Centurion University of Technology and Management Odisha

Choice Based Credit System

Course Structure & Syllabus

Electronics and Communication Engineering

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT

School of Engineering & Technology

2022

Basket IV: Core Courses: Electronics and Communication Engineering

Corse Code	Course Title	Credits	T-P-PJ
CUTM1028	OOPs with C++ Programming	4	1-2-1
CUTM1029	Data Structures using C++	4	1-2-1
CUTM1030	Advanced Web Programming	4	1-2-1
CUTM1031	Java Technologies	4	2-1-1
CUTM1039	Embedded System Programming with ARM-Cortex	6	3-2-1
CUTM1040	VLSI Design (NSQF level 5)	6	3-2-1
CUTM1046	Electronic Devices and Systems	3	2-1-0
CUTM1041	Electromagnetic Interference and Compatibility	3	2-0-1
CUTM1042	Electromagnetic Field Theory & Transmission Lines	3	2-1-0
CUTM1043	Network Analysis	3	2-1-0
CUTM1045	Analog Communication Systems	3	2-1-0
CUTM1044	Digital Communication Systems	3	2-1-0
CUTM1047	Principles of wireless communication	3	2-1-0
CUTM1048	Digital Signal & Image Processing	3	2-1-0
CUTM104	Antenna Design and Analysis	3	2-1-0
CUTM1050	Wireless communication for IIOT	3	2-0-1
	Total Credits	**58**	

Basket IV: Core Courses Syllabus

OOPs with C++ Programming

Code	Course Title	Credit	T-P-PJ
CUTM1028	OOPs with C++ Programming	4	1-2-1

Objective

- To understand how C++ improves C with object-oriented features
- To learn how to design C++ classes for code reuse
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism
- To learn how to use exception handling in C++ programs

Course outcome

- Apply the object-oriented programming approach in connection with C++
- Illustrate the process of data file manipulations using C++
- Apply virtual and pure virtual function & complex programming situations
- Write an error free program of minimum 200 lines of code

Course content

Module I: Revision of C programming (8 hrs)

Revision of C Programming, Pointers, Functions (Call by value and reference), Recursion, Arrays using Pointers, Structures, Union, Enumeration and Typedef, File handling.

Programs:

1. Write a Program to perform Parameter passing.
2. Write a program to create a scientific calculator.
3. Write a program to convert a decimal to binary number using recursion.
4. Write a program to Read 'n' employee details and display the top 10 employees as per the salary.
5. Write a program to evaluate MCQ questions of an examination and generate the results using files.

Module II: Basics of Object oriented concepts (8 hrs)

Object oriented concepts Classes and Objects, Encapsulation, Abstraction, Overloading, Inheritance, Polymorphism.

Beginning with C++, Tokens, Static Members, Constant Members, Expressions, Control Structure, Functions: parameter passing, inline function, function overloading.

Programs:

1.Write a program to read a number and check whether the number is Prime number , Palindrome number , Magic number , Armstrong number , Strong number or not.

2. Write definitions for two versions of an overloaded function. This function's 1st version sum() takes an argument, int array, and returns the sum of all the elements of the passed array. The 2nd version of sum() takes two arguments, an int array and a character ('E' or 'O'). If the passed character is 'E', it returns the sum of even elements of the passed array and is the passed character is 'O', it returns the sum of odd elements. In case of any other character, it returns 0 (zero).

Module III: Class-Object-Constructor (10 hrs)

Classes: data members, member function, array of objects, static data members, constant members function, and friend function.

Constructors, Encapsulating into an object, Destructors.

Programs:

1. Define a class to represent a book in a library. Include the following members:

Data Members

Book Number, Book Name, Author, Publisher, Price, No. of copies issued, No. of copies

Member Functions

(i) To assign initial values

(ii) To issue a book after checking for its availability

(iii) To return a book

(iv) To display book information.

2. A bank maintains two kinds of accounts for customers, one called as savings and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

Define a class to represent a bank account. Include the following members: Data members: 1. Name of the depositor. 2. Account number. 3. Type of account. 4. Balance amount in the

account. Member functions: 1. To assign initial values. 2. To deposit an amount. 3. To withdraw an amount after checking the balance. 4. To display the name and balance. Write a main program to test the program

3. Declare a class to represent fixed-deposit account of 10 customers with the following data members:
Name of the depositor, Account Number, Time Period (1 or 3 or 5 years), Amount.
The class also contains following member functions:
(a) To initialize data members.
(b) For withdrawal of money (after alf of the time period has passed).
(c) To display the data members.
4. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

Module IV: Inheritance **(8 hrs)**

Associations, Inner Classes, Memory Management and pointers
Inheritance: Derived classes, member accessibility, forms of inheritance, virtual base classes.

Programs:

1. Write a Program to describe about all types of inheritance.
2. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

3. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure. The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required.

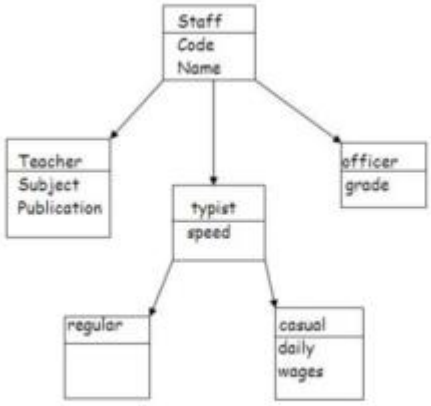

fig: class relationships (for exercise 8.3)

Module V: Polymorphism (8 hrs)

Polymorphism (Compile time Polymorphism, Run time Polymorphism), Virtual Functions, Abstract class, virtual destructors, Interfaces.

Programs:

1. Write a Program to overload ++ operator.
2. Write a program to overload + operator by concatenating strings.
3. Write a program to describe about virtual function.

Module VI: Exception Handling (8 hrs)

Exception Handling, Managing Console I/O Operations, Streams & Files: streams, hierarchy of stream classes, working with files

Programs:

1. Write a Program to describe about exception handling mechanism.
2. Write a Program to describe multi catch statement.

3. Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below.

Name	Code	Cost
Turbo C++	1001	250.95
C primer	905	95.70
.....	.....	.....
.....	.....	.....

Note that the name and code are left-justified and the cost is right justified with a precision of two digits. Trailing zeros are shown.

4. Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.

5. Write a program that reads character from the keyboard one by one. All lower case characters get store inside the file LOWER, all upper case characters get stored inside the file UPPER and all other characters get stored inside OTHERS.

Module VII: Templates **(8 hrs)**

Advance Topics in C++ Object Design and Templates STL (Standard Type Libraries)RTTI (Run Time Type Identification) Advanced Typecasting ,new data types, new operators, class implementation, namespace scope , operator keywords, new headers , C++ Containers

Programs:

1. Write a function template for finding the minimum value contained in an array.

2. Imagine a publishing company that markets both books and audio-cassette versions of its works. Create a class called Publication that stores the title (a string) and price of a publication. From this class derive two classes: Book, which adds a page count (type int); and Tape, which adds a playing time in minutes (type float). Each of the three class should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data. Write a main() program that creates an array of pointers to Publication. In a loop, ask the user for data about a particular book or Tape, and use new to create a object of type Book or Tape to hold the data. Put the pointer to the object in the data for all books and tapes, display the resulting data for all the books and taps entered, using a for loop and a single statement such as pubarr[i]->putdata();to display the data from each object in the array.

Text Books:

1. E Balagurusamy, "Object Oriented Programming with C++", Tata McGraw Hill, Sixth Edition.
2. Herbert Schlitz, "The Compete Reference C++", Tata McGraw Hill, Fourth Edition.

Reference Books:

1. Ashok Kamthane, "Object Oriented Programming with ANSI and Turbo C++", Pearson.
2. Behrouz A. Forouzan & Richard F. Gilberg "A Structured approach using C++" Cengage Learning Indian Edition.

