DEPARTMENT OF AUTOMATION & ROBOTICS

FMTH0303-3.0

Laboratory Plan

Semester: V Year: **2018 - 19**

Laboratory Title: **OOP and Python Practice**	Lab Code: **16EARP305**
Total Hours: **22**	Duration of ESA: 2 hours
ISA Marks: **80**	ESA Marks: **20**

Experiment wise Plan

List of experiments/jobs planned to meet the requirements of the course.

Category: Demonstration		**Total Weightage: 20**		**No. of lab sessions: 2**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the theory**
1	Write programs using the concept of OOP (C++/Java) Language Fundamentals and concept of command line arguments.	1	10	
	Learning Objectives: The students should be able to: 1. Demonstrate how to compile and run a program in command prompt. 2. Write programs using operators and control statements. 3. Write programs for accepting command line arguments and process them in program. 4. Demonstrate how to compile and run a Java program using different IDE's like eclipse, Net beans etc.			Object Oriented Programming -I
2	Write programs using the concept of arrays, Strings and String Buffer class and exception Handling.	1	10	
	Learning Objectives: The students should be able to: 1. Write programs using different types of arrays and strings. 2. Write a program to catch different types of exceptions.			Object Oriented Programming -I

	3. Demonstrate how the String Buffer is used in a program.			
Category: Exercise		**Total Weightage: 20**		**No. of lab sessions: 2**
3	Develop a swing based GUI using swing components and containers and connect it to database .	1	10	Object Oriented Programming -I
	Learning Objectives: The students should be able to: 1. Develop a GUI using swing components and containers. 2. Demonstrate how to insert, update and retrieve data from a database by using a simple swing based program. 3. Demonstrate the procedure of database connection.			
4	Write programs using the concept of Generic class, Inheritance, Interface and Package.	1	10	
	Learning Objectives: The students should be able to: 1. Write a program to create base class and derived class and demonstrate the inheritance concept using the same program. 2. Write a program to create interface and demonstrate how to use the interface for other programs also. 3. Use the built in packages to write programs for defined task. 4. Create the user packages and demonstrate how to use the user package in other programs or other classes. 5. Demonstrate how to create parameterized constructors and how to use different types of access specifies in a program.			Object Oriented Programming -I
Category: Exercise		**Total Weightage: 30**		**No. of lab sessions: 3**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the theory**
5	Write a program using the concepts of python scripting elements python constructs, data structures.	1	10	Python programming-II

	Learning Objectives: The students should be able to: 1. Demonstrate how to compile and run a program in command prompt. 2. Write programs using operators and control statements. 3. Write programs for accepting command line arguments and process them in program. 4. Demonstrate how to compile and run a python program using different IDE's like anaconda ,ipython etc.			
6	Write programs using the concept of functions, modules, packages and regular expressions	1	10	Python programming-II
	Learning Objectives: The students should be able to: 1. Write programs using functions and modules. 2. Write a program to use packages and regular expressions			
7	Write a python program to use the language scripting elements and constructs, data structures, and repository of standard library, to develop real world applications.	1	10	Python programming-II
	Learning Objectives: The students should be able to: 1. Write a program using scripting elements and data structures. 2. Create the user packages and demonstrate how to use the user package in other programs or other classes. 3. Write a program to create interface and demonstrate how to use the interface for other programs also			
Category: Structured Enquiry		**Total Weightage: 10**		**No. of lab sessions: 3**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the theory**
8	Solving a Maze: Program a robot to solve a maze by finding the goal position in	2	10	

	the maze starting from a starting position. You will need a data structure to keep track of positions found in the maze that are yet to be explored, starting with positions around the starting position. You will compare the maze solutions found using a Stack versus a Queue for storing unexplored positions.			
	Learning Objectives: The students should be able to: 1. Select fundamentals concepts of object oriented programming concepts/python, based on the problem scenario to implement programs.			Object Oriented Programming –I/ Python programming-II
Category: Open Ended		**Total Weightage: 20**		**No. of lab sessions: 2**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the theory**
9	Implement a project using C++/Java/python concepts, for automation and robotics applications. (FOR SEE)	2	20	
	Learning Objectives: The students should be able to: 1. Use the C++/Java/python concepts to implement the project. 2. Select the appropriate tool/software to implement the project. 3. Write a technical report using IEEE standard. 4. Present the technical report for the implemented project. 5. Demonstrate the learning experiences of working in a team.			Object Oriented Programming –I/ Python programming-II

Laboratory Plan

FMTH0303-3.0

Semester: V *Year:* **2018-2019**

Laboratory Title: **DBMS Practice**	Laboratory Code 16EARP306
Total Contact Hours: **48**	Duration of ESA: 3 Hours
Total ISA Marks: **80**	Total ESA Marks: **20**

Experiment wise Plan

List of experiments/jobs planned to meet the requirements of the course.

