Autonomous Program Structure of
Second Year B. Tech. Fourth Semester
(Electronics and Telecommunication Engineering)
Academic Year: 2021-2022 Onwards

Course Code	Course Title	Teaching Scheme Hours/ Week			Examination Scheme				Total Marks	Credit
		Lecture	Tutorial	Practical	In Sem	End Sem	Oral	Practical		
20EC401	Digital Electronics	3	1	0	50	50	0	0	100	4
20EC402	Analog and Digital Communication	3	1	0	50	50	0	0	100	4
20EC403	Machine Learning with Python	3	0	0	50	50	0	0	100	3
20EC404	Embedded Systems	3	0	0	50	50	0	0	100	3
20EC405	Object Oriented Programming	3	0	0	50	50	0	0	100	3
20EC401L	Digital Electronics Lab	0	0	2	25	0	25	0	50	1
20EC402L	Analog and Digital Communication Lab	0	0	2	25	0	25	0	50	1
20EC403L	Machine Learning with Python Lab	0	0	2	25	0	25	0	50	1
20EC404L	Embedded Systems Lab	0	0	4	25	0	0	25	50	2
20EC405L	Object Oriented Programming Lab	0	0	2	25	0	0	25	50	1
20AC401	Audit Course	0	0	2	0	0	0	0	0	No Credit
	Total	**15**	**2**	**14**	**375**	**250**	**75**	**50**	**750**	**23**
	Grand Total	**31**			**625**		**125**			

APPROVED BY
Secretary Governing Body
MKSSS's Cummins College of Engineering
Pune-411052

Cummins College of Engineering for Women, Karvenagar, Pune-411052

APPROVED BY
Chairman Governing Body
MKSSS's Cummins College of Engineering
For Women, Pune-411052

20EC401 DIGITAL ELECTRONICS

Teaching Scheme
Lectures: 3 Hours / Week
Tutorial:1 Hour/Week

Examination Scheme
In Semester: 50 Marks
End Semester: 50 Marks
Credits: 4

Prerequisite: 20ES01: Basic Electrical and Electronics Engineering

Course Objectives:

1. To solve the Sum of Products (SOP) and Product of Sum (POS) equations using K-map
2. To design Combinational logic circuits such as Adder, Multiplexer, De-multiplexer, Decoder, Comparator
3. To design sequential circuits like Counters, Shift Registers
4. Study various types of Programmable Logic Devices (PLDs)

Course Outcomes:

After completion of the course, students will be able to

CO1 Illustrate reduction of logical expressions using k-map and realize the functions using logic gates
CO2 Design combinational and sequential digital logic circuits
CO3 Design digital systems using Finite State Machines
CO4 Classify digital logic families and implement combinational logic circuits using PLD

Unit I: Fundamentals of Digital Logic (08)

Number system: Hex Number, Standard representation of logic functions, Truth table, SOP and POS forms, Canonical form, Min and Max terms, Minimization of logical functions up to 4 variables using K-map, Don't care conditions.

Unit II: Combinational Logic Design (08)

Circuit designs using Adders, Subtractors, Code converters, Digital Comparators, Multiplexers, De-multiplexers, Decoders, Encoders, Parity generator and checker, Arithmetic logic unit.

Unit III: Sequential Logic Design (08)

1 bit memory cell, Clocked SR, JK, T, D and MS-JK flip-flop, Use of preset and clear terminals, Excitation table for flip-flops, Conversion of flip-flops, Applications of flip-flops: Registers, Shift registers, Sequence Generators, Counters: Synchronous and Asynchronous counters.

Unit IV: State Machines (10)

Mealy and Moore machines representation, State diagram, State table, State assignment, Design of State Machines using State assignment and State reduction, Design of sequence detector using Finite State Machine (FSM), Applications of FSM**:** Traffic light controller, Lift controller, Vending Machines.

Unit V: Digital Logic Families and Programmable Logic Devices (08)

Classification of logic families, Characteristics of digital ICs: Speed of operation, Power Dissipation, Figure of merit, Fan in, Fan out, Current and Voltage parameters, Noise immunity, Operating temperatures and Power supply requirements, Introduction to PLDs and their types: PAL, CPLD and FPGA, Interfacing of TTL to CMOS and CMOS to TTL, Comparison between CPLD and FPGA.