Data Structures using C++

Code	Course Title	Credit	T-P-PJ
CUTM1029	Data Structures using C++	3	1-2-1

Objective

- Be familiar with techniques of algorithm analysis and Recursive method
- Be familiar with implementation of linked data structures such as linked lists and binary trees
- Be familiar with several sub-quadratic sorting algorithms including quick sort, merge sort and heap sort
- Be familiar with some graph algorithms such as shortest path and minimum spanning tree

Course outcome

- Evaluate algorithms and data structures in terms of time and memory complexity of basic operations
- Define basic static and dynamic data structures and relevant standard algorithms for them: stack, queue, dynamically linked lists, trees, graphs, heap, priority queue, hash tables, sorting algorithms, min-max algorithm
- Determine and demonstrate bugs in program, recognize needed basic operations with data structures
- Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures

Course content

Module I: Problem Solving Analysis **(6 hrs)**

Define the problem, Identify the problem, Introduction to Problem Solving, Problem solving basics, Defining creativity v/s innovation

Find Creative Solutions using creativity tools

Effective problem solving approaches , Critical thinking and information analysis , Brainstorming, Reverse Brainstorming, Imagineering, Mind Mapping, Six Thinking Hats: A Tool to Strengthen Critical Thinking, Collaboration, Communication, and Creativity Skills , Analyzing the situation, Gathering information, Identifying solution criteria , Decision Making Methods , Charts and Diagrams , Applying outcome-based thinking

Evaluate and Select solution

Pro's and Con's, Force field analysis, Feasibility/Capability Analysis, Decision analysis, evaluating problems , Choosing among alternatives , Qualitative analysis, discussing qualitative analysis techniques , Establishing objectives , Assigning weight to objectives in order to make the best decision , Creating a satisfaction scale to choose between alternatives

Implementing Decisions

Create an action plan, Break solution into action steps, Prioritize actions and assign roles (setting priorities for taking action) ,Follow-up at milestones

Programs:

1. Problem solving (Control structures, Arrays) using Raptor Tool.

Module II: Array & Stack **(9 hrs)**

Analysis of different Algorithms, Asymptotic analysis, Algorithm analysis, Complexity Analysis, Application of Data structures

Basic Data Structures, Arrays, Stacks and its applications (Recursion, Infix to Postfix Conversion and Postfix Evolution

Programs:

1. Write a program to perform the following menu driven program on the input array.

a. Insertion

b. Deletion

c. Searching

d. Sorting

e. Merging

f. Display

g. Exit

2. Write a program to perform the following menu driven program on the STACK.

a. Push

b. Pop

c. Display

d. Exit

Module III: Queue & Linked List **(9 hrs)**

Queues, Priority Queues, Dequees.

Linked lists: Single Linked List and Operations on Single Linked List (Creation Insertion , Deletion , Sorting and Reverse).

Programs:

1. Write a program to perform the following menu driven program on the Queue.

a. Insertion

b. Deletion

c. Display

d. Exit

2. Write a program to create a single linked list performs the following menu driven program.

a. Insertion at front

b. Insertion at end

c. Insertion at particular position

d. Deletion at front

e. Deletion at end

f. Deletion at particular position

g. Display

Module IV: Stack & Queue Using Linked List **(8 hrs)**

Circular linked list and Double linked list, Stack implementation using Linked List and Queue implementation using Linked List

Programs:

1. Write a program to create a Double linked list performs the following menu driven program.

a. Insertion at front

b. Insertion at end

c. Insertion at particular position

d. Deletion at front

e. Deletion at end

f. Deletion at particular position

g. Display

2. Write a program to create a circular linked list and display it.

3. Write a program to implement Stack Using Linked List.

4. Write a program to implement Queue Using Linked List.

Module V: Trees **(10 hrs)**

Trees and hierarchical orders ,Introduction to trees , Abstract trees , Tree traversals , Forests , Ordered trees , Binary trees , Perfect binary trees , Complete binary trees , Search trees , Binary search trees , AVL trees

Programs:

1. Write a program to create Binary tree and display it.

2. Write a program to create a BST and display it.

3. Write a program to print all pairs from two BSTs whose sum is greater than the given value.

4. Write a program to remove duplicate entries from the BST.

5. Write a program to create a AVL tree and display it.

Module VI: Searching & Sorting **(8 hrs)**

Searching & Sorting algorithms , Objectives of Searching , The Sequential Search , Analysis of Sequential Search , The Binary Search , Analysis of Binary Search , Introduction to sorting , Insertion sort , Bubble sort , Heap sort ,Merge sort ,Quick sort

Programs:

1. Write a program to perform linear and binary search.

2. Write a program to perform selection sort, Bubble sort and Insertion sort.

3. Write a program to perform merge and quick sort.

4. Write a program to perform Heap sort.

Module VII: Hashing **(8 hrs)**

Hash functions and hash tables ,Hashing & Introduction to hash tables ,Hash functions , Mapping down to 0 ... M – 1 , Chained hash tables , Scatter tables , Open addressing , Linear probing , Quadratic probing , Double hashing, Poisson distribution , Collision Resolution
Graph Terminology and Traversals.

Programs:

1. Write a program to perform Linear Probing.
2. Write a program to perform Double Hashing

Text Books:

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahani, 2nd Edition.
2. Data Structures and Algorithms in C++, Michael T.Goodrich, R, Tamassia and D.Mount, wiley Student Edition, 7th edition, John Wiley and Sons.

Reference Books:

1. Data Structures and Algorithms Analysis in C++ by Mark Allen Weiss.
2. Data Structures and Algorithms in C++, 3rd edition, Adam Drozdek, Cengage Learning.

Source of reference;http://courseware.cutm.ac.in/courses/data-structures-using-c/

Advanced Web Programming

Code	Course Title	Credit	T-P-PJ
CUTM1030	Advanced Web Programming	4	1-2-1

Objective

- Understand client server architecture and able to use the skills for web project development.
- Create job opportunities as a web developer

Course outcome

- Develop a static, interactive and well-formed webpage using JavaScript, CSS3 and HTML5.
- Use PHP7 to improve accessibility of a web document.
- Gain necessary skills for designing and developing web applications.

Course content

Module I: Web Programming Concepts(7hrs)

Architecture of the Web (1)

HTTP Protocols(1)

Difference HTTP1.0 and HTTP 1.1,Stateless nature of the protocol,Methods (GET, POST, HEAD, PUT, DELETE),HTTP session,Statuscodes,Persistentconnections,HTTPS

HTML(1)

Document Object Model (DOM),Elements,Events

HTML 5(2)

Elements,Objects,Events,Canvas,Audio& Video Support,Geo-location Support

CSS(2)

Styling HTML with CSS,Inline Styling (Inline CSS),External Styling (External CSS),CSS Fonts,The CSS Box Model,The id Attribute,The class Attribute,HTML Style Tags

Practice

1. Write an HTML code to display your CV on a web page.
2. Write an HTML code to create a Home page having three links: About Us, Our Services and Contact Us. Create separate web pages for the three links.
3. Write an HTML code to create a Registration Form. On submitting the form, the user should be asked to login with this new credentials.
4. Write an HTML code to create your Institute website, Department Website and Tutorial website for specific subject.

5. Write an HTML code to create a frameset having header, navigation and content sections.
6.Write an HTML code to demonstrate the usage of inline CSS.
7. Write an HTML code to demonstrate the usage of internal CSS.
8. Write an HTML code to demonstrate the usage of external CSS.
9: Design your own website using HTML CSS
10: Design form using HTML and apply CCS

Module II: JavaScript & j Query(14 hrs)

JavaScript (10)

Introduction to JavaScript:Variable, statements, Operators, Comments, constructs, Functions, expressions,Javascriptconsole,Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date,Methods,Arrays, Array Methods,Booleans, Comparisons Control Structures:Conditions, Switch,Loop For, Loop While, Break

Functions: Function Definitions, Function Parameters, Function Invocation, Function Closures

Objects: Object Definitions, Object Properties, Object Methods, Object Prototypes

Object Oriented Programming:

Method,Constructor,Inheritance,Encapsulation,Abstraction,Polymorphism,JavascriptValidations, Document Object Model, Document and Events (DOM Manipulation)

HTML DOM: DOM Intro, DOM Methods, DOM Document, DOM Elements, DOM HTML, DOM CSS,DOM Animations, DOM Events, DOM EventListener, DOM Navigation, DOM Nodes, DOM Nodelist,Debugging,Type Conversion, Regular expressions, Errors, Debugging

Forms: Forms Validation, Forms API,JS Browser BOM, Window, Screen, Location, History, Navigator, Popup Alert, Timing, Cookies,JavascriptWindows,Pushing code quality via JSLinttool,Security in Java Script

jQuery(4)

Basics of jQuery,jQuery selection and events, jQuery Effects,jquery traversal and manipulation,Data attributes and templates, jQuery Plugins,Jquery / Google Web Toolkit

Practice:

1.Write a Java script to prompt for users name and display it on the screen.

2. Design HTML form for keeping student record and validate it using Java script.

3. Write programs using Java script for Web Page to display browsers information.

4: Validate form page using JavaScript

5: use JQuery effect in page

6. Write a jQuery Code to Find the data passed with the on() method for each element.

7.Find the position of the mouse pointer relative to the left and top edges of the document.

