATTERNS** **9**

Design Principles – Design Patterns – GRASP – GoF – Dynamic Object Modeling – Static Object Modeling.

UNIT IV **SYSTEM TESTING** **9**

Software testing: Software Verification Techniques – Object Oriented Checklist:- Functional Testing – Structural Testing – Class Testing – Mutation Testing – Levels of Testing – Static and Dynamic Testing Tools - Software Maintenance – Categories – Challenges of Software Maintenance – Maintenance of Object Oriented Software – Regression Testing

UNIT V **SOFTWARE QUALITY AND METRICS** **9**

Need of Object Oriented Software Estimation – Lorenz and Kidd Estimation – Use Case Points Method – Class Point Method – Object Oriented Function Point – Risk Management – Software Quality Models – Analyzing the Metric Data – Metrics for Measuring Size and Structure – Measuring Software Quality - Object Oriented Metrics

SUGGESTED ACTIVITIES:

1. Discuss the different phases in any domain like Health Monitoring System using extreme programming

2. Describe Business Requirement Specification (BRS) and SRS (Software Requirement Specification) for any Project like Automatic Intelligent Plant Watering System. using any one of requirement analysis tool

3. Identify the classes , relationship between classes and draw standard UML diagrams using any one UML modeling tool (eg: ArgoUML that supports UML 1.4 and higher) for a system (eg: Conference Management System, student management system)

4. Test the above UML for all the scenarios identified using Selenium /JUnit / Apache JMeter

5. Perform COCOMO estimation for Book Management System to find effort and development time considering all necessary cost estimation factors. (Use GanttPRO Software for estimation)

COURSE OUTCOMES:

On completion of the course the student would be able to :

CO1: Design object oriented software using appropriate process models.

CO2: Differentiate software processes under waterfall and agile methodology.

CO3: Design and Develop UML diagrams for software projects.

CO4: Apply Design Patterns for a software process.

CO5: Categorize testing methods and compare different testing tools for software processes.

CO6: Analyze object oriented metrics and quality for software engineering processes.

TOTAL:45 PERIODS

REFERENCES:

1. Yogesh Singh, RuchikaMalhotra, " Object – Oriented Software Engineering", PHI Learning Private Limited ,First edition,2012
2. Ivar Jacobson. Magnus Christerson, PatrikJonsson, Gunnar Overgaard, "Object Oriented Software Engineering, A Use Case Driven Approach", Pearson Education, Seventh Impression, 2009
3. Craig Larman, "Applying UML and Patterns, an Introduction to Object-Oriented Analysis and Design and Iterative Development", Pearson Education, Third Edition, 2008.
4. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen,
5. Roger S. Pressman, "Software Engineering: A Practitioner's Approach, Tata McGraw-Hill Education, 8th Edition, 2015

MC4301 **MACHINE LEARNING** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To gain knowledge on foundations of machine learning and apply suitable dimensionality reduction techniques for an application
- To select the appropriate model and use feature engineering techniques
- To gain knowledge on Probability and Bayesian Learning to solve the given problem
- To design and implement the machine learning techniques for real world problems
- To analyze, learn and classify complex data without predefined models also

UNIT I INTRODUCTION **9**

Human Learning - Types – Machine Learning - Types - Problems not to be solved - Applications - Languages/Tools– Issues. Preparing to Model: Introduction - Machine Learning Activities - Types of data - Exploring structure of data - Data quality and remediation - Data Pre-processing

UNIT II MODEL EVALUATION AND FEATURE ENGINEERING **9**

Model Selection - Training Model - Model Representation and Interpretability - Evaluating Performance of a Model - Improving Performance of a Model - Feature Engineering: Feature Transformation - Feature Subset Selection

UNIT III BAYESIAN LEARNING **9**

Basic Probability Notation- Inference – Independence - Bayes' Rule. Bayesian Learning: Maximum Likelihood and Least Squared error hypothesis-Maximum Likelihood hypotheses for predicting probabilities- Minimum description Length principle -Bayes optimal classifier – Naive Bayes classifier - Bayesian Belief networks -EM algorithm.