Text Books:

1. R .P Jain, **"Modern digital electronics"**, *TMH Publication,* (3rd Edition), (2007).
2. Anand Kumar, **"Fundamentals of digital circuits"**, *PHI Publication,* (1st Edition), (2001).
3. Stephen D. Brown and Zvonko G Vranesic, **"Fundamentals of Digital Logic with Verilog Design"**, *Pearson Education*, (2nd Edition),(2008).

Reference Books:

1. Wakerly, **"Digital Design Principles and Practices"**, *Pearson Education*, (3rd Edition), (2004).
2. M. Morris Mano, **"Digital Logic and Computer Design"**, *Pearson Education,* (3rd Edition), (2004).

Online Resources:

1. NPTEL Course **"Digital Circuits and Systems"**
   https://nptel.ac.in/courses /117/106 /117106086/

20EC402 ANALOG AND DIGITAL COMMUNICATION

Teaching Scheme
Lectures: 3 Hours / Week
Tutorial: 1 Hour / Week

Examination Scheme
In Semester: 50 Marks
End Semester: 50 Marks
Credits: 4

Prerequisite: 20EC302 Signals and Systems

Course Objectives:

1. To introduce analog modulation and demodulation techniques
2. To study sampling process and pulse analog modulation techniques
3. To explore source coding techniques PCM, DPCM, DM, ADM
4. To explain conversion of digital data to digital signal
5. To explore binary and M-ary digital modulation techniques

Course Outcomes:

After completion of the course, students will be able to

CO1 Interpret generation and detection of Amplitude modulation and Frequency modulation

CO2 Apply sampling process and describe pulse analog modulation techniques with their generation and detection

CO3 Apply source coding techniques and evaluate Bitrate, Bandwidth and Signal-to-noise ratio

CO4 Interpret and apply data formats, Multiplexing, Synchronization and Intersymbol Interference and Matched filter for reliable baseband transmission

CO5 Analyze bandpass modulation techniques and evaluate: Bit rate, Bandwidth and Euclidean distance

Unit I: Analog Modulation (06)

Amplitude Modulation, Types of AM: DSB-SC, SSB-SC, DSB-FC, Spectrum of AM, Modulation Index, Technical AM standards, AM generation and detector, Super heterodyne radio receiver, Angle modulation, Bandwidth of FM, FM generation, FM detectors, FM- Superheterodyne radio receiver.

Unit II: Pulse Analog Modulation (05)

Sampling Process: Sampling theorem (time and frequency domain), Types of sampling, Aliasing, Aperture effect, Pulse analog modulation techniques - PAM, PPM, PWM.

Unit III: Digital Transmission of Analog Signal (08)

Block diagram of digital communication system, Pulse Code Modulation (PCM) Generation and Reconstruction, Quantization Noise, Non-uniform Quantization and Companding, Delta Modulation (DM), Adaptive Delta Modulation(ADM), Differential Pulse Code Modulation(DPCM), Adaptive Differential Pulse-Code Modulation (ADPCM).

Unit IV: Baseband Digital Transmission (08)

Digital Multiplexing: Multiplexers and hierarchies, Data formats and their spectra, Synchronization: Bit Synchronization, Scramblers, Frame Synchronization, Inter-symbol Interference, Equalization, Eye diagram.

Unit V: Bandpass Digital Techniques (09)

Binary Phase Shift Keying (BPSK), Differential Phase Shift Keying (DPSK), Quadrature Phase Shift Keying (QPSK), M-Ary PSK, Quadrature Amplitude Shift Keying (QASK), Binary Frequency Shift Keying (BFSK), M-Ary FSK, Minimum shift keying (MSK), Introduction to GMSK.

Unit VI: Optimal Reception of Digital Signal (06)

Optimum Filter, Matched Filter, Probability of Error of Matched Filter, Correlation receiver, Error probability for BASK, BPSK and BFSK.

Text Books:

1. George Kennedy, **"Electronic Communication Systems"**, *McGraw-Hill*, (5th Edition), (2013).
2. Simon Haykin, Michael Moher, "**Communication Systems**", *Wiley*, (5th Edition), (2009).
3. Donald L. Schilling, GoutamSaha, Herbert Taub, "**Principles of Communication System**", T*ata McGraw-Hill Education Pvt. Ltd*, (4th Edition), (2015).