8.Count the number of milliseconds between the two click events on a paragraph

9.Find all the text nodes inside a paragraph and wrap them with an italic tag

Module III: AJAX& JSON(8 hrs)

AJAX(3)

Design Introduction to Ajax,Web services and Ajax,Ajax using HTML, CSS, JavaScript,Ajax Framework and DOM,XMLHttpRequest,Ajax Architecture

Working with JSON (5)

JSON – Introduction,Need of JSON,JSON Syntax Rules,JSON Data - a Name and a Value,JSONObjects,JSONArrays,JSON Uses JavaScript Syntax,JSONFiles,JSON& Security Concerns, Cross Site Request Forgery (CSRF), Injection Attacks,JSXMLHttpRequestfunctions,JavaScriptXMLHttpRequest& Web APIs,JSON& Client Side Frameworks,JSON& Server Side Frameworks,Replacing XML with JSON,JSON parsing,AJAX using JSON and jQuery

Practice:

1.Create an simple application using AJAX to show the table of numbers given by user at runtime.

2.Access web service using Ajax and handle using JSON

Module IV: Responsive Web Design (5 hrs)

Introduction

The Best Experience for All Users

• Desktop

• Tablet

• Mobile

Bootstrap

Overview of Bootstrap

Need to use Bootstrap

Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid

Typography

Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress

Bars, Pagination, List Groups, Panels, Dropdowns, Collapse, Tabs/Pills, Navbar, Forms, Inputs

Bootstrap Grids, Grid System, Stacked/Horizontal

Bootstrap Themes, Templates

Practice:

1.Create a responsive website using bootstap

Module V: PHP(10 hrs)

PHP(10):

Introduction to PHP,Working with arrays,Functions,Forms,Handling date and Times,Working with Files,Session and state management,Database operations from PHP

Practice:

1.Develop student registration web application using PHP

2.Write a PHP database application that collects comments from users and makes it possible for users to view all the comments that have been submitted. You will need three files: an HTML page with a form where the user can enter a comment; a PHP program to process the input from this form by adding the comment to the database; and a PHP program that displays all the comments.

Module VI: Introduction to Drupal(5 hrs)

Drupal Basics, Content Management System, Content Management Framework, Web Application,Framework,Drupal Workflow, Bootstrap, hooks, callbacks, output,Modules (Core and Contributed), Nodes, Blocks, Regions,The Admin Interface (Overview),Content Management, Site Building, Site Configuration, User Management, Reports, Help,ContentTranslation,User Contributed Modules,Layouts in Drupal, File Systems

Practice:

1.Setup Drupal server and develop a site on it

Module VII: XML & Web Security (6 hrs)

(6 hrs)

XML (2)

Introduction to XML,XML Validation,Reason for XML,XML Tree Structure, XML DOM,XML DTD,XML Schema

XML style language(2)

XML and XSLT, XML Parsing,XML parsers (DOM & SAX),XML WSDL,RSS Feed

Web Security(2)

SQL Injection,Cross-Site Scripting (XSS),Security standards (OWASP)

Practice:

1. Creating XML Document

2.DTD creation

3.Test SQL Injection for student resgistration application

Text/Reference Books

1.Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Black Book Kindle Edition,byKogent Learning Solutions Inc.

2.HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed Kindle Edition,by DT Editorial Services

3.Programming PHP: Creating Dynamic Web Pages, Third Edition, by Kevin Tatroe, O'REILLY

4.Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON kindle Edition by Lindsay Bassett,O'REILLY

5.Bootstrap: Responsive Web Development by Jake Spurlock, Paperback

Project Work

1.Online Quiz System

2.Online Student feedback System

3.. Online Tutorial System

4.Restaurant Billing System

5.Online MCQ Database Bank System

Source of reference:https://nqr.gov.in/qualification-title?nid=3002

Courseware Link: http://courseware.cutm.ac.in/courses/advanced-web-programming/**Course**

Java Technologies

ode	Course Title	Credit	T-P-PJ
CUTM1031	Java Technologies	4	2-1-1

Objective

- Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- Be aware of the important topics and principles of software development
- Have the ability to write a computer program to solve specified problems
- Have the ability to write a computer program to solve specified problems
- Be able to use the Java SDK environment to create, debug and run simple Java programs

Course outcome

- Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs
- Read and make elementary modifications to Java programs that solve real-world problems
- Identify and fix defects the common safety issues in code
- Document a Java program using Javadoc
- Use a version control system to track source code in a project
- Qualify confidently any interview process where Java is the requirement

Course content

Module I: Introduction to Java (8 hrs)

Features and Installation, Java Programming Basics, Decision Making and Looping, Class and Object, Inheritance
Practice 1 (1 Hr)
Practice 2 (1 Hr)

Module II: Package and Safe Code (5 Hr)

Interfaces, Packages and Access Protection, Exception Handling (Fault Tolerant Programming)
Practice 3 (1 Hr)

Module III: Collection and Threads (5 Hr)
ArrayList, Vector, Set, Map, Multi-threaded Programming, Synchronization
Practice 4 (1 Hr)

Module IV: Language and Utility Packages (5 Hr)
String Handling, Wrappers, Runtime Memory Management, Cloning, Calendar, Date and Time Facilities, Scanner, Internationalization
Practice 5 (1 Hr)
Practice 6 (1 Hr)
Module V: Input/ Output and Applets (5 Hr)
Byte and Character Stream I/O, Persistence, Applet: Architecture, Skeleton, and Implementation
Practice 7 (1 Hr)
Practice 8 (1 Hr)
Module VI: GUI Programming (5 Hr)
AWT: Container, Components, Layout Managers, Event Handling
Practice 9 (1 Hr)
Practice 10 (1 Hr)
Module VII: Networking and Advanced (5 Hr)
Networking Fundamental, Client-Server Communication, Remote Method Invocation (RMI), Java Virtual Machine (JVM) Tuning, Java Profiler
Practice 11 (1 Hr)
Practice 12 (1 Hr)

Text Book(s):

1. Java The Complete Reference, Fifth Edition, C25 Herbert Schildt, McGraw-Hills

Reference Book(s):

1. Murach's Java Programming, 5th Edition, Joel Murach, Mike Murach & Associates, 2011, ISBN-78-1-943872-07-7
2. Introduction to Java Programming, Comprehensive, 10th ed., Y. Daniel Liang, 2014. ISBN-10: 0133813460, ISBN-13: 9780133813463

Course outline Prepared by: Sashi Bhusan Maharana
Date: 28th May 2020

Source of reference;

https://nqr.gov.in/qualification-title?nid=3002
https://www.cdac.in/index.aspx?id=DAC&courseid=0#
https://canvas.harvard.edu/courses/63117/assignments/syllabus
https://canvas.harvard.edu/courses/69911/assignments/syllabus
https://xid.harvard.edu/xid-apps/submitAccountForm.do

YouTube Resources:	freeCodeCamp.org Codearchery Edureka free project Jenkov

Online Source(s):

1. https://docs.oracle.com/javase/tutorial/java/index.html

2. https://www.programiz.com/java-programming

3. https://marcus-biel.com/

Software/Tool(s): Java 8, Eclipse IDE

Online Compiler: https://ideone.com/

Online Coding Practice: https://www.hackerrank.com/

List of Practices:

Practice 1 (Module-I)

Program-1:

Write a program that computes the standard deviation of a set of floating point numbers that the user enters. First the user says how many numbers N are to follow. Then the program asks for and reads in each floating point number. Finally it writes out the standard deviation. The standard deviation of a set of numbers Xi is:

SD = Math.sqrt(avgSquare - avg2)

Here, avg is the average of the N numbers, and avg2 is its square.

avgSquare is the average of Xi * Xi. In other words, this is the average of the squared value of each floating point number.