Category: Demonstration		**Total Weightage: 10**		**No. of lab sessions: 1**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the practice**
1	Preparing an ER diagram for given database	1	10	Basic Knowledge of data base design
	Learning Objectives: The students should be able to 1. Demonstrate how structure of a database can be expressed graphically by an ER diagram. 2. Demonstrate how to represent attributes, relationships among entity sets, link attribute to entity sets and entity sets to relationships			
Category: Exercise		**Total Weightage: 10**		**No. of lab sessions: 1**
2	Execute basic SQL queries on a given database. (DDL, DML, DCL commands)	1	10	DDL, DML, DCL commands

	Learning Objectives: The students should be able to: 1. Demonstrate how to use DDL, DML and DCL commands on a database. 2. Demonstrate how to specify different types of constraints on a table while creating a table.			
Category: Structured Enquiry		**Total Weightage: 60**		**No. of lab sessions: 10**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the theory**
3	Execute nested, correlated queries using exist, like, union, intersection and joins on a given database.	2	10	Nested queries
	Learning Objectives: The students should be able to: 1. Write SQL queries to retrieve the required data, using correlated queries, nested queries, joins, and using keywords exist, like, union and intersection. 2. Demonstrate how to join two tables using different types of joins and use keywords exist, like, union, and intersection to retrieve data.			
4	Execute SQL queries on - group by, having clauses and aggregate functions on a given database to retrieve the required data.	2	20	Nested queries using clauses-group by, having &aggregate functions.
	Learning Objectives: The students should be able to: 1. Write SQL queries using group by, having clauses and aggregate functions to retrieve the required data.			
5	Specifying views in SQL	2	10	Views of SQL
	Learning Objectives: The students should be able to 1. Write SQL queries to create & update Views			
6	Design a database for the given schema using normalization concept and execution of given	2	10	Normalization-1NF,2NF,3NF & BCNF

	queries on the database and execution of queries.			
	Learning Objectives: The students should be able to: 1. Design the database for the given schema using normalization concepts and use the given RDBMS software and implement the database.			
7	Design a database for the given specifications & implement the database and write and execute the queries for the given statements.	2	10	Basic Knowledge of data base design, DDL, DML, DCL commands
	Learning Objectives: The students should be able to: 1. Draw the ER diagram for a given specifications. 2. Design a database based on the specifications given and create tables by specifying different types of constraints on database and write SQL queries for given statements and execute them. 3. Select the proper RDBMS software to implement the database.			
Category: Open Ended		**Total Weightage: 20**		**No. of lab sessions:**
Expt./ Job No.	**Experiment / Job Details**	**No. of Lab Session(s) per batch (estimate)**	**Marks / Experiment**	**Correlation of Experiment with the theory**
8	Implement a project using Java/database management systems concepts, for automation and robotics applications. (FOR ESA)		20	
	Learning Objectives: The students should be able to: 1. Use the java /database management concepts to implement the project. 2. Select the appropriate tool/software to implement the project. 3. Write a technical report using IEEE standard. 4. Present the technical report for the implemented project. 5. Demonstrate the learning experiences of working in a			

	team.	

Course Content

Course Code: **16EARE403**	Course Title: **Machine learning and ROS**	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 40
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 40		Exam Duration : 3 hours

Content	Hours
UNIT – 1	
Chapter 1:Introduction to Robot operating system ROS concepts, creating ROS packages writing a minimal ROS publisher, compiling ROS nodes, running ROS nodes, examining running minimal publisher node, scheduling node timing, writing a minimal ROS subscriber compiling and running minimal subscriber, minimal subscriber and publisher node summary writing ROS nodes more ROS tools: catkin simple, ROSlaunch, simplifying cmakelists.txt with catkin simple automating starting multiple nodes viewing output in a ROS console recording and playing back data with ROSbag.	5 hrs
Chapter 2:Messages, Classes and Servers in ROS Defining custom messages, ROS services- service messages, ROS service nodes, manual interaction with ROS services, example ROS service client, running, example service and client, using C++ classes in ROS creating library modules in ROS, introduction to action servers and action clients- creating an action server package, defining custom action-server messages, designing an action client running the example code, introduction to parameter server.	5 hrs

Chapter 3: **Introduction to machine learning** Introduction Machine Learning ,Well posed learning problem, Types of learning, supervised learning ,unsupervised learning and reinforcement learning, Learning Associations, Designing of learning system, perspectives & issues in machine learning, Concept learning task, concept learning search, Find-S: Finding a maximally specific hypotheses, version spaces & candidate elimination algorithm, Remarks - version spaces & candidate elimination algorithm, inductive bias.	5 hrs
UNIT – 2	
Chapter 4: Computational learning theory and decision tree learning Motivation, Estimating hypotheses accuracy, Basics of sampling theory, general approach for deriving confidence intervals, comparing learning algorithm. Probably learning an approximately correct hypothesis, sample complexity for finite hypnosis spaces, sample complexity for infinite hypothesis spaces, instance based learning-K nearest neighbor learning, locally weighted regression, Representation, decision tree algorithm, hypotheses space search in decision tree algorithm inductive bias in decision tree algorithm, issues in DTL, Bayesian decision theory classification.	8 hrs
Chapter 5:Kernel methods and Graphical models Embedding's into feature spaces, the kernel trick, Multiple kernel learning, Kernel dimensionality reduction Canonical Cases for Conditional Independence, Example Graphical Models, Naive Bayes' Classifier, Hidden Markov Model, Linear Regression, d-Separation Belief Propagation, Linkage–Based clustering algorithms-means and other cost minimization clustering.	7 hrs
UNIT – 3	
Chapter 6:Reinforcement Learning The learning task,Q-learning,Nondeterministic rewards & actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.	5 hrs
Chapter 7: Artificial neural network Biological motivation, neural network representations, and appropriate problems for neural network learning, perceptron's, multilayer networks and the back propagation, algorithm, an illustrative example: face recognition	5 hrs