Reference Books:

1. B.P Lathi, **"Modern Digital and Analog Communication Systems"**, Oxford University Press, (3rd Edition), (2003).
2. Bernard Sklar, **"Digital Communications Fundamentals and Applications"**, *Prentice Hall P T R*, (2nd Edition), (2009).
3. A. B. *Carlson* and P. B. Crilly, **"Communication Systems"**, *McGraw-Hill*, (5th Edition), (2002).
4. T. L. Singal, **"Analog and Digital Communication"**, *Tata McGraw-Hill*, (1st Edition), (2012).

Online Resources:

1. NPTEL Course **"Principles of Communication Systems "**
   https://nptel.ac.in/courses /108/104/108104091/
2. NPTEL Course **"Principles of Digital Communications"**
   https://nptel.ac.in/courses /108/101/10810113/

20EC403 MACHINE LEARNING WITH PYTHON

Teaching Scheme
Lectures: 3 Hours / Week

Examination Scheme
In Semester: 50 Marks
End Semester: 50 Marks
Credits: 3

Prerequisite: 20BS01 Linear Algebra and Univariate Calculus, 20BS03 Multivariate Calculus, 20BSEC301 Calculus and Probability

Course Objectives:

1. To explain the basics of Python Programming Language
2. To discuss and describe the concepts of Machine learning (ML)
3. To apply ML algorithms on real world datasets for regression and prediction
4. To apply and analyse ML algorithms on real world datasets for classification and clustering

Course Outcomes:

After completion of the course, students will be able to

CO1 Explain python fundamentals
CO2 Apply python for problem solving
CO3 Explain machine learning concepts and algorithms
CO4 Apply machine learning concepts and algorithms for prediction and classification problems
CO5 Analyze the performance of machine learning algorithms

Unit I: Python Fundamentals and Libraries (08)

Data Types, Operators, Indexing and Slicing, Strings, Lists, Arrays, Tuples, Conditional statements, Control Flow, Sets, Dictionaries, Arithmetic and Boolean Operations, Data frames, Python editors, Python libraries: Numpy, Matplotlib, Scikit learn, Pandas.

Unit II: Fundamentals of Machine Learning (08)

Basic concepts in machine learning, Parametric and non-parametric modeling, Overfitting and Underfitting, Feature selection, Dimensionality reduction techniques- PCA, LDA; Training, Testing and Validation errors, Confusion matrix and Evaluation Parameters.

Unit III: Regression (09)

Introduction to Regression, Simple linear regression, Multiple linear regression, Non-Linear Regression, Evaluation metrics in regression models.

Unit IV: Classification (09)

Introduction to Classification, K-Nearest Neighbours, Decision Trees and Random Forest Algorithm, Logistic Regression, Logistic regression vs Linear regression, Support Vector Machine, Introduction to Bayesian probability, Naive Bayes algorithm, Evaluation metrics in classification.

Unit V: Clustering (08)

Introduction to Clustering, K-Means Clustering, Hierarchical Clustering and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) Clustering.

Text Books:

1. Vinod Chandra S. S., Anand Hareendran S., "**Artificial Intelligence and Machine learning**", *PHI*, (1st Edition) (2014).
2. U. Dinesh Kumar and Manaranjan Pradhan, "**Machine Learning using Python**", *John Wiley & Sons* (1st Edition), (2020).
3. Mark Lutz, "**Programming Python",** *O'Reilly Media, Inc.,* (4th Edition), (2010).
4. Ethem Alpaydın "**Introduction to Machine Learning**", *The MIT Press,* (2nd Edition), (2010).
5. Christopher Bishop, "**Pattern Recognition and Machine Learning**", *Springer,* (1st Edition), (2007).

Reference Books:

1. Chris Albon, **"Machine Learning with Python Cookbook",** *O'Reilly Media*, Inc., (1st Edition), (2018).
2. Aurélien Géron, **"Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor Flow",** *O' Reilly Media, Inc.,* (2nd Edition), (2019).
3. Kevin Murphy, **"Machine Learning: A Probabilistic Perspective",** *MIT Press,* (1st Edition), (2012).