For example, if N = 4, say the numbers were:

Xi Xi * Xi

2.0 4.0

3.0 9.0

1.0 1.0

2.0 4.0

sum 8.0 18.0

Now:

avg = 8.0/4 = 2.0

avg2 = 4.0

avgSquare = 18.0/4 = 4.5

SD = Math.sqrt(4.5 - 4.0) = Math.sqrt(.5) = 0.7071067812

To do this you will need to do several things inside the loop body for each floating point value as it comes in: add it to a sum, square it and add it to a sum of squares. Then after the loop is finished apply the formula.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 2 (Module-I)

Program-1:

Better encapsulation of the Goods class would call making instance variables private and using getter and setter methods to access them. A further refinement would be to make the class abstract and to define additional child classes. Here is a revised Goods class:

```
public abstract class GoodsSGA
{private String description;
private double price;
private int quantity;
```

public GoodsSGA(String des, double pr, int quant)

{description = des;price = pr;

quantity = quant;}

double getPrice()

{return price;}

void setPrice(double newPrice)

{price = newPrice;}

int getQuantity()

{return quantity;}

void setQuantity (int newQuantity)

{quantity = newQuantity;}

public String toString()

{return "item: " + description + " quantity: " + quantity + " price: " + price ;}}

Revise the source code for the classes Food, Toy, and Book. (Perhaps call the revised classes FoodSG, ToySG, and BookSG.) create a new class ToiletrySG for things like bubble bath. Create a new testing class, StoreSG to test your revised classes.

Note: the child classes will need to use the getter and setter methods to access the instance variables that are declared as private in GoodsSG.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 3 (Module-II)

Program-1:

User-Friendly Division Practice:

Put in a loop so that the user is repeatedly asked for the numerator and the divisor. For each set of data, the program prints out the result, or an informative error message if there is a problem (division by zero or poor input data).

The program continues looping, even if there is a problem Exit the loop when data entered for the numerator start with characters "q" or "Q". Don't print out an error message in this case. Don't ask for the divisor if the user just asked to quit.

Here is sample output from one run:

Enter the numerator: 12

Enter the divisor: 4

12 / 4 is 3

Enter the numerator: 12

Enter the divisor : 0

You can't divide 12 by 0

Enter the numerator: glarch

You entered bad data.

Please try again.

Enter the numerator: quit

You will need to use the method charAt() from the String class.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 4 (Module-III)

Program-1:

In mathematics, several operations are defined on sets. The union of two sets A and B is a set that contains all the elements that are in A together with all the elements that are in B. The intersection of A and B is the set that contains elements that are in both A and B. The difference of A and B is the set that contains all the elements of A except for those elements that are also in B.

Suppose that A and B are variables of type set in Java. The mathematical operations on A and B can be computed using methods from the Set interface. In particular:

A.addAll(B) computes the union of A and B; A.retainAll(B) computes the intersection of A and B; and A.removeAll(B) computes the difference of A and B. (These operations change the contents of the set A, while the mathematical operations create a new set without changing A, but that difference is not relevant to this exercise.)

For this exercise, you should write a program that can be used as a "set calculator" for simple operations on sets of non-negative integers. (Negative integers are not allowed.) A set of such integers will be represented as a list of integers, separated by commas and, optionally, spaces and enclosed in square brackets. For example: [1,2,3] or [17, 42, 9, 53,108]. The characters +, *, and - will be used for the union, intersection, and difference operations. The user of the program will type in lines of input containing two sets, separated by an operator. The program should perform the operation and print the resulting set.

Here are some examples:

Input	Output
[1, 2, 3] + [3, 5, 7]	[1, 2, 3, 5, 7]
[10,9,8,7] * [2,4,6,8]	[8]
[5, 10, 15, 20] - [0, 10, 20]	[5, 15]

To represent sets of non-negative integers, use sets of type TreeSet<Integer>. Read the user's input, create two TreeSets, and use the appropriate TreeSet method to perform the requested operation on the two sets. Your program should be able to read and process any number of lines of input. If a line contains a syntax error, your program should not crash. It should report the error and move on to the next line of input. (Note: To print out a Set, A, of Integers, you can just say System.out.println(A). We've chosen the syntax for sets to be the same as that used by the system for outputting a set.)

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 5 (Module-IV)

Program-1:

Password Checker:

Write a program that repeatedly asks the user for a proposed password until the user enters an acceptable password. When the user enters an acceptable password, the program writes a message and exits.

Acceptable passwords:

Are at least 7 characters long.

Contain both upper and lower case alphabetic characters. Contain at least 1 digit. The logic of this program can be quite tricky. Hint: use toUpperCase(), toLowerCase, and equals(). You will also need nested ifs.

Here is a run of the program:

Enter your password:

snowflake

That password is not acceptable.

Enter your password:

SnowFlake

That password is not acceptable.

Enter your password:

snowflake47

That password is not acceptable.

Enter your password:

Snowflake47

Acceptable password.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 6 (Module-IV)

Program-1:

Secret Code:

A text message has been encoded by replacing each character of the message with an integer. Each integer is an index into a key-phrase that contains all the lower case letters of the alphabet as well as the space character. The key-phrase may contain the same character in several locations. The encoded text is series of integers, like this:

35 10 10 33 9 24 3 17 41 8 3 20 51 16 38 44 47 32 33 10 19 38 35 28 49

To decode the message, look up each integer in the key-phrase and output the corresponding character. For example, say that the key-phrase is this (the index of each character has been written above it):

1111111111222222222233333333334444444445 5

01234567890123456789012345678901234567890123456789012345678901

six perfect quality black jewels amazed the governor

using each integer from the encoded text as an index into the phrase results in the decoded message:

attack the bridge at dawn

Write a program that decodes a secret message contained in a text file. The first line of the text file contains the key-phrase. Then the file contains a sequence of integers, each of which indexes the key-phrase. Find the character corresponding to each integer and output the secret message. Note if a character character such as 'e' occurs several places in the key-phrase it may be encoded as different integers in different parts of the secret message.

(The recipient of the secret message gets only the file of integers and must put the key-phrase at the top of the file.) For example, here is the contents of a secret message file ready for the program:

six perfect quality black jewels amazed the governor

35 10 10 33 9 24 3 17 41 8 3 20 51 16 38 44 47 32 33 10 19 38 35 28 49

Here is a sample run of the program:

C:\> java Decode < secretFile.txt

attack the bridge at dawn

You will need the charAt() method of String.

Here is another secret message file, with key-phrase inserted, that you can use to test your program:

six perfect quality black jewels amazed the governor

31 16 2 3 4 42 48 7 27 9 10 43 12 13 35 15 1 40 18 3

20 15 33 23 24 32 26 29 28 27 21 31 25 14 34 14 36

42 38 19 40 41 27 3 44 50 46 42 48 49 50 6

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 7 (Module-V)

Program-1:

Stop Word Remover:

Write a program that reads in a file of text, perhaps the text of a novel. The program copies the same text to an output file, except that all the useless words such as "the", "a", and "an" are removed. (Decide on what other words you with to remove. The list of words removed is called a stop list.) Do this by reading the text file token by token using hasNext() and next(), but only writing out tokens not on the stop list.

Prompt the user for the names of the input and output files.

Fairly Easy: The output file will have only N tokens per line. Do this by counting tokens as you output them. N will be something like 10 or 12.

Improved Program: Preserve the line structure of the input file. Do this by reading each line using nextLine() and then creating a new Scanner for that line. (Look at the on-line

documentation for Scanner.) With each line's Scanner, use hasNext() and next() to scan through its tokens.

Harder: Write out no more than N characters per line. N will be something like 50. Do this by keeping count of the number of characters written out per line. The length() method of String will be useful. If X characters has already been written to the current line, and if X plus the length of the current token exceeds N, then start a new line.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 8 (Module-V)

Program-1:

E-Mail Address Extractor:

Write a program that scans a text file for possible e-mail addresses. Addresses look like this: someone@somewhere.net

Read tokens from the input file one by one using hasNext() and next(). With the default delimiters of Scanner, an entire e-mail address will be returned as one token. Examine each token using the indexOf() method of String. If a token contains an at sign @ followed some characters later by a period, regard it as a possible e-mail address and write it to the output file. Programs such as this scan through web pages looking for e-mail addresses that become the targets of spam. Because of this, many web pages contain disguised e-mail addresses that can't easily be automatically extracted.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 9 (Module-VI)

Program-1:

User-friendly Fat Calculator, with Advice:

Further modify the calories from fat calculator so that it includes another TextField that will be set with the text "Too many fat calories" if the percentage of calories from fat is equal or greater than 30 percent, or to "Healthy amount of fat" if the percentage is less than that.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 10 (Module-VI)

Program-1:

Three Button Monte:

Write a program to implement a game:

There are three buttons in the frame. Two of the buttons cause the program to quit using System.exit(0); the remaining button changes the frame to green (a win!) The winning button is different each time the game is played.

The easy way to do this (although it seems unfair to the user) treats each button the same way. The actionPerformed() method does not check which button was clicked. When any button is clicked, the method picks a random integer from 0 to 2 and performs the "winning" action if the integer happens to be 0. Otherwise, it performs the "losing" action. To the user, it seems like there is a "winning" button and two "losing" buttons. But, in fact, it does not matter which button was clicked.