Course Content

Course Code: 16EARE401	Course Title: Measurement Systems	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 40 hours
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 40		Duration of ESA: 3 Hrs

Content	Hrs
Unit – I	
Chapter No. 1. Introduction to Measurement Systems Need for study of Measurement Systems, Classification of Types of Measurement Applications, Computer-Aided Machines and Processes, Functional Elements of an Instrument , Active and Passive Transducers , Analog And Digital Modes of Operation , Null and Deflection Methods , Input-Output Configuration of Instruments and Measurement Systems, Static Characteristics and Static Calibration, Dynamic Characteristics.	5 hrs
Chapter No. 2. Sensors and Signal conditioning Sensor characterization, Relations between physical quantities, Sensor Classification, Specifications, Error reduction techniques, Loading errors, Signal conditioning processes, The operational amplifier, Filtering, Wheatstone bridge, Pulse modulation.	5 hrs
Chapter No. 3. Motion Measurement Fundamental Standards, Relative Displacement: Translation and Rotational, Relative Velocity: Translation and Rotational, Relative-Acceleration Measurements, Displacement Pickups, Velocity Pickups, Acceleration Pickups, Calibration and Vibration Pickups, Jerk	5 hrs

Pickups.	
Unit – II	
Chapter No. 4. Force, Torque, and Shaft Power Measurement Standards and Calibration, Basic Methods of Force Measurement, Characteristics of Elastic Force Transducers, Torque measurement on Rotating shaft, Shaft Power Measurement (Dynamometers), Vibrating Wire Force Transducers.	5 hrs
Chapter No. 5. Pressure & Sound Measurement Standards and Calibration, Basic Methods of Pressure Measurement, Deadweight Gages and Manometers, Elastic Transducers, Vibrating-Cylinder and Other Resonant Transducers, Dynamic Testing of Pressure-Measuring Systems, High-Pressure Measurement, Low-Pressure Measurement, Sound Measurement.	5 hrs
Chapter No. 6. Flow and Temperature Measurement Local Flow Velocity, Magnitude and Direction, Gross Volume Flow Rate, Standards and Calibration of Temperature Measurement, Thermal-Expansion methods, Thermoelectric Sensors, Electrical-Resistance Sensors, Junction Semiconductor Sensors, Digital Thermometers, Radiation Methods.	5 hrs
Unit – III	
Chapter No.7. Data Acquisition Systems Data conversion devices, Signal sampling and aliasing, Sampling theorem, Quantization, Encoding, Digital to analog conversion methods, Analog to digital conversion methods, Sample & Hold circuit, Flash ADC, Successive approximation ADC, Dual slope ADC, Sigma Delta ADC, Multiplexers.	5 hrs
Chapter No. 8. Transmission and Recording of Data Cable Transmission of Analog Voltage and Current Signals, Cable Transmission of Digital Data, Fiber-Optic Data Transmission, Analog Voltmeters and Potentiometers, Electrical Instruments, Digital Voltmeters and Multimeters, Signal Generation, Electromechanical XT and XY Recorders, Fiber Optic Sensors.	5 hrs

Course Content

Course Code: 17EARC304	Course Title: Measurement Systems	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 40 hours
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 40		Duration of ESA: 3 Hrs

Content	Hrs
Unit – I	
Chapter No. 1. Introduction to Measurement Systems Need for study of Measurement Systems, Classification of Types of Measurement Applications, Computer-Aided Machines and Processes, Functional Elements of an Instrument , Active and Passive Transducers , Analog And Digital Modes of Operation , Null and Deflection Methods , Input-Output Configuration of Instruments and Measurement Systems, Static Characteristics and Static Calibration, Dynamic Characteristics.	5 hrs
Chapter No. 2. Sensors and Signal conditioning Sensor characterization, Relations between physical quantities, Sensor Classification, Specifications, Error reduction techniques, Loading errors, Signal conditioning processes,	5 hrs