UNIT VI PARAMETRIC MACHINE LEARNING **9**

Logistic Regression: Classification and representation – – Cost function – Gradient descent – Advanced optimization – Regularization - Solving the problems on overfitting. Perceptron – Neural Networks – Multi – class Classification - Backpropagation – Non-linearity with activation functions (Tanh, Sigmoid, Relu, PRelu) - Dropout as regularization

UNIT V NON-PARAMETRIC MACHINE LEARNING **9**

k- Nearest Neighbors- Decision Trees – Branching – Greedy Algorithm - Multiple Branches Continuous attributes – Pruning. Random Forests: ensemble learning. Boosting – Ad boost algorithm. Support Vector Machines – Large Margin Intuition – Loss Function - Hinge Loss SVM Kernels

SUGGESTED ACTIVITIES:

1. Explore the significant steps involved in data preprocessing in Machine Learning
2. Choose a model and train a model in machine learning.
3. Explain the application of Bayes Theorem and how it's useful to predict the future
4. Make the difference between supervised Learning and unsupervised Learning Techniques
5. Differentiate Perceptron, Neural Network, Convolutional Neural Network and Deep Learning

TOTAL:45 PERIODS

COURSE OUTCOMES:

CO1: Understand about Data Preprocessing, Dimensionality reduction

CO2: Apply proper model for the given problem and use feature engineering techniques

CO3: Make use of Probability Technique to solve the given problem.

CO4: Analyze the working model and features of Decision tree

CO5: Choose and apply appropriate algorithm to learn and classify the data

REFERENCES

1. Ethem Alpaydin, "Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series)", Third Edition, MIT Press, 2014
2. Tom M. Mitchell, "Machine Learning", India Edition, 1st Edition, McGraw-Hill Education Private Limited, 2013
3. Saikat Dutt, Subramanian Chandramouli and Amit Kumar Das, "Machine Learning", 1st Edition, Pearson Education, 2019
4. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Revised Edition, Springer, 2016.
5. Aurelien Geron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 2nd Edition, O"Reilly, 2019
6. Stephen Marsland, "Machine Learning – An Algorithmic Perspectivel, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

MC4302 **INTERNET OF THINGS** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To understand the concepts of IoT and its working models
- To know the various IoT protocols
- To understand about various IoT Physical devices and Endpoints
- To know the security and privacy issues connected with IoT
- To apply the concept of Internet of Things in a real-world scenario.

UNIT I FUNDAMENTALS OF IOT **9**

Definition and Characteristics of IoT, Sensors, Actuators, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs, IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Embedded Systems, IoT Levels and Templates, Domain Specific IoTs – Home, City, Environment, Energy, Agriculture and Industry.

UNIT II IOT PROTOCOLS **9**

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFIDProtocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security

UNIT III IOT PHYSICAL DEVICES AND ENDPOINTS **9**

Introduction to Arduino and Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, and reading input from pins.

UNIT IV INTERNET OF THINGS PRIVACY, SECURITY AND GOVERNANCE **9**

Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security

UNIT V APPLICATIONS **9**

IOT APPLICATIONS - IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

SUGGESTED ACTIVITIES:

1: Study of 5 different types of sensors and actuators available in Market

2: Study of commercial IoT available in any one domain

3: Study the recent developments in IoT Protocol 4: Implement simple Python programs for IoT

4: Study on the latest government policies on IoT security and Privacy

5: A study on how to use IoT to solve some problems in your neighbourhood.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

On completion of the course, the student will be able to:

CO1: Define the infrastructure for supporting IoT deployments

CO2: Understand the usage of IoT protocols for communication between various IoT devices

CO3: Design portable IoT using Arduino/Raspberry Pi /equivalent boards.

CO4: Understand the basic concepts of security and governance as applied to IoT

CO5: Analyze and illustrate applications of IoT in real time scenarios

REFERENCES

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. Olivier Hersent, David Boswarthick, Omar Elloumi , "The Internet of Things – Key applications and Protocols", Wiley, 2012. .
3. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals, Networking Technologies, Protocols, and Use cases for the Internet of Things", Cisco Press, First Edition,2017.
4. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), "Architecting the Internet ofThings", Springer, 2011
5. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895
6. Peter Friess,'Internet of Things – From Research and Innovation to Market Deployment', River Publishers, 2014

MC4102 **OBJECT ORIENTED SOFTWARE ENGINEERING** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To understand the phases in object-oriented software development
- To gain fundamental concepts of requirements engineering and analysis.
- To know about the different approach for object-oriented design and its methods
- To learn about how to perform object-oriented testing and how to maintain software
- To provide various quality metrics and to ensure risk management.