This is similar to some electronic gambling devices in casinos, where it appears to the user that there are "winning moves" and "losing moves" but in fact the machine actually ignores what the user has done and just declares a "win" every now and then, according to predetermined odds.

You will need the Random class:

Random randNum = new Random(); // create a Random number object

int someInt = randNum.nextInt(3); // someInt gets a number from 0 to 2

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

https://www.hackerrank.com/domains/java

Practice 11 (Module-VII)

Content Delivery with Networking:

Write a Client-Server program where the client queries with a name of file and the server delivers the content of requested files to the client over the network.

(Improve the program by making the server multi-threaded)

Practice 12 (Module-VII)

Greet the user with Remote Method Invocation:

Write a program using RMI, where the user invokes a method on remote object with username as parameter and receives a greeting message based on time of the day along with username.

Projects

However, not limited to:

1. Chat application
2. Text Editor application
3. GUI based Scientific Calculator
4. Paint application
5. Slam book

(*PROJECT REVIEWS WILL COMMENCE BEYOND CLASS HOURS)

Monitoring:
Credit will be received only on making an honest effort. It is expected that students will finish watching all lecture video and complete all challenge problems by the end of each lecture week.
Borrowing code from other sources is allowed only with proper attribution and credit given to the original author(s).

List of Common Programs to solve using Java:

1. Program to calculate area of a triangle

3. Program to solve quadratic equation

3.Program to swap two zariables (with and without using third variable)

4.Program to generate random numbers in various ways

5.Program to convert miles to kilometers and vice-versa

6.Program to convert celsius to fahrenheit and vice-versa

7.Program to check if a number is odd or even

8.Program to check if input year is leap year

9.Program to test primality

10.Program to print all prime numbers in an interval using "Sieve of Eratosthenes"

11.Program to generate factorial of all elements in an array

12.Program to display the multiplication table up to 20

13.Program to print the fibonacci sequence

14.Program to check armstrong number, perfect number, Harshad number

15.Program to generate armstrong numbers in an Interval

16.Program to find the sum of Harshad numbers in an interval

17.Program to display powers of two Using lambda

18.Program to perform conversions among decimal to binary, octal and hexadecimal

19.Program to display ASCII table

20.Program to find HCF/GCD and LCM

21.Program to find factors of given natural number

22.Program to make a simple calculator

23.Program to shuffle deck of cards

24.Program to generate fibonacci sequence using recursion

25.Program to find sum of natural numbers using recursion

26.Program to find factorial of number using recursion

27.Program to convert decimal to binary using recursion

28.Program to add two matrices

29.Program to obtain transpose of a matrix

30.Program to multiply two matrices

31.Program to check if a string is palindrome

32.Program to remove punctuations from a string

33.Program to sort words lexicographically

34.Program to illustrate different set operations

35.Program to count frequency of each vowel in a string

36.Program to find hash value of a file

Embedded System Programming with ARM-Cortex

Code	Course Title	Credit	T-P-PJ
CUTM1039	Embedded System Programming with ARM-Cortex	6	3-2-1

Objective

- To allow students in Embedded System sectors to learn programming / Interfacing peripherals to ARM Cortex based Microcontroller

Course outcome

- Describe the architectural features and instructions of 32 bit ARM Cortex M3 microcontroller.
- Understand the basic hardware components and their selection method based on the characteristics and attributes of an Embedded System.
- Understand various Sensors, Actuators & Interfacing Modules.

Course content

Module I: EMBEDDED C (4 Hrs)

Embedded System, Programming Embedded system, Factor for selecting the Programing language, Embedded C programming Language, Embedded C vs C.

Practice:

1. Familiarization with tools (STM32CubeMX, KeiluVision IDE, Flash Magic & Proteus Simulator).
2. Programming STM32 using KeiluVision& STM32CubeMX.

Module II: ARM-32 bit MICROCONTROLLER (6 Hrs)

ARM Design Philosophy & RISC Architecture, Programmer's Model. ARM Cortex M, Cortex M Architecture, ARM Cortex-M Internals & Debugging.

Practice:

1. Familiarization with Different Processors and Controllers Boards (8, 16, 32, 64 bits)

Module III: STM32 GPIO MANAGEMENT (14 Hrs)

GPIO Configuration, Driving De-initialization, Interfacing IO devices and its type – LEDs, Switches, Buzzer, Seven Segment Display, LCD (4 bit, 8 bit Mode), Keypad (4*4), DC Motor, Stepper Motor, Servo motor, Relay.

Practice:

1. Write an Embedded C program to interface LEDs with STM32.
2. Write an Embedded C program to interface Switch with STM32.
3. Write an Embedded C Program to design up counter & down counter using Seven Segment Display. (1 digit, 2 digit)
4. Write an Embedded C program to interface buzzer to control with the help of Switch.
5. Write an Embedded C program to display characters on Alphanumeric LCD.
6. Write an Embedded C program to interface Keypad and LCD with STM32.
7. Write an Embedded C program to interface DC Motors, Stepper Motor, and Servo Motor rotate clockwise, anticlockwise and in angle (45^o, 90^o, 180^o).
8. Write an Embedded C program to interface relay to control the AC Appliances.

Module IV: STM32 INTERRUPT MANAGEMENT & UART (14 Hrs)

NVIC Controller, Enabling Interrupt, Interrupt Priority Levels, UART Initialization, UART communication in polling Mode & in Interrupt Mode. Wireless Technologies- Bluetooth, Wi-Fi, RF.

Practice:

1. Write an embedded C program to generate an Interrupt process using STM32.
2. Write an Embedded C program to interface STM32 to Bluetooth Module to send & receive Data.
3. Write an Embedded C program to interface STM32 to GPS module to get a Location Coordinate.
4. Write an Embedded C program to interface STM32 to GSM module to Send & Receive SMS.
5. Write an Embedded C program RF module with STM32 to send and receive the data wirelessly.
6. Write and Embedded C program to design a system to read the RFID cards using STM32.
7. Write and Embedded C program to connect ESP8266 with STM32 to create a Webserver.

Module V: STM32 TIMERS , ADC, & DAC (10 Hrs)

Timers Basics, General Purpose Timer, SysTick Timer, ADC & DAC Basics, Initialization, DAC Peripherals & Modules. Analog Sensors and its Types(Ultrasonic Sensor, Temperature, Humidity, Soil Moisture Sensor, PIR sensor)

Practice:

1. Write an Embedded C Programs to generate Delay using Timer.
2. Write an Embedded C program to display output for given analog input using internal ADC. (Use of Analog Sensors like Ultrasonic Sensor, Temperature, Humidity, Soil Moisture Sensor, PIR sensor)
3. Write an embedded C program to generate Triangular and Square waves using DAC.

Module VI: STM32 I2C & SPI (10 Hrs)

I2C specification, Protocol configuration, I2C Peripherals. SPI Specification, Protocol configuration, it's Peripheral and Modules.

Practice:

1. Write an Embedded C program to build I2C communication between STM32 and Arduino
2. Write an Embedded C program to build SPI communication STM32 to the Arduino board.

Module VII: PWM & CAN (8 Hrs)

RTC feature and its Module, CAN Protocols Overview, Application, Architecture, Data Transmission & Data Frames.

Practice:

1. Write an Embedded C program to implement a Real-Time Clock.
2. Write an Embedded C program to Speed Control of DC motor using PWM.
3. Write an Embedded C program to change the intensity of Light using PWM.

Text Books:

1. Shibu K V, ―Introduction to Embedded Systems‖, Tata McGraw Hill Education Private Limited, 2nd Edition
2. Noviello, Carmine. "Mastering STM32." Obtenido de http://www2. keil. com/mdk5/uvision,2017.
3. Norris, Donald. Programming with STM32: Getting Started with the Nucleo Board and C/C++. McGraw Hill Professional, 2018.

Reference Books:

1. STM32F10xx User Manual
2. https://www.udemy.com/course/stm32cubemx-complete-training/learn/lecture/9606338#overview

1.https://www.udemy.com/course/embedded-c-programming-for-embedded-systems/

VLSI Design

Code	Course Title	Credit	T-P-PJ
CUTM1040	VLSI Design	6	3-2-1

Objective

- The objective of the course is to provide understanding of the entire logic design process with the analysis from combinational and sequential digital circuit design.
- Provide understanding of the techniques essential to the Verilog programming for Verification and Testing.
- To learn the architecture of most prominent vendor in the FPGA market, Xilinx FPGAs and Altera FPGAs.