The operational amplifier, Filtering, Wheatstone bridge, Pulse modulation.	
Chapter No. 3. Motion Measurement Fundamental Standards, Relative Displacement: Translation and Rotational, Relative Velocity: Translation and Rotational, Relative-Acceleration Measurements, Displacement Pickups, Velocity Pickups, Acceleration Pickups, Calibration and Vibration Pickups, Jerk Pickups.	5 hrs
Unit – II	
Chapter No. 4. Force, Torque, and Shaft Power Measurement Standards and Calibration, Basic Methods of Force Measurement, Characteristics of Elastic Force Transducers, Torque measurement on Rotating shaft, Shaft Power Measurement (Dynamometers), Vibrating Wire Force Transducers.	5 hrs
Chapter No. 5. Pressure & Sound Measurement Standards and Calibration, Basic Methods of Pressure Measurement, Deadweight Gages and Manometers, Elastic Transducers, Vibrating-Cylinder and Other Resonant Transducers, Dynamic Testing of Pressure-Measuring Systems, High-Pressure Measurement, Low-Pressure Measurement, Sound Measurement.	5 hrs
Chapter No. 6. Flow and Temperature Measurement Local Flow Velocity, Magnitude and Direction, Gross Volume Flow Rate, Standards and Calibration of Temperature Measurement, Thermal-Expansion methods, Thermoelectric Sensors, Electrical-Resistance Sensors, Junction Semiconductor Sensors, Digital Thermometers, Radiation Methods.	5 hrs
Unit – III	
Chapter No.7. Data Acquisition Systems Data conversion devices, Signal sampling and aliasing, Sampling theorem, Quantization, Encoding, Digital to analog conversion methods, Analog to digital conversion methods, Sample & Hold circuit, Flash ADC, Successive approximation ADC, Dual slope ADC, Sigma Delta ADC, Multiplexers.	5 hrs
Chapter No. 8. Transmission and Recording of Data Cable Transmission of Analog Voltage and Current Signals, Cable Transmission of Digital Data, Fiber-Optic Data Transmission, Analog Voltmeters and Potentiometers, Electrical Instruments, Digital Voltmeters and Multimeters, Signal Generation, Electromechanical XT and XY Recorders, Fiber Optic Sensors.	5 hrs

Course Content

Course Code: **17EARC305**	Course Title: **Machine learning and ROS**	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 40
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 40		Exam Duration : 3 hours

Content	Hours
UNIT – 1	
Chapter 1:Introduction to Robot operating system ROS concepts, creating ROS packages writing a minimal ROS publisher, compiling ROS nodes, running ROS nodes, examining running minimal publisher node, scheduling node timing, writing a minimal ROS subscriber compiling and running minimal subscriber, minimal subscriber and publisher node summary writing ROS nodes more ROS tools: catkin simple, ROSlaunch, simplifying cmakelists.txt with catkin simple automating starting multiple nodes viewing output in a ROS console recording and playing back data with ROSbag.	5 hrs
Chapter 2:Messages, Classes and Servers in ROS Defining custom messages, ROS services- service messages, ROS service nodes, manual interaction with ROS services, example ROS service client, running, example service and client, using C++ classes in ROS creating library modules in ROS, introduction to action servers and action clients- creating an action server package, defining custom action-server messages, designing an action client running the example code, introduction to parameter server.	5 hrs
Chapter 3: **Introduction to machine learning** Introduction Machine Learning ,Well posed learning problem, Types of learning, supervised learning ,unsupervised learning and reinforcement learning, Learning Associations, Designing of learning system, perspectives & issues in machine learning, Concept learning task, concept learning search, Find-S: Finding a maximally specific hypotheses, version spaces & candidate elimination algorithm, Remarks - version spaces & candidate elimination algorithm, inductive bias.	5 hrs
UNIT – 2	
Chapter 4: Computational learning theory and decision tree learning Motivation, Estimating hypotheses accuracy, Basics of sampling theory, general approach for deriving confidence intervals, comparing learning algorithm. Probably learning an approximately correct hypothesis, sample complexity for finite hypnosis spaces, sample complexity for infinite hypothesis spaces, instance based learning-K nearest neighbor learning, locally weighted regression, Representation, decision tree algorithm, hypotheses space search in decision tree algorithm inductive bias in decision tree algorithm, issues in DTL, Bayesian decision theory classification.	8 hrs

Chapter 5:Kernel methods and Graphical models Embedding's into feature spaces, the kernel trick, Multiple kernel learning, Kernel dimensionality reduction Canonical Cases for Conditional Independence, Example Graphical Models, Naive Bayes' Classifier, Hidden Markov Model, Linear Regression, d-Separation Belief Propagation, Linkage–Based clustering algorithms-means and other cost minimization clustering.	7 hrs
UNIT – 3	
Chapter 6:Reinforcement Learning The learning task,Q-learning,Nondeterministic rewards & actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.	5 hrs
Chapter 7: Artificial neural network Biological motivation, neural network representations, and appropriate problems for neural network learning, perceptron's, multilayer networks and the back propagation, algorithm, an illustrative example: face recognition	5 hrs

Course Content

Course Code: 17EARC301	Course Title: Object Oriented Programming and Database Management Systems	
L-T-P: 4-0-0	Credits: 3	Contact Hrs: 50
ISA Marks: 50	ESA Marks: 50	Total Marks: 50
Teaching Hrs: 50		Exam Duration: 3 hrs