UNIT I SOFTWARE DEVELOPMENT AND PROCESS MODELS **9**

Introduction to Software Development – Challenges – An Engineering Perspective – Object Orientation – Software Development Process – Iterative Development Process – Process Models – Life Cycle Models – Unified Process – Iterative and Incremental – Agile Processes.

UNIT II MODELING OO SYSTEMS **9**

Object Oriented Analysis (OOA / Coad-Yourdon), Object Oriented Design (OOD/Booch), Hierarchical Object-Oriented Design (HOOD), Object Modeling Technique (OMT) – Requirement Elicitation – Use Cases – SRS Document – OOA - Identification of Classes and Relationships, Identifying State and Behavior – OOD - Interaction Diagrams – Sequence Diagram – Collaboration Diagrams - Unified Modeling Language and Tools.

UNIT III DESIGN PATTERNS **9**

Design Principles – Design Patterns – GRASP – GoF – Dynamic Object Modeling – Static Object Modeling.

UNIT IV SYSTEM TESTING **9**

Software testing: Software Verification Techniques – Object Oriented Checklist:- Functional Testing – Structural Testing – Class Testing – Mutation Testing – Levels of Testing – Static and Dynamic Testing Tools - Software Maintenance – Categories – Challenges of Software Maintenance – Maintenance of Object Oriented Software – Regression Testing

UNIT V SOFTWARE QUALITY AND METRICS **9**

Need of Object Oriented Software Estimation – Lorenz and Kidd Estimation – Use Case Points Method – Class Point Method – Object Oriented Function Point – Risk Management – Software Quality Models – Analyzing the Metric Data – Metrics for Measuring Size and Structure – Measuring Software Quality - Object Oriented Metrics

SUGGESTED ACTIVITIES:

1. Discuss the different phases in any domain like Health Monitoring System using extreme programming

2. Describe Business Requirement Specification (BRS) and SRS (Software Requirement Specification) for any Project like Automatic Intelligent Plant Watering System. using any one of requirement analysis tool

3. Identify the classes , relationship between classes and draw standard UML diagrams using any one UML modeling tool (eg: ArgoUML that supports UML 1.4 and higher) for a system (eg: Conference Management System, student management system)

4. Test the above UML for all the scenarios identified using Selenium /JUnit / Apache JMeter

5. Perform COCOMO estimation for Book Management System to find effort and development time considering all necessary cost estimation factors. (Use GanttPRO Software for estimation)

COURSE OUTCOMES:

On completion of the course the student would be able to :

CO1: Design object oriented software using appropriate process models.

CO2: Differentiate software processes under waterfall and agile methodology.

CO3: Design and Develop UML diagrams for software projects.

CO4: Apply Design Patterns for a software process.

CO5: Categorize testing methods and compare different testing tools for software processes.

CO6: Analyze object oriented metrics and quality for software engineering processes.

TOTAL:45 PERIODS

REFERENCES:

1. Yogesh Singh, RuchikaMalhotra, " Object – Oriented Software Engineering", PHI Learning Private Limited ,First edition,2012
2. Ivar Jacobson. Magnus Christerson, PatrikJonsson, Gunnar Overgaard, "Object Oriented Software Engineering, A Use Case Driven Approach", Pearson Education, Seventh Impression, 2009
3. Craig Larman, "Applying UML and Patterns, an Introduction to Object-Oriented Analysis and Design and Iterative Development", Pearson Education, Third Edition, 2008.
4. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen,
5. Roger S. Pressman, "Software Engineering: A Practitioner's Approach, Tata McGraw-Hill Education, 8th Edition, 2015

MC4103 **PYTHON PROGRAMMING** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To develop Python programs with conditionals, loops and functions.
- To use Python data structures – lists, tuples, dictionaries.
- To do input/output with files in Python
- To use modules, packages and frameworks in python
- To define a class with attributes and methods in python

UNIT I BASICS OF PYTHON **9**

Introduction to Python Programming – Python Interpreter and Interactive Mode– Variables and Identifiers – Arithmetic Operators – Values and Types – Statements. Operators – Boolean Values – Operator Precedence – Expression – Conditionals: If-Else Constructs – Loop Structures/Iterative Statements – While Loop – For Loop – Break Statement-Continue statement – Function Call and Returning Values – Parameter Passing – Local and Global Scope – Recursive Functions

UNIT II DATA TYPES IN PYTHON **9**

Lists, Tuples, Sets, Strings, Dictionary, Modules: Module Loading and Execution – Packages – Making Your Own Module – The Python Standard Libraries.