Online Resources:

1. NPTEL Course on **"Machine Learning"** https://nptel.ac.in/courses/106/106/106106202/
2. NPTEL Course on **"Introduction to Machine Learning"** https://onlinecourses.nptel.ac.in/noc21_cs85/preview
3. NPTEL Course on **"Introduction to Machine Learning"** https://onlinecourses.nptel.ac.in/noc21_cs70/preview

20EC404 EMBEDDED SYSTEMS

Teaching Scheme
Lectures: 3 Hours / Week

Examination Scheme
In Semester: 50 Marks
End Semester: 50 Marks
Credits: 3

Prerequisite: 20ES01: Basic Electrical and Electronics Engineering, 20ES02 Fundamentals of Programming Language I

Course Objectives:

1. To explain the applications of microprocessors and microcontrollers.
2. To introduce the architecture and features of typical microcontrollers.
3. To interface real world I/O devices
4. To explain hardware and software development tools

Course Outcomes:

After completion of the course, students will be able to

CO1 Describe architecture of microcontrollers
CO2 Develop program for on chip peripheral
CO3 Develop program for externally interfaced peripheral device
CO4 Design microcontroller-based system using sensors

Unit I: Introduction to Embedded System (08)

Embedded system: Need of Embedded System - Applications, Classification and working of Embedded System (block diagram), Embedded system characteristics, Microprocessors and Microcontrollers, CISC and RISC Processors, Harvard and Von Neumann Architectures, Architecture of a Microcontroller, Microcontroller Family, Microcontroller selection process for application, Microcontroller Application Development tools: Simulator, Emulator, ISP, Cross assembler.

Unit II: Microcontroller Architecture (06)

Microcontroller architecture, Pin configuration, RESET, Crystal interface, Program Status Word (PSW), Internal memory organisation, Port Structure: GPIO (LED interface), Stack and Stack Pointer, Serial communication: Concept of RS 232C.

Unit III: On-chip Peripherals (07)

Microcontroller On-chip peripherals: Counters/Timers, ADC and DAC, Introduction to sensors and transducers - LM35 sensor interface, Calibration, Interrupts.

Unit IV: Real World Interfacing – External Peripheral Interface (06)

Interfacing LCD, switch, Stepper motor, Relays, buzzer and DC motor control using PWM.

Unit V: Sensor Interface (06)

Interface sensors: Accelerometer, Gas sensor, Temperature and Humidity Sensor: DHT-11, Float sensor, Gyro sensor.

Unit VI: Design a Minimum System Using Microcontroller (09)

Case Study: Temperature monitoring and controlling, Home automation system.

Text Books:

1. Mohammad Mazidi, Janice Mazidi and Rolin McKinlay, **"The Microcontroller and Embedded Systems using Assembly and C"**, Pearson Education, (2nd Edition), (2014).
2. C. Ravichandran, M. Arulalan, **"Microcontroller-based system design"**, *Suchitra Publication*, (1st Edition), (2017).

Reference Books:

1. Myke Predko, **"Programming and customizing the microcontroller"**, *Tata McGraw Hill*, (2nd Edition), (2014).
2. Kenneth Ayala ``**The MICROCONTROLLER-Architecture, Programming and Applications",** *West Publishing Company*, (3rd Edition), (2014).

Online Resources:

1. www.intel.com
2. www.microchip.com

20EC405 OBJECT ORIENTED PROGRAMMING

Teaching scheme
Lecture: 3 hours/week

Examination Scheme
In Semester: 50 Marks
End Semester: 50 Marks
Credits: 3

Prerequisite: 20ES05 Fundamentals of Programming Language II

Course Objectives:
1. Recall the concepts and techniques of the OOP paradigm
2. Understand and utilize commonly used classes and object as superclass
3. Develop ability to understand and use interfaces, enums and Java collection framework
4. Understand the importance of streams, APIs and the concept and usage of multithreading

Course Outcomes:
After completion of the course, students will be able to
CO1 Utilize common Java constructs such as Interfaces, Enums, Lambdas, Streams and built-in classes such as String, Arrays to develop programs
CO2 Analyze the application requirement and choose appropriate collection from Java Collections Framework for storing data
CO3 Apply the concepts of multithreading
CO4 Analyze the design requirement and identify the appropriate design pattern to be applied

Unit I: Classes and Objects (07)
Basics of stack, heap, memory allocation (objects/primitives and instance/local variables). Nested classes, Inner classes, Method local classes, Anonymous classes, Object as superclass: Object class methods, importance and implementation of to String(), equals(), hashCode() methods, Immutability of objects, Wrapper classes like Byte, Double, Float, Integer, Long, Short, Autoboxing and unboxing. Commonly used classes: String, StringBuilder, Objects, Arrays, Math.