Content	Hrs
Unit - I	
Chapter 1 Introduction to Software Development Lifecycle and Unified Modeling Language: Software Development Lifecycle, SDLC Models, Requirement Modeling Framework, Computer Communication Methods Unified Modeling Language (UML): UML Building Blocks, UML Diagrams - Class Diagram, Object Diagram, Component Diagram, UML Modeling Types, UML Basic Notations, UML-SysML, Using the Tools, Testing the Solution, Coding the Solution, Case Studies - Modeling the Sequence Diagram for the Plant Operation, Modeling the Control Strategy Action	6
Chapter 2 Data Modeling using the ER Model: Using High-Level Conceptual Data Models for Database Design, An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design, Relationship Types of Degree Higher than Two, ER Diagrams, Naming Conventions and Design Issues	6
Chapter 3 Introduction to Object-Oriented Programming - I: Introduction to .NET Environment, The Java Virtual Machine, Variables and Data Types, Conditional and Looping Constructs, Arrays, Fields and Methods, Constructors, Overloading Methods, Garbage Collection, Nested Classes, Simple Inheritance, Multilevel Inheritance, Overriding, Overloading, Defining Interfaces, Implementing Interfaces, Polymorphism, Abstract Classes, Access Control, Access Modifiers, Access Protection	8
Unit - II	
Chapter 4 Object-Oriented Programming - II: Final Classes, Final Variables and Methods, Finalizer Method: finalise (), Exception Handling, Fundamentals of Exception Handling, Exception Types, Constructors and Methods in Throwable Class, Java's Built-in Exceptions, Unchecked and Checked Exception, Creating Your Own Exception Sub-Classes	4
Chapter 5 Object-Oriented Programming - III: Features of Python Variables, Operators and Branching, Core elements of Programs - Bindings, Strings, Input/Output, IDEs, Control Flow and Iteration, Functions - Decomposition and Abstraction, Functions and Scope, Keyword Arguments, Specifications, Lists, Tuples, Sets, Mutation, Aliasing, Cloning, Functions as Objects, Dictionaries, Example with a Dictionary, Fibonacci and Dictionaries, Global Variables, Classes and Inheritance: Object-Oriented Programming, Class Instances, Methods Classes, Examples, Hierarchies	10
Chapter 6 Introduction to Database Management Systems: Introduction to DBMS with an example, Characteristics of Database Approach, Actors on and Behind the Scene,	6

Advantages and Disadvantages of using DBMS, Data models, Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, Database System Environment	
Unit - III	
Chapter 7 Relational Data Model and SQL: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraint Violations, SQL Data Definition and Data Types, Specifying Basic Constraints in SQL, Schema Change Statements in SQL, Insert, Delete and Update Statements in SQL, Specifying Constraints as Assertion and Trigger, Indexing Techniques, Views in SQL, Basic Queries in SQL, More Complex SQL Queries, Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form	5
Chapter 8 Object-Relational Databases and Semantic Modeling Approach: Overview of Object Database Concepts, Object-Relational Features: Object Database Extensions to SQL, The ODMG Object Model and the Object Definition Language ODL, Object Database Conceptual Design, The Object Query Language OQL, Semantic Introduction to Databases, Semantic Modeling, Semantic Binary Schemas, Schema Quality Criteria, Subschemas and User views, Transaction Processing Concepts	5

DEPARTMENT OF AUTOMATION & ROBOTICS

Laboratory Plan **FMTH0303-3.1**

Semester: V *Year: 2019-20*

Laboratory Title: Object-Oriented Programming and Database Management Systems Lab	*Lab. Code*: 17EARP301
Total Hours: 24	*Duration of Exam*: 2 hrs
Total Exam Marks: 100	*Total ISA. Marks*: 80

Experiment-wise plan

1. ***List of experiments/jobs planned to meet the requirements of the course.***

Category: Demonstration ***Total Weightage: 35*** ***No. of lab sessions: 7***

Learning Outcomes:

The students should be able to:

1. *Design and model using UML diagrams and ER models.*
2. *Demonstrate how to compile and run a program in JAVA, Python, and .NET environment.*
3. *Write programs using class, inheritance, and other fundamentals of OOP.*
4. *Write SQL statements concerning data manipulation using retrieving, inserting, updating, and deleting commands.*
5. *Write packages/procedure for manipulating data and triggers to enhance data retrieval.*

Expt./Job No.	***Experiment/job Details***	***No. of Lab. Session/s per batch (estimate)***	***Marks/ Experiment***	***Correlation of Experiment with the theory***
1	SysML - Getting used to tool, use case, creating class diagram, sequence diagram, and state diagram.	*1*	*5*	*Introduction to Software Development Lifecycle and Unified Modeling Language*
2	Creating ER models considering different relationship and attributes.	*1*	*5*	*Data Modeling using the ER Model*
3	Write programs in Java or .NET using the concept of OOP like arrays, strings, functions, overloading, and exception handling.	*1*	*5*	*Introduction to Object-Oriented Programming - I*
4	Write programs in JAVA or .NET using the concept of a generic class, inheritance,	*1*	*5*	*Object-Oriented*

	interface, and package.			*Programming - II*
5	Write programs in PYTHON using the concept of generic classes, inheritance, interface, and package.	*1*	*5*	*Object-Oriented Programming - III*
6	Write SQL statements related to data manipulation, like insert, delete, and update.	*1*	*5*	*Relational Data Model and SQL*
7	Write statements to create views, procedures, packages, and indexing for fast retrieval.	*1*	*5*	*Relational Data Model and SQL*