UNIT III FILE HANDLING AND EXCEPTION HANDLING **8**

Files: Introduction – File Path – Opening and Closing Files – Reading and Writing Files –File Position –Exception: Errors and Exceptions, Exception Handling, Multiple Exceptions

UNIT IV MODULES, PACKAGES AND FRAMEWORKS **10**

Modules: Introduction – Module Loading and Execution – Packages – Making Your Own Module – The Python Libraries for data processing, data mining and visualization- NUMPY, Pandas,Matplotlib, Plotly-Frameworks- -Django, Flask, Web2Py

UNIT V OBJECT ORIENTED PROGRAMMING IN PYTHON **9**

Creating a Class, Class methods, Class Inheritance, Encapsulation, Polymorphism, class method vs. static methods, Python object persistence.

SUGGESTED ACTIVITIES:

1. Display a multiplication Table Both players are given the same string, S ; Both players have to make substrings using the letters of the string S.
2. Player A has to make words starting with consonants. Player B has to make words starting with vowels. The game ends when both players have made all possible substrings. Do Scoring.
3. Write a function definition for JTOI() in Python that would display the corrected version of entire content of the file .TXT (has wrongly alphabet J in place of alphabet I) with all the alphabets "J" to be displayed as an alphabet "I" on screen.

4. Consider a CSV file of profit of 10 items in monthly sales of a year . Read this file using Pandas or NumPy or using the in-built matplotlib function. Perform the following task.

5. Read Total profit of all months and show it using a line plot Read all product sales data and show it using a multi-line plot Read each item sales data of each month and show it using a scatter plot Read each item product sales data and show it using the bar chart Read sales data of bathing soap of all months and show it using a bar chart. Calculate total sale data an year for each product and show it using a Pie chart

6. Create a Python class called Bank Account which represents a bank account, having as attributes: account Number (numeric type), name (name of the account owner as string type), balance. Create a constructor with parameters: account Number, name, balance. Create a Deposit() method which manages the deposit actions. Create a Withdrawal() method which manages withdrawals actions

COURSE OUTCOMES:

On completion of the course the student would be able to :

CO1: Develop algorithmic solutions to simple computational problems

CO2: Represent compound data using Python lists, tuples and dictionaries.

CO3: Read and write data from/to files in Python Programs

CO4: Structure simple Python programs using libraries, modules etc.

CO5: Structure a program by bundling related properties and behaviours into individual objects.

TOTAL : 45 PERIODS

REFERENCES

1. Reema Thareja, "Python Programming using Problem Solving Approach", Oxford University Press, First edition, 2017

2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", Second Edition, Shroff, O'Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/

3. Guido van Rossum, Fred L. Drake Jr., "An Introduction to Python – Revised and Updated for Python 3.2, Network Theory Ltd., First edition, 2011

4. John V Guttag, "Introduction to Computation and Programming Using Python", Revised and Expanded Edition, MIT Press, 2013

5. Charles Dierbach, "Introduction to Computer Science using Python", Wiley India Edition, First Edition, 2016

MC4202 **ADVANCED DATABASE TECHNOLOGY** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To understand the working principles and query processing of distributed databases.
- To understand the basics of spatial, temporal and mobile databases and their applications.
- To distinguish the different types of NoSQL databases.
- To understand the basics of XML and create well-formed and valid XML documents.
- To gain knowledge about information retrieval and web search.

UNIT I DISTRIBUTED DATABASES **9**

Distributed Systems – Introduction – Architecture – Distributed Database Concepts – Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing

UNIT II SPATIAL AND TEMPORAL DATABASES **9**

Active Databases Model – Design and Implementation Issues - Temporal Databases - Temporal Querying - Spatial Databases: Spatial Data Types, Spatial Operators and Queries – Spatial Indexing and Mining – Applications – Mobile Databases: Location and Handoff Management, Mobile Transaction Models – Deductive Databases - Multimedia Databases.