Course outcome

- Analyze combinational and sequential circuit design concepts.
- Develop FSMs & ASMs for the given problems.
- Write Verilog code, compile, simulate and execute on any VLSI design platform.
- Apply Verilog HDL for FPGA Programming.
- Implement Digital Circuits on Xilinx FPGAs and Altera FPGAs using Verilog HDL.

Course content

Module: Introduction to VERILOG (10 hrs)

Introduction to Verilog HDL & Hierarchical Modeling Concepts,Lexical Conventions &DataTypes,System Tasks & Compiler Directives,Modules, Ports and Module Instantiation Methods,Modeling methods,Design Verification using Test benches

Practice

1. Introduction to Xilinx EDA Tool.
2. Introduction to XST Tool and ISIM Tool
3. Xilinx Tool Flow: Simulation and Synthesis
4. Module and Ports in Verilog
5. Data Types in Verilog Programing.

Module II: Boolean Algebra and Logic Minimization (8hrs)

Binary Arithmetic and 1's and 2's Complementation, Basic Theorems and Properties, Canonical and Standard Form, AlgebraicSimplification of Digital Logic Gates,The Karnaugh Map Method, Prime and Essential Implications, Don't Care Map Entries.

Practice

1. Gate level Modelling in Verilog.
2. Data flow Modelling in Verilog.
3. Behavioral Modelling in Verilog.

Module III: Combinational Circuit Design (12hrs)

Arithmetic Circuits: Adder/Subtractor Circuits, Ripple Carry Adder, Universal Ripple carry Adder, BCD Adder, MultipliersComparators, Multiplexer, Demultiplexer, Decoder, Encoder and Priority Encoder, Code Converters: Binary to Gray, Binary to BCD.

1. Design of Arithmetic Circuits using Verilog.
2. Design of Encoder and Decoder using Verilog.
3. Design of Data selector and Data Distributor using Verilog.
4. Design of comparator and Code converters using Verilog.

Module IV: Sequential Circuit Design (14hrs)

Latch, Flip-Flop: S-R,D,J-K,T, Flip-Flop Conversion and Excitations Counter: Asynchronous and Synchronous counter Design, Register: SISO, SIPO,PISO and PIPO, Universal Shift Register, Johnson counter and Ring Counter.

Practice

1. Design SR and D-Flip Flop Using Continuous and Procedural Assignments.
2. Design JK-Flip Flop And T-Flip Flop Using Verilog.
3. Design Shift Registers (SISO, SIPO, PISO, PIPO) using Verilog.
4. Design Ripple Counter and Up/Down Synchronous Binary Counter Using Verilog.

Module V: State Machines (10 hrs)

Basic Finite state machines (FSM) structures, Mealy and Moore type FSM,Design of controller and Data path units,Controller Design using FSMs & ASMs

Practice

1. Design of Sequence Detectors allowing overlapping as well as non-overlapping.
2. Design of Mealy and Moore type FSM using Verilog.

3. Design of data controller using ASM.

Module VI: FPGA Architecture and Prototyping (5 hrs)

Introduction to Programmable Logic and FPGAs, Popular CPLD & FPGA Families, Architecture of Xilinx and Altera FPGAs

Practice

1. Proto-typing of a design using FPGA Design Kit

Module VII: Synthesis and Timing (6hrs)

FPGA Design Flow,Implementation Details Advanced FPGA Design tips, Logic Synthesis for FPGA, Static Timing Analysis

Practice

1. Design mapping and optimization
2. Analyze and resolve design problems
3. Report generation
4. Verilog gate-level netlist generation and post-synthesis timing data (SDF) extraction
5. Design constraints generation for placement and routing

Text Books:

1. M.Morris Mano., "Digital Design", Pearson Education, 4th Edition.
2. Palnitkar, S. (2003). Verilog HDL: a guide to digital design and synthesis (Vol. 1). Prentice Hall Professional.

Reference Books:

1. Kohavi, Z., & Jha, N. K. (2009). Switching and finite automata theory. Cambridge University Press.
2. Jain, R. P. (2003). Modern digital electronics. Tata McGraw-Hill Education.

Electronic Devices and Systems

Code	Course Title	Credit	T-P-PJ
CUTM1046	Electronic Devices and Systems	3	2-1-0

Objective

- The course is designed to be a broad introduction to electronic systems for students from diverse engineering disciplines. Completing the course will provide the necessary foundation to understand the role, capabilities and constraints of electronics in contemporary engineering systems.
- This course develops a basic understanding of the fundamentals and principles of analog and digital circuits and electronic devices. This understanding is a critical step towards being able to design new electronic circuits or use them appropriately as part of a larger engineering system.

Course outcome

- Understand operation of semiconductor devices.
- Understand the current voltage characteristics of semiconductor devices.
- Design and analyze of electronic circuits.
- Evaluate frequency response to understand behavior of Electronics circuits
- Analyze dc circuits and relate ac models of semiconductor devices with their physical Operation.
- To apply concepts for the design of Regulators and Amplifiers.
- To verify the theoretical concepts through laboratory and simulation experiments.
- To implement mini projects based on concept of electronics circuit concepts.

Course content

Module I: Semiconductor Diodes, Diode Applications (5 Hrs)

Semiconductor Physics, Semiconductor Diode and analysis, Zener Diodes, Light-Emitting Diodes, realization of logic gates using diodes, rectifier circuits.

Practice:

1. V-I Characteristics of PN Junction diode and ZENER diode. (Both hardware and MULTISIM)
2. Implementation of Full-Wave rectifier. (Both hardware and MULTISIM).
3. Implementation of AND/OR gates. (Both hardware and MULTISIM)

Module II: Bipolar Junction Transistors, Field-Effect Transistors (4 Hrs)

Transistor Construction and characteristics, Construction and Characteristics of JFETs, MOSFET

Practice:

1. Input - Output Characteristics of CB/CE/CC configuration BJT. (Both hardware and MULTISIM)
2. Output and transfer Characteristics of FET. (Both hardware and MULTISIM)

Module III: DC Biasing—BJTs. BJT AC Analysis, FET Biasing , FET Amplifiers , BJT and JFET Frequency Response (6 Hrs)

Load-Line Analysis, Operating Point, biasing techniques, AC analysis of BJT and FET, General Frequency Considerations, Low-Frequency Analysis of BJT, FET

Practice:

1. Frequency response of self biased BJT amplifier. (Both hardware and MULTISIM)
2. Frequency response of CS FET amplifier. (Both hardware and MULTISIM)

Module IV: Operational Amplifiers, Op-Amp Applications, Linear-Digital ICs (6 Hrs)

Practical Op-Amp Circuits and various parameter analyses, various applications, Timer and PLL ICs

Practice:

1. Design and implement of Adder, Subtractor using IC 741. (Both hardware and MULTISIM)
2. Design and implement of Integrator and Differentiator using IC 741. (Both hardware and MULTISIM)
3. Design and implement of any oscillator using IC 741. (Both hardware and MULTISIM

Module V: Power Amplifiers, Feedback and Oscillator Circuits (5 Hrs)

Various classes of power amplifier and their characteristics, design of various oscillators and its applications

Practice:

1. Design and implement of Class-C power amplifier (Both hardware and MULTISIM)
2. Design and implement of RC Phase shift oscillator (Both hardware and MULTISIM)

3. Design and implement of UJT relaxation oscillator (Both hardware and MULTISIM)

Module VI: Power Supplies (Voltage Regulators) (4 Hrs)

Various filters, Voltage regulator, Design and different IC voltage regulators.

Practice:

1. Regulation characteristics of transistor voltage regulator.(Both hardware and MULTISIM).
2. Regulation characteristics of IC7805/7912 voltage regulators.(Both hardware and MULTISIM).

Module VII: Other Two-Terminal Devices, *pnpn* and Other Devices (4 Hrs)

Varactor diode, photodiode, solar cell, tunnel diode, LCD, thermisters, SCR construction, applications, UJT

Text Books:

1. Electronic Devices and Circuit Theory, Eleventh Edition: Robert L. Boylestad. Louis Nashelsky
2. Electronic Principles and Applications , Ninth Edition: Charles A. Schuler

Reference Books

1. Electronic Devices and Circuits : GSN Raju
2. Electronic Devices: Systems and Applications: Robert Diffenderfer.

Link for Text Book

1: www.rtna.ac.th/departments/elect/Data/EE306/Electronic%20Devices%20and%20Circuit%20Theory.pdf

2: http://dl.booktolearn.com/ebooks2/engineering/electrical/9780073373836_Electronics_Principles_and_Applications_9th_7d8d.pdf

Electromagnetic Interference and Compatibility

Code	Course Title	Credit	T-P-PJ
CUTM1041	Electromagnetic Interference and Compatibility	3	2-0-1

Objective

- To familiarize with the fundamentals that are essential for electronics industry in the field of EMI / EMC
- To understand EMI sources and its measurements
- To understand the various techniques for electromagnetic compatibility.
- Acquire broad knowledge of various EM radiation measurement techniques.
- Model a given electromagnetic environment/system so as to comply with the standards.