Unit II: Interfaces, Enums and Annotations (07)
Interfaces: Defining an Interface, Implementing an Interface, Using an Interface as a Type, Common interfaces: Comparable, Comparator, Iterable, Iterator, Runnable, Enums: Declaring Enum, adding fields to Enum, Enum class in Java, Built in methods, Abstract method implementations in Enum, Annotations: Basics, Declaring annotation type, Predefined annotations, Annotation processing overview.

Unit III: Generics and Collections (08)
Generics: Introduction to generics, Generic types, Generic methods, Bounded type parameters, generics and inheritance, Wildcards, Java Collection Framework (JCF): Introduction to JCF, Structure of JCF (program to interface, use of generics, Collection interface), Commonly used collections with implementations: Features, Methods offered and complexity analysis for: List (ArrayList, LinkedList), Set (HashSet, LinkedHashSet, TreeSet), Map (HashMap, LinkedHashMap, TreeMap), Queue (ArrayBlockingDeque, Priority Queue), Exploration of Javadocs to understand interface and implementations.

Unit IV: Java 8 Useful Features (07)

Java 8 interfaces: Default and Static Methods, Lambdas, Method references, Streams: Introduction, Examples, Intermediate operations: filter, Map, Flat Map, Distinct, Limit, sorted, Terminal operations: for Each, to Array, Collect and reduce, Date Time APIs: Problems with existing Date and Calendar, Local Date, Local Time, Local Date Time, Zoned Date Time, Period and Duration.

Unit V: Multithreading (07)

Multithreading vs multiprocessing, Thread, Runnable, memory model (Thread stack, frames, method stacking, stack overflow), Need for synchronization: race conditions, Happens before guarantee, Examples, Use of atomic classes, Deadlock, Starvation, Use of volatile, Publisher/Subscriber model: wait, notify/notify all.

Unit VI: Design Patterns (06)

Purpose of using design patterns, types of design patterns: Creational, Structural, Behavioral, Common design patterns such as: singleton, Factory, Builder, Proxy, Façade, Observer, Model View Controller (MVC).

Text Books:

1 Herbert Schilt, **"JAVA Complete Reference"**, *Tata McGraw Hill*, (9th Edition), (2014).
2. Eckel B., **"Thinking in Java"**, *Pearson Education*, (3rd Edition), (2017).

Reference Books:

1. Joshua Bloch,**"Effective Java"**, Addison-Wesley Professional, (3rd Edition), (2017).

Online Resources:

1. NPTEL Course **"Programming in Java"**
   https://nptel.ac.in/courses/106/105/106105191/

20EC401L DIGITAL ELECTRONICS LAB

Teaching Scheme
Practical: 2 Hours / Week

Examination Scheme
In Semester: 25 Marks
Oral: 25 Marks
Credits: 1

Course Objectives:
1. To solve the SOP and POS equations using K-map
2. To design different Combinational logic circuits
3. To design different Sequential circuits
4. To study various types of PLDs

Course Outcomes:
After completion of the course, students will be able to
CO1 Design and Implement digital circuits
CO2 Analyze combinational and sequential circuits
CO3 Implement Combinational Logic Circuits using PLDs
CO4 Utilize software tools for the simulation of digital circuits

List of Experiments:
1. Design and implement Full Adder and Full Subtractor circuit using Decoder IC.
2. a) Design and implement 8:1 Multiplexer.
   b) Design and analysis of logic functions using Multiplexer.
3. Design and implement 1- digit BCD adder using IC 7483.
4. a) Design and implement an 8-bit Magnitude comparator using IC 7485.
   b) Design and analyze the 5 bit Magnitude comparator using single IC 7485.
5. a) Design Asynchronous MOD-N counter using IC 7490.
6. a) Design and implement Synchronous Up/ Down counter.
   b) Design and analyze Synchronous MOD-NN up counter.
7. Implement Boolean expression using PLD.
8. Design and simulate Finite State Machines.