UNIT III NOSQL DATABASES **9**

NoSQL – CAP Theorem – Sharding - Document based – MongoDB Operation: Insert, Update, Delete, Query, Indexing, Application, Replication, Sharding–Cassandra: Data Model, Key Space, Table Operations, CRUD Operations, CQL Types – HIVE: Data types, Database Operations, Partitioning – HiveQL – OrientDB Graph database – OrientDB Features

UNIT IV XML DATABASES **9**

Structured, Semi structured, and Unstructured Data – XML Hierarchical Data Model – XML Documents – Document Type Definition – XML Schema – XML Documents and Databases – XML Querying – XPath – XQuery

UNIT V INFORMATION RETRIEVAL AND WEB SEARCH **9**

IR concepts – Retrieval Models – Queries in IR system – Text Preprocessing – Inverted Indexing – Evaluation Measures – Web Search and Analytics – Current trends.

TOTAL: 45 PERIODS

Suggested Activities:

1. Create a distributed database for any application (ex. book store) and access it using PHP and Python
2. Create spatial database of any place and perform query operations
3. Creating Databases and writing simple queries using MongoDB, DynamoDB, Voldemort KeyValue Distributed Data Store Hbase and Neo4j.

4. Creating XML Documents, Document Type Definition and XML Schema for any e-commerce website and perform XML Querying
5. Perform sentiment analysis for any web document using text preprocessing techniques.

COURSE OUTCOMES:

On completion of the course, the student will be able to:

CO1: Design a distributed database system and execute distributed queries.

CO2: Manage Spatial and Temporal Database systems and implement it in corresponding applications.

CO3: Use NoSQL database systems and manipulate the data associated with it.

CO4: Design XML database systems and validate with XML schema.

CO5: Apply knowledge of information retrieval concepts on web databases.

REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudharshan, "Database System Concepts", Seventh Edition, McGraw Hill, 2019.
2. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Seventh Edition, Pearson Education/Addison Wesley, 2017.
3. Guy Harrison, "Next Generation Databases, NoSQL, NewSQL and Big Data", First Edition, Apress publishers, 2015 27
4. Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining: Concepts and Techniques", Third Edition, Morgan Kaufmann, 2012.
5. Brad Dayley, "Teach Yourself NoSQL with MongoDB in 24 Hours", Sams Publishing, First Edition, 2014.
6. C. J. Date, A. Kannan, S. Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006

MC4302 **INTERNET OF THINGS** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To understand the concepts of IoT and its working models
- To know the various IoT protocols
- To understand about various IoT Physical devices and Endpoints
- To know the security and privacy issues connected with IoT
- To apply the concept of Internet of Things in a real-world scenario.

UNIT I FUNDAMENTALS OF IOT **9**

Definition and Characteristics of IoT, Sensors, Actuators, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs, IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Embedded Systems, IoT Levels and Templates, Domain Specific IoTs – Home, City, Environment, Energy, Agriculture and Industry.

UNIT II IOT PROTOCOLS **9**

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFIDProtocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security

UNIT III IOT PHYSICAL DEVICES AND ENDPOINTS **9**

Introduction to Arduino and Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, and reading input from pins.

UNIT IV INTERNET OF THINGS PRIVACY, SECURITY AND GOVERNANCE **9**

Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security

UNIT V APPLICATIONS **9**

IOT APPLICATIONS - IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

SUGGESTED ACTIVITIES:

1: Study of 5 different types of sensors and actuators available in Market

2: Study of commercial IoT available in any one domain

3: Study the recent developments in IoT Protocol 4: Implement simple Python programs for IoT

4: Study on the latest government policies on IoT security and Privacy

5: A study on how to use IoT to solve some problems in your neighbourhood.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

On completion of the course, the student will be able to:

CO1: Define the infrastructure for supporting IoT deployments

CO2: Understand the usage of IoT protocols for communication between various IoT devices

CO3: Design portable IoT using Arduino/Raspberry Pi /equivalent boards.