Category: Exercises ***Total Weightage: 20*** ***No. of lab sessions: 2***

Learning Outcomes:

The students should be able to:

1. *Design and model using UML diagrams.*
2. *Implement classes in JAVA or .NET environment.*
3. *Compile and build JAR/DLL files.*
4. *Design and mode ER models for different scenarios.*
5. *Construct a database schema with data manipulation SQL statement, a proper procedure in place, and create triggers for fast data retrieval.*

Expt./Job No.	***Experiment/job Details***	***No. of Lab. Session/s per batch (estimate)***	***Marks/Experiment***	***Correlation of Experiment with the theory***
1	*Develop a class diagram concerning sensor, actuators and controls, implement these classes, and build JAR/DLL files.*	*1*	*10*	*Introduction to Software Development Lifecycle and Unified Modeling Language* *Introduction to Object-Oriented Programming - I* *Object-Oriented Programming - II* *Object-Oriented Programming - III*
6.	*Develop an ER model and construct a database schema for a given manufacturing scenario.*	*1*	*10*	*Data Modeling using the ER Model*

				Relational Data Model and SQL *Relational Data Model and SQL*

Category: Structured Enquiry ***Total Weightage: 25*** ***No. of lab sessions: 2***

Learning Outcomes:

The students should be able to:

1. *Design, develop and implement application utilizing previously developed JAR/DLL files.*
2. *Store data from the application into the database.*
3. *Design, development and implement the user interface for visualization of data from the database.*

Expt./Job No.	***Experiment/job Details***	***No. of Lab. Session/s per batch (estimate)***	***Marks/ Experiment***	***Correlation of Experiment with the theory***
1.	*Implement a project which utilizes previously generated JAR/DLL files and database schema to store data from automation devices and control the actuators. Additionally, proper checks have to be implemented and with necessary visualization.*	*2*	*25*	

Category: Open Ended ***Total Weightage: 20*** ***No. of lab sessions: 2***

Learning Outcomes:

The students should be able to:

1. *Use the OOP concepts to implement the project.*
2. *Use database concept to implement the project*
3. *Select the appropriate tool/software to implement the project.*
4. *Write a technical report using a predefined template.*
5. *Present the technical report of the implemented project.*
6. *Demonstrate the learning experiences of working in a team.*

Expt./Job No.	***Experiment/job Details***	***No. of Lab. Slots per batch (estimate)***	***Marks/ Experiment***	***Correlation of Experiment with the theory***
1.	*Implement a project using C++/Java/python/DB concepts, for automation and robotics applications.*	*2*	*20*	

Course Content

Course Code: 17EARE301	Course Title: **Artificial intelligence for autonomous systems**	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 40
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 40		Exam Duration : 3 hours

Content	Hours
UNIT – 1	
Chapter 1: **Introduction to Artificial intelligence and autonomous systems** Foundation of artificial intelligence, robotics and the AI approach, Semi-autonomous control, Seven areas of AI**,** The Concept of Rationality The Nature of Environments, The Structure of Agents, Problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies, Informed Search Strategies, Knowledge representation in AI, knowledge based agents, propositional logic, predicate calculus, inference rules	5hrs
Chapter 2: Robotic software architectures Subsumption architecture, Three-layer architecture, Pipeline architecture, Hierarchical Paradigm- Attributes of the Hierarchical Paradigm, Reactive Paradigm-Attributes of Reactive Paradigm , Hybrid Deliberative/Reactive Paradigm-Attributes of Hybrid Paradigm, Architectural Aspects, Managerial Architectures-Autonomous Robot Architecture (AuRA), Sensor Fusion Effects (SFX), State-Hierarchy Architectures, Model-Oriented Architectures, Interleaving Deliberation and Reactive Control.	5 hrs.
Chapter 3: Biological Foundations of the Reactive Paradigm Agency and computational theory , Animal Behaviors, Reflexive behaviors , Coordination and Control of Behaviors, Innate releasing mechanisms ,Concurrent behaviors ,Perception in Behaviors , Action-perception cycle ,Two functions of perception Gibson: Ecological approach , Neisser: Two perceptual systems , Schema Theory , Behaviors and schema theory , Principles and Issues in Transferring Insights to Robots	5 hrs
UNIT – 2	
Chapter 4: Capturing intelligence - Designing a reactive implementation with common sensing techniques for robotics perception Behaviors as Objects in OOP, Steps in Designing a Reactive Behavioral System, Case Study: Unmanned Ground Robotics Competition, Assemblages of Behaviors, Logical sensors, Behavioral Sensor Fusion, Designing a Sensor Suite, Proprioceptive Sensors, Proximity Sensors, Computer Vision, Range from Vision, Case Study: Hors d'Oeuvres, Anyone?	8 hrs
Chapter 5: Multi-agents and navigation in robotics Heterogeneity, Control, Cooperation, Emergent Social Behavior, Topological Path Planning, Relational Methods, Associative Methods, Case Study of Topological Navigation with a Hybrid Architecture	7 hrs