Course outcome

- Designing electronic systems that function without errors or problems related to electromagnetic compatibility.
- Diagnose and solve basic electromagnetic compatibility problems.
- Real-world EMC design constraints and make appropriate tradeoffs to achieve the most cost-effective design that meets all requirements.
- Understand the effect of EM noise in system environment and its sources.
- Identifying of EMI hotspot and various techniques like Grounding, Filtering, Soldering, etc
- Concepts of PCB tracing, termination and implementation
- Understanding about the functions of a ground, understanding about cables and connectors.
- Understanding the various aspects of shielding.

Course content

Module I: EMC Fundamentals (3hrs Theory)

Definition of EMI and EMC, Sources and Simulators, Propagation Methods.

Module II: Electrical Noise in EMI, EMI analysis and Suppression (3hrs Theory)

Basic Aspects of EMI in System Environment, Electrical Noise Sources, Common-Mode and Differential-Mode Currents, Power and/or Return Bounce, Identification of EMI Hot Spot, RF Current Return and Flux Cancellation, Loop Area between Circuit and components, Primary Grounding, Filtering, Shielding

Module III: EMC Standard and EMC Measurements (3hrs Theory)

Overview of EMC Standards, Radiated and Conducted Emission (RE/CE) Standards, Radiated and Conducted Immunity (RI/CI) Standards, Electrostatic Discharge (ESD) Standards, Overview of EMC Measurements, Testing Equipment, Radiated Emission Test Setup, Measurement of Signals and Noise, Interpretation of Measurement Results.

Module IV: PCB Trace Routing and Termination (3hrs Theory)

Typical PCB Trace Topologies, Trace Routing Design Guidelines, Routing Differential Pair Signals, Layer Jumping – Use of Vias, Routing over a Split Plane, Fundamental Concepts of Trace Termination, Termination Methodologies and Implementation, Simulation Examples.

Module V: Grounding Technique and Filtering (3hrs Theory)

Time (Transient) response of R-L, R-C, R-L-C series circuits for Zero input, Step input, pulse input -Initial conditions-solution method using differential equation and Laplace transforms.

Module VI: Shielding and Electrical Gaskets (3hrs Theory)

Transmission Line Theory of Shielding, Absorption Loss, Reflection Loss, Shielding Effectiveness, Shielding Materials, Apertures in Shielded Walls, Waveguide below Cut-off, The Need for Gaskets, Common Gaskets Material Use, Properties and Characteristics of RF Gaskets.

Module VII: Advanced EMI/EMC (3hrs Theory)

EMC/EMI Modeling Techniques and Applications, Virtual EMC Lab, New Radiation Testing Technology – from Near-field Scanning to Far-field Prediction,Novel Radiation Mitigations Design

Projects:

1. Common Impedance Coupling Simulation using CST Studio Suite
2. Lightning Strike Analysis using CST Studio Suite
3. Emissions Simulation via Cascading using CST Studio Suite
4. Electrostatic Discharge using CST Studio Suite
5. Simulating Shielding Effectiveness using CST Studio Suite
6. Simulating Conducted Emissions from a Motor Control using CST Studio Suite
7. Simulating Conducted Emissions from a DC/DC Converter using CST Studio Suite

Note: All existing VLSI projects/Circuit Boards/Machines can be simulated and tested above.

Text Books:

1. Henry W. Ott, “Electromagnetic Compatibility Engineering”, John Wiley & Sons, 2009
2. V.P. Kodali, “Engineering Electromagnetic Compatibility”, IEEE Publication, S. Chand & Co. Ltd., New Delhi.
3. Ralph Morrison, "Grounding and Shielding: Circuits and Interference”, John Wiley & Sons

Reference Books

1. “Electromagnetic Interference and Compatibility”, IMPACT series IIT-Delhi,Modules1-9
2. DraganPoljak, "Advanced Modeling in Computational Electromagnetic Compatibility”, John Wiley &Sons.
3. Dipak L. Sengupta and Valdis V. Liepa,"Applied Electromagnetics and Electromagnetic Compatibility”, John Wiley & Sons
4. Charles S. Walker, "Capacitance, Inductance and Crosstalk Analysis”, Artech House, 1990.
5. DANIEL MÅNSSON and RAJEEV THOTTAPPILLIL, "ELECTROMAGNETIC COMPATIBILITY, EMC", NPTEL
6. EMI/EMC Testing, Society of Applied Microwave Electronics Engineering and Research Sameer

Electromagnetic Field Theory & Transmission Lines

Code	Course Title	Credit	T-P-PJ
CUTM1042	Electromagnetic Field Theory & Transmission Lines	3	2-1-0

Objective

- To introduce the fundamental theory and concepts of electromagnetic waves and transmission lines
- To impart knowledge on the concepts of electrostatics, electric potential, energy density and their applications.
- To impart knowledge on the concepts of magnetostatics, magnetic flux density, scalar and vector potential and its applications.
- To impart knowledge on the concepts of Faraday‘s law, induced emf and Maxwell‘s equations.
- Model and design the transmission lines at high frequencies.
- To apply Smith chart use for solution of transmission line problems and impedance matching.

Course outcome

- Apply the principles of electrostatics to the solutions of problems relating to electric field and electric potential.
- Apply the principles of electrostatics to the solutions of problems relating to boundary conditions and electric energy density.
- Apply the principles of magnetostatics to the solutions of problems relating to magnetic field and magnetic potential,
- Apply the principles of magnetostatics to the solutions of problems relating to boundary conditions and magnetic energy density.
- Understand the concepts related to Faraday‘s law, induced emf and Maxwell‘s equations.
- Apply Maxwell‘s equations to solutions of problems relating to transmission lines and uniform plane wave propagation.

Course content

Module I: Electrostatics (3hrs Theory + 2hrs Practice)

Introduction to Electrostatic Fields, Gauss's Law and Applications, Electric Potential, Maxwell’s Two Equations for Electrostatic Fields,Electric Current and Current Density, Continuity Equation, Relaxation Time, Laplace’s and Poisson’s Equations.

Practice:

1. To Calculate the Electric field of a dipole using Coulomb's law in Matlab

2. Simulation of Electric Potential and Electric Field in Matlab

Module II: Magnetostatics (3hrs Theory + 2hrs Practice)

Biot-Savart Law: Current Flow – which path does it take, Ampere's Circuital Law, Magnetic Flux Density: Closed Loop Circuits, Magnetic Scalar and Vector Potentials, Forces due to Magnetic Fields, Inductances and Magnetic Energy.

Practice:

1. Magnetic field by an infinitely long line current using matlab

2. Magnetic field of a Circular current loop using Biot Savart's Law

Module III: Maxwell's Equations (3hrs Theory + 1hr Practice)

Maxwell's Equations and Boundary Conditions.

Practice:

1. Maxwell's Equation using matlab

Module IV: Electromagnetic Waves (3hrs Theory + 4hrs Practice)

Wave Equations for Conducting and Perfect Dielectric Media, Uniform Plane Waves, Wave Propagation in Lossless and Conducting Media, Polarization, Reflection and Refraction of Plane Waves – Normal and Oblique Incidences for both Perfect Conductor and Perfect Dielectrics, Brewster Angle, Critical Angle and Total Internal Reflection, Surface Impedance. Poynting Vector and Poynting Theorem.

Practice:

1. Linear and Circular Polarization of waves using matlab

2. 1-D standing wave using matlab

3. 2-D standing wave (TE) using matlab

4. 2-D standing wave (TM) using matlab

5. Design of Wireless Power Transfer using matlab

Module V: Introduction to Transmission Line Modelling (3hrs Theory + 3hrs Practice)

Introduction to Transmission line equations, Primary & Secondary constants Expressions for Characteristic Impedance, Propagation Constant, Phase and Group Velocities, Losslessness/Low Loss Characterization, Distortion , Loading, Transmission Line Effects, SC and OC Lines, Reflection Coefficient, VSWR, λ/8, λ/4, λ/2 line impedance Transformations, Smith Chart – Configuration and Applications, Impedance Control.