CO4: Understand the basic concepts of security and governance as applied to IoT

CO5: Analyze and illustrate applications of IoT in real time scenarios

REFERENCES

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. Olivier Hersent, David Boswarthick, Omar Elloumi , "The Internet of Things – Key applications and Protocols", Wiley, 2012. .
3. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals, Networking Technologies, Protocols, and Use cases for the Internet of Things", Cisco Press, First Edition,2017.
4. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), "Architecting the Internet ofThings", Springer, 2011
5. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895
6. Peter Friess,'Internet of Things – From Research and Innovation to Market Deployment', River Publishers, 2014

MC4103 **PYTHON PROGRAMMING** **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To develop Python programs with conditionals, loops and functions.
- To use Python data structures – lists, tuples, dictionaries.
- To do input/output with files in Python
- To use modules, packages and frameworks in python
- To define a class with attributes and methods in python

UNIT I BASICS OF PYTHON 9

Introduction to Python Programming – Python Interpreter and Interactive Mode– Variables and Identifiers – Arithmetic Operators – Values and Types – Statements. Operators – Boolean Values – Operator Precedence – Expression – Conditionals: If-Else Constructs – Loop Structures/Iterative Statements – While Loop – For Loop – Break Statement-Continue statement – Function Call and Returning Values – Parameter Passing – Local and Global Scope – Recursive Functions

UNIT II DATA TYPES IN PYTHON 9

Lists, Tuples, Sets, Strings, Dictionary, Modules: Module Loading and Execution – Packages – Making Your Own Module – The Python Standard Libraries.

UNIT III FILE HANDLING AND EXCEPTION HANDLING 8

Files: Introduction – File Path – Opening and Closing Files – Reading and Writing Files –File Position –Exception: Errors and Exceptions, Exception Handling, Multiple Exceptions

UNIT IV MODULES, PACKAGES AND FRAMEWORKS 10

Modules: Introduction – Module Loading and Execution – Packages – Making Your Own Module – The Python Libraries for data processing, data mining and visualization- NUMPY, Pandas,Matplotlib, Plotly-Frameworks- -Django, Flask, Web2Py

UNIT V OBJECT ORIENTED PROGRAMMING IN PYTHON 9

Creating a Class, Class methods, Class Inheritance, Encapsulation, Polymorphism, class method vs. static methods, Python object persistence.

SUGGESTED ACTIVITIES:

1. Display a multiplication Table Both players are given the same string, S ; Both players have to make substrings using the letters of the string S.
2. Player A has to make words starting with consonants. Player B has to make words starting with vowels. The game ends when both players have made all possible substrings. Do Scoring.
3. Write a function definition for JTOI() in Python that would display the corrected version of entire content of the file .TXT (has wrongly alphabet J in place of alphabet I) with all the alphabets "J" to be displayed as an alphabet "I" on screen.

4. Consider a CSV file of profit of 10 items in monthly sales of a year . Read this file using Pandas or NumPy or using the in-built matplotlib function. Perform the following task.

5. Read Total profit of all months and show it using a line plot Read all product sales data and show it using a multi-line plot Read each item sales data of each month and show it using a scatter plot Read each item product sales data and show it using the bar chart Read sales data of bathing soap of all months and show it using a bar chart. Calculate total sale data an year for each product and show it using a Pie chart

6. Create a Python class called Bank Account which represents a bank account, having as attributes: account Number (numeric type), name (name of the account owner as string type), balance. Create a constructor with parameters: account Number, name, balance. Create a Deposit() method which manages the deposit actions. Create a Withdrawal() method which manages withdrawals actions

COURSE OUTCOMES:

On completion of the course the student would be able to :

CO1: Develop algorithmic solutions to simple computational problems

CO2: Represent compound data using Python lists, tuples and dictionaries.

CO3: Read and write data from/to files in Python Programs

CO4: Structure simple Python programs using libraries, modules etc.

CO5: Structure a program by bundling related properties and behaviours into individual objects.

TOTAL : 45 PERIODS

REFERENCES

1. Reema Thareja, "Python Programming using Problem Solving Approach", Oxford University Press, First edition, 2017

2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", Second Edition, Shroff, O'Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/

3. Guido van Rossum, Fred L. Drake Jr., "An Introduction to Python – Revised and Updated for Python 3.2, Network Theory Ltd., First edition, 2011

4. John V Guttag, "Introduction to Computation and Programming Using Python", Revised and Expanded Edition, MIT Press, 2013

5. Charles Dierbach, "Introduction to Computer Science using Python", Wiley India Edition, First Edition, 2016

Principal
J.K.K.Munirajah College of Technology
(Autonomous)
T.N.Palayam, Gobi (Tk),
Erode (Dt) - 638 506