UNIT – 3	
Chapter 6:Localization and Map Making Sonar Sensor Model,Bayesian,Conditional probabilities ,Conditional probabilities , Updating with Bayes' rule ,Dempster-Shafer Theory ,Shafer belief functions Belief function for sonar Dempster's rule of combination Weight of conflict metric,HIMM sonar model and Comparison of Methods ,Example computations ,Performance Errors due to observations from stationary robot ,Tuning ,Localization ,Continuous localization and mapping ,Feature-based localization Exploration ,Frontier-based exploration ,Generalized Voronoi graph methods .	6hrs
Chapter 7: Deep learning and natural language processing Deep Learning Improvement of the Deep Neural Network Vanishing Gradient Over fitting Computational Load. Language models, text classification, information retrieval	4 hrs

==========

Course Content

Course Code: 17EARE304	Course Title: Digital System Design and FPGA programming	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 50
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 50		Exam Duration: 3 hrs

Content	Hrs
Unit – 1	
Chapter No. 1. Review of Logic Design Fundamentals: Combinational logic, Boolean algebra and algebraic Simplification Karnaugh maps, designing with NAND and NOR gates, hazards in combinational circuits, flip-flops and latches, Mealy sequential circuit design, design of a Moore sequential circuit, equivalent states and reduction of state tables, sequential circuit timing, tristate logic and busses. Advanced Design Issues: Meta-stability, Noise Margins, Power, Fan-out, Timing Considerations, Brief overview of programmable logic devices, simple programmable logic devices (SPLDs), complex programmable logic devices (CPLDs), field-programmable gate arrays (FPGAs),	9 hrs
Chapter No. 2. Introduction to State Machine Charts and Microprogramming: State machine(SM) charts, derivation of SM charts, realization of SM charts, implementation of the dice game, microprogramming,: Design Examples	6 hrs
Unit – 2	
Chapter No. 3. Designing with Field Programmable Gate Arrays: Implementing functions in FPGAs, implementing functions using Shannon's decomposition, carry chains in FPGAs, cascade chains in FPGAs, examples of logic blocks in commercial FPGAs, dedicated memory in FPGAs, dedicated multipliers in FPGAs, cost of programmability, FPGAs and One-Hot state assignment	7 hrs
Chapter No. 4. Modeling and design with HDL Basic Concepts, Dataflow Descriptions, Behavioral Descriptions ,Structural Descriptions, Design examples, Timing and Delays, BCD to 7-Segment Display Decoder, BCD Adder, 32-Bit Adders, Traffic Light Controller, Shift-and-Add Multiplier, Array Multiplier. **Introduction to Verilog and VHDL**: Data Types, Modeling Concepts, Task and Functions, Specify Block and Timing Checks , Architecture study of popular FPGA families	8 hrs
Unit – 3	
Chapter No. 5. Testing and Verification What is Verification, what is a Test bench, The Importance of Verification, Convergence Model, What Is Being Verified, Functional Verification Approaches, Testing Versus Verification, Design and Verification Reuse, Cost of Verification	5 hrs
Chapter No. 6 Case studies on FPGA technologies in Automation and Robotics applications I. Robotic Car from Georgia Institute of Technology II. Robotic Controller: ASIC versus FPGA III. Expanding a robot's life: Low power object recognition via FPGA-based DCNN deployment IV. FPGA-powered parallel, pipelined vision algorithms	5 hrs

==========

Course Content

Course Code: 17EARE304	Course Title: Digital System Design and FPGA programming	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 50
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 50		Exam Duration: 3 hrs

Content	Hrs
Unit – 1	
Chapter No. 1. Review of Logic Design Fundamentals: Combinational logic, Boolean algebra and algebraic Simplification Karnaugh maps, designing with NAND and NOR gates, hazards in combinational circuits, flip-flops and latches, Mealy sequential circuit design, design of a Moore sequential circuit, equivalent states and reduction of state tables, sequential circuit timing, tristate logic and busses. Advanced Design Issues: Meta-stability, Noise Margins, Power, Fan-out, Timing Considerations, Brief overview of programmable logic devices, simple programmable logic devices (SPLDs), complex programmable logic devices (CPLDs), field-programmable gate arrays (FPGAs),	9 hrs
Chapter No. 2. Introduction to State Machine Charts and Microprogramming: State machine(SM) charts, derivation of SM charts, realization of SM charts, implementation of the dice game, microprogramming,: Design Examples	6 hrs
Unit – 2	
Chapter No. 3. Designing with Field Programmable Gate Arrays: Implementing functions in FPGAs, implementing functions using Shannon's decomposition, carry chains in FPGAs, cascade chains in FPGAs, examples of logic blocks in commercial FPGAs, dedicated memory in FPGAs, dedicated multipliers in FPGAs, cost of programmability, FPGAs and One-Hot state assignment	7 hrs
Chapter No. 4. Modeling and design with HDL Basic Concepts, Dataflow Descriptions, Behavioral Descriptions ,Structural Descriptions, Design examples, Timing and Delays, BCD to 7-Segment Display Decoder, BCD Adder, 32-Bit Adders, Traffic Light Controller, Shift-and-Add Multiplier, Array Multiplier. **Introduction to Verilog and VHDL**: Data Types, Modeling Concepts, Task and Functions, Specify Block and Timing Checks , Architecture study of popular FPGA families	8 hrs
Unit – 3	
Chapter No. 5. Testing and Verification What is Verification, what is a Test bench, The Importance of Verification, Convergence Model, What Is Being Verified, Functional Verification Approaches, Testing Versus Verification, Design and Verification Reuse, Cost of Verification	5 hrs
Chapter No. 6 Case studies on FPGA technologies in Automation and Robotics applications I. Robotic Car from Georgia Institute of Technology II. Robotic Controller: ASIC versus FPGA III. Expanding a robot's life: Low power object recognition via FPGA-based DCNN deployment IV. FPGA-powered parallel, pipelined vision algorithms	5 hrs