Practice:

1. Reflection and transmission of a plane wave (S-wave)
2. Reflection and transmission of a plane wave (P-wave)
3. Radiation by an infinitesimal dipole

Module VI: Waveguides (3hrs Theory)

Introduction, Rectangular Waveguides, electric and magnetic field patterns in TE10 and TE11 mode configuration, modes of TE wave in rectangular waveguide, field equations, impossibility of TEM wave propagation in waveguides, cutoff frequency of rectangular waveguide, propagation constant, wave impedance, phase velocity, group velocity, dominant mode and degenerate modes, related problems.

Module VII: Electromagnetic Computational Techniques (3hrs Theory)

Introduction, Finite Difference Method (FDM), Finite Element Method (FEM) and Method of moments (MOM) technique.

Text Books:

1. Matthew N.O. Sadiku, “Elements of Electromagnetics”, Oxford Univ. Press.
2. G.S.N.Raju, “Electromagnetic Field Theory and Transmission Lines”, Pearson Education (Singapore) Pvt., Ltd.

Reference:

1.E.C. Jordan and K.G. Balmain, “Electromagnetic Waves and Radiating Systems”, PHI.
2.Seungbum Hong, "Electrodynamics: An Introduction", Coursera.

3.Seungbum Hong, "Electrodynamics: Electric and Magnetic Field", Coursera.

4.Seungbum Hong, "Electrodynamics: In-depth Solutions for Maxwell's Equations", Coursera.

5.Husain Habib, "Electromagnetic Tutorials part 1 with MATLAB & GeoGebra", Udemy.

Network Analysis

Code	Course Title	Credit	T-P-PJ
CUTM1043	Network Analysis	3	2-1-0

Objective

- To learn techniques of solving circuits involving different active and passive elements.
- To analyze the behavior of the circuit's response in time domain.
- To analyze the behavior of the circuit's response in frequency domain.
- To synthesize an electrical network from a given impedance/admittance function.

Course outcome

- Apply the knowledge of basic circuital law and simplify the network using reduction techniques.
- Analyze the circuit using Kirchhoff's law and Network simplification theorems.
- Infer and evaluate transient response, Steady state response, network functions.
- Obtain the maximum power transfer to the load , and Analyze the series resonant and parallel resonant circuit.
- Evaluate two-port network parameters.

Course content

Module I: Basic Circuit Concepts and Basic Laws (3hrs Theory)

Circuit concepts –Resistor(R)-Inductor(L)-Capacitor(C)-Voltage and Current Sources - Source transformation-Voltage, Current relationship for passive bilateral elements -Ohm's law, Kirchhoff's laws.

Module II: Methods of Analysis (3hrs Theory)

Nodal analysis, mesh analysis, super node and super mesh, Nodal Versus Mesh Analysis for D.C excitations.

Module III: Circuit Theorems (3hrs Theory + 4hr Practice)

Superposition theorem- Thevenin's theorem- Norton's theorem- Maximum power transfer theorem- Reciprocity theorem.

Practice:

3. Verification of Superposition Theorem
4. Verification of Thevenin's Theorem
5. Verification of Maximum Power Transfer Theorem
6. Verification of Reciprocity Theorem

Module IV: Network Topology and Two Port Networks (3hrs Theory + 4hrs Practice)

Network topology, Incidence matrix, Tie-set matrix, Cut-set matrix, Dual networks- Two port network, Impedance Parameter, Admittance Parameter, Transmission line.

Practice:

4. Determination of Z parameters of Port Network
5. Determination of Y parameters of Port Network
6. Determination of h parametersof Port Network
7. Determination of ABCD parametersof Port Network

Module V: Time Response of Circuits (DC Excitation) (3hrs Theory)

Time (Transient) response of R-L, R-C, R-L-C series circuits for Zero input, Step input, pulse input -Initial conditions-solution method using differential equation and Laplace transforms.

Module VI: Single Phase A.C Circuits (3hrs Theory + 3hrs Practice)

Sinusoidal alternating quantities – Phase and Phase difference – Complex and polar forms of representations, J-notation, R.M.S, Average values and form factor for different periodic wave forms - Steady state analysis of R,L and C (in series, parallel and series parallel combinations) with sinusoidal excitation-Concept of Reactance, Impedance Susceptance and Admittance-Power Factor and significance-Real and Reactive power, Complex Power.

Practice:

3. Study of Step Response of R-L Network
4. Study of Step Response of R-L Network

5. Study of Time Response of R-L-C Network.

Module VII: S Domain Analysis (3hrs Theory + 1hr Practice)

Transform Impedance and Transform Circuits, Series and Parallel Combination of Elements, Terminal Pairs, Network Function for the One port and Two port, Poles and Zeros of Network Functions.

Practice:

8. Frequency response of a series and parallel resonant circuit by laboratory set up

Text Books:

4. M. E. VAN VALKENBURG, “Network Analysis”, PHI Publications.
5. A Sudhakar and Shyammhoan S Palli, “Network Analysis”, MC Graw Hill Publishers.

Reference Books

7. Smarajit Ghosh, “Network Theory Analysis & Synthesis”, MC Graw Hill Publishers.
8. B.R.GUPTA, "Network Analysis & Synthesis", S.Chand.
9. BH ferri, "Linear Circuits 2: AC Analysis", Coursera
10. B Tapas Kumar, "Linear Analysis", NPTEL.

Analog Communication Systems

Code	Course Title	Credit	T-P-PJ
CUTM1045	Analog Communication Systems	3	2-1-0

Objective

- Impart the basic concepts of analog modulation schemes.
- Describe different types of noise and predict its effect on various analog communication systems.
- Know the techniques of analog communication and noise analysis in analog communication.

Course outcome

- Analyze energy and power spectral density of the signal.
- Develop an understanding of the performance of analog communication systems.
- Calculate bandwidth and power requirements for analog systems.
- Analyze the different characteristics of the receiver.

Course Outline

Module I (5 Hours) **[3 hrs.Theory + 2 hrs. Practice]**

Basic block diagram of analog communication. Need for modulation, Fourier transform, Properties of Fourier transform: Duality property, Frequency shifting property, Modulation property. Introduction to AM: Time-Domain description, Frequency – Domain description, Generation of AM wave: square law modulator, switching modulator. Detection of AM waves: square law detector, envelop detector.

Practice: (*using Hardware/MATLAB)*

1. DSB+C Modulation using Trainer Kit/ MATLAB
2. DSB+C Demodulation using Trainer Kit/ MATLAB

ModuleII (6 Hours) **[3 hrs.Theory + 3hrs. Practice]**

Double side band suppressed carrier modulation (DSBSC): Time-Domain description, Frequency-Domain representation, Generation of DSBSC waves: balanced modulator, ring modulator. Coherent detection of DSBSC modulated waves.

SINGLE SIDE-BAND MODULATION (SSB):

Quadrature carrier multiplexing, SSB modulation, Frequency-Domain description of SSB wave, Time-Domain description. Phase discrimination method for generating an SSB modulated wave, Demodulation of SSB waves.

Practice: (*using Hardware/MATLAB)*

3. Quadrature Carrier Multiplexing using MATLAB/SCILAB.
4. DSB-SC Modulation and Demodulation using Trainer Kit/ MATLAB.

VESTIGIAL SIDE-BAND MODULATION (VSB):

Frequency – Domain description, Generation of VSB modulated wave, Time – Domain description, Comparison of amplitude modulation techniques, Frequency translation, Frequency division multiplexing, Superheterodyne receiver.

Practice: (*using Hardware/MATLAB).*

5. Frequency Division Multiplexing using Trainer Kit/MATLAB.

Module III (5 Hours) [3 hrs.Theory + 2hrs. Practice]

ANGLE MODULATION (FM):

Basic definitions, FM, narrow band FM, wide band FM, transmission bandwidth of FM waves, Generation of FM waves: indirect FM and direct FM.

Practice: (*using Hardware/MATLAB)*

6. FM Modulation using Trainer Kit/MATLAB/SCILAB

Module IV (5 Hours) [3 hrs. Theory + 2 hrs. Practice]

DEMODULATION OF FREQUENCY MODULATED SIGNALS:

Demodulation of FM waves, FM stereo multiplexing, Phase-locked loop, Non-linear model of the phase – locked loop, Linear model of the phase-locked loop, Nonlinear effects in FM systems.

Practice: (*using Hardware/MATLAB)*

7. FM Demodulation using Trainer Kit/MATLAB/SCILAB.
8. FM Demodulation using PLL.

Module V (4 Hours) [2hrs.Theory + 2hrs. Practice]