20EC402L ANALOG AND DIGITAL COMMUNICATION LAB

Teaching Scheme
Practical: 2 Hours / Week

Examination Scheme
In Semester: 25 Marks
Oral :25 Marks
Credits: 1

Course Objective

1. To demonstrate generation and detection of AM and FM signals
2. To comprehend PCM, DM, ADM waveform coding techniques
3. To observe data format along with spectral analysis
4. To explore binary and M-ary shift keying techniques

Course Outcome

After completion of the course, students will be able to

CO1 Measure and calculate modulation index, spectrum of AM and FM signal
CO2 Measure and compare bit-rate, signal-to-noise ratio, quantization error of waveform coding techniques
CO3 Plot and analyse spectrum of data formats
CO4 Measure and compare bandwidth, and bit-rate of digital modulation techniques
CO5 Interpret communication standards for analog and digital techniques with technical presentation

List of Experiments:

1. Measure modulation index and observe waveforms of AM and FM (generation and detection)
2. Simulate Super heterodyne radio receiver for FM on suitable software
3. For the given kit measure Bit-rate, Signal to noise ratio and Quantization error for PCM
4. Measure and plot slope overload and Granular noise in Delta modulation and ADM
5. Measure spectrum of BFSK signal
6. Measure and compare bit rate and bandwidth of BPSK and QPSK signals
7. Interpret spectral analysis of line codes (NRZ, RZ, Polar RZ, Bipolar (AMI), Manchester) generated on the given kit
8. Simulate and compare error probability for Binary and M-ary Shifting keying
9. Seminar presentation on Communication Standards

20EC403L MACHINE LEARNING WITH PYTHON LAB

Teaching Scheme
Practical: 2 Hours / Week

Examination Scheme
In Semester: 25 Marks
Oral:25 Marks
Credits: 1

Course Objectives:

1. To explain the basics of Python Programming Language
2. To apply the concepts of Machine Learning (ML) for data visualization and analysis
3. To apply ML algorithms for regression and prediction problems
4. To apply ML algorithms for classification and clustering

Course Outcomes:

After completion of the course, students will be able to

CO1 Apply Python programming to read and visualize datasets
CO2 Develop an algorithm and write program for solving regression from real world applications
CO3 Develop an algorithm and write program for classification of data from real world applications
CO4 Develop an algorithm and write program using clustering algorithms

List of Experiments:

1. A) Introduction to Python and Python libraries.
   B) Download/Access and read datasets (Kaggle) in Python.
2. Write a program in Python for data visualization and calculate statistical summary.
3. Develop an algorithm and solve real world regression/prediction problems using ML techniques such as linear regression.
4. Develop an algorithm and solve real world regression/prediction problems using ML techniques such as logistic regression.
5. Develop an algorithm and solve real world problems using Naive bayes Algorithm / SVM
6. Develop an algorithm and solve real world problems using Decision tree / Random Forest Algorithm.
7. Develop an algorithm and solve real world problems using K-means clustering algorithm.
8. Develop an algorithm and solve real world problems using Hierarchical clustering algorithm.

20EC404L EMBEDDED SYSTEM LAB

Teaching Scheme
Practical: 4 Hours / Week

Examination Scheme
In Semester: 25 Marks
Practical :25Marks
Credits: 2

Course Objective

1. To develop hardware interfacing skill
2. To develop software skill in embedded domain
3. To develop skill of designing embedded system using sensors
4. To explore students to development tools required for embedded system

Course Outcome

After completion of the course, students will be able to

CO1 Implement given problems using development tools required for embedded system
CO2 Develop program for on chip peripheral
CO3 Develop program for externally interfaced peripheral device
CO4 Design microcontroller-based system using sensors

List of Experiments:

1. Introduction to Studio 7 and programming Arduino board, Arduino CC (IDE)
2. Introduction to Simulator TinkerCad (online).
3. Interfacing LEDs with different patterns (GPIO)
4. Interfacing switch, LED, relay and buzzer (GPIO)
5. Interfacing Stepper motor (GPIO)
6. Develop the program to generate different waveforms using DAC.
7. Write program to transmit and receive data serially
8. Interfacing LCD Display.
9. Interface ADC and display the data on LCD as well as on serial port.
10. Interface LM 35 sensor to internal ADC and display the value on LCD.
11. Toggle GPIO port with fixed time interval using On-chip timer (without interrupt).
12. Toggle GPIO port with fixed time interval using On-chip timer (with interrupt).
13. DC Motor Speed Control using PWM.
14. Interface following sensors to ADC of microcontroller.
    1. Accelerometer
    2. Gas sensor
    3. Temperature and Humidity sensor DHT-11
15. OPEN ENDED PROBLEM : Interface
    1. Float sensor
    2. Gyro sensor
    3. IR sensor for counting movement or open and close the door using DC motor

20EC405L OBJECT ORIENTED PROGRAMMING LAB

Teaching Scheme
Practical: 2 Hours / Week

Examination Scheme
In Semester: 25 Marks
Practical: 25 Marks
Credits: 1

Course Objectives

1. To recall the concepts and techniques of the OOP paradigm
2. To utilize the commonly used classes and object as super class
3. To implement interfaces, enums and the Java collection framework
4. To understand the importance of streams, APIs and the concept and usage of multithreading

Course Outcomes

After completion of the course, students will be able to

CO1 Develop Java programs using common Java constructs
CO2 Develop Java programs by applying appropriate design pattern
CO3 Analyse the application requirement and write programs using appropriate collection from Java Collections Framework
CO4 Apply the concept of multithreading to solve given problems

List of Experiments:

1. Write a program to explore Object class methods (toString, equals, hashcode)
   a) Create an Employee class with fields first name and last name.
   b) Create and print Employee class objects.
   c) Create multiple Employee class objects and compare whether they are equal or not.
2. Declare an interface and create an anonymous class that implements it.
3. Write a program to create a Person class with fields; first name and last name and getter setters. Modify it to make it immutable. In the Person class, add a field of type Address, which has fields as city and pin code with getters and setters. Prove that the Person class is now mutable. Modify the classes such that Person becomes immutable again.
4. Write a program to concatenate two strings e.g. "hello" and "world" and check if the result of concatenation equals "helloworld" (using equals() and ==). Trim the strings to get the two strings back.
5. Write a program to declare an Enum representing days of the week. Add fields to indicate the day name and number. Print the days using values().
6. Write a program to implement basic arithmetic calculator using Enum. Calculator should support operations: addition, subtraction, multiplication and division.
7. Create a set of movies (Movie class having name and imdb rating). Print the list of movies in the ascending order of their names. Now print the movies in descending order of imdb rating, using comparator interface.
8. Create a list of names of your friends. Check if a name of a friend is present in the list. Remove a name from the list and add another. Print the list to see where the added name is appearing in the list.

9. Dissecting equals and hashcode contract - Create 4 objects from a student class (which has name and roll number as fields). All objects should have the same name and roll numbers. Add the objects to a set when – (i) the class does not have equals and hashcode overridden (ii) when the class has these methods overridden. Inspect what happens in these two cases and justify the contract.
10. Create a map of device ids and subscribers associated with it. Create a list of all device ids associated with a subscriber.
11. Given a stream of amplitude values of a sampled signal (as a List), filter out the samples having amplitudes above and below the threshold values (upper threshold: 4.5V and lower threshold: 0.5V). These filtered samples are to be passed through the multiplier system (multiplication factor = 2). Create a list of signal values at the output of the multiplier system. Find min and max amplitudes in the result.
    Example:
    Input: {0.45, 1.0, 2.2, 3.5, 4.7, 5.0, 0.21, 1.2}
    Output: {2.0, 4.4, 7.0, 2.4}
    Min: 2.0, Max: 7.0
12. Write a program to demonstrate deadlock using two threads and two locks. Hint: Threads need to acquire locks in opposite order.
13. Write a Maven project to generate a QR code. Hint: Use QRGen (https://github.com/kenglxn/QRGen) as dependency.