==========

Laboratory Plan

FMTH0303-3.1

Semester: VII

Year: 2020-21

Laboratory Title: Project	Lab Code: 18EARW401
Total Hours: 30	Duration of Exam: 3 Hrs
Total ESA Marks: 50	Total ISA. Marks: 50
Lab. Plan Author: Sachin Karadgi	Date: 10-Sep-2021
Checked By: Arunkumar C Giriyapur	Date: 10-Sep-2021

Prerequisites:

Subjects learnt up to VI semester.

Course Outcomes-CO

At the end of the course student will be able to:

1. Carry out market survey, do need analysis and identify suitable problems.
2. Write a project proposal, which will involve developing a complete solution for the identified problem from the real world.
3. Apply the principles of engineering design to plan and manage the project.
4. Apply suitable design processes and develop the best possible solution.
5. Develop proof of concepts and models for verification.
6. Prepare production drawings, bill of materials and process plans.

==========

Course Articulation Matrix: Mapping of Course Outcomes (CO) with Program outcomes (PO)

Laboratory (Course) Title: **Project** Laboratory (Course) code: 18EARW401 Semester: VII Year: 2020-21

Course Outcomes (CO) / Program Outcomes (PO)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1. Carry out market survey, do need analysis and identify suitable problems.	**H**	**H**												
2. Write a project proposal, which will involve developing a complete solution for the identified problem from the real world.		**H**	**H**		**M**					**H**				
3. Apply the principles of engineering design to plan and manage the project.			**H**											
4. Apply suitable design processes and develop the best possible solution.			**H**		**M**		**M**							
5. Develop proof of concepts and models for verification.			**H**											
6. Prepare production drawings, bill of materials and process plans.			**H**							**H**				

Degree of compliance **L**: Low **M**: Medium **H**: High

Competency addressed in the Course and corresponding Performance Indicators

Competency	Performance Indicators
1.3 Demonstrate competence in engineering fundamentals	1.3.1 Apply elements of mechanical engineering principles and laws to solve problems
1.3 Demonstrate competence in engineering fundamentals	1.3.2 Apply basic electrical and electronics engineering principles and laws to solve problems
1.3 Demonstrate competence in engineering fundamentals	1.3.3 Apply computer programming skills to solve problems by building algorithms ,flow charts and debugging
1.4 Demonstrate the competence in engineering knowledge appropriate to automation and robotics program	1.4.1 Apply discipline specific laws and principles to solve an interdisciplinary engineering problem
2.1 Demonstrate an ability to identify and characterize an engineering problem	2.1.1 Identifies known and unknown information, uncertainties, and biases when presented with a complex ill-structured problem
2.1 Demonstrate an ability to identify and characterize an engineering problem	2.1.3 Identifies all relevant constraints and requirements and formulate an accurate description of the problem
2.2 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.2.2 Partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design.
2.2 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.2.3 Selects appropriate analysis tools and applies those proficiently to implement the model/solution
3.2 Demonstrate an ability to generate a diverse set of alternative design solutions	3.2.1 Apply formal idea generation tools to develop multiple engineering design solutions
3.2 Demonstrate an ability to generate a diverse set of alternative design solutions	3.2.2 Build models, prototypes, etc., to develop diverse set of design solutions
3.2 Demonstrate an ability to generate a diverse set of alternative design solutions	3.2.3 Identify the suitable criteria for evaluation of alternate design solutions
5.1 Demonstrate an ability to identify/ create modern engineering tools, techniques and resources	5.1.1 Identify modern engineering tools, techniques and resources for engineering activities
7.1 Demonstrate an understanding of the impact of engineering and industrial practice on social, environmental and economic contexts	7.1 1 Identify risks/impacts in the life-cycle of an engineering product or activity
10.3 Demonstrate the ability to integrate different modes of communication	10.3.1 Create engineering-standard figures, reports and drawings to complement writing and presentations

E.g.: 1.2.3: Represents program outcome ‘1’, competency ‘2’ and performance indicator ‘3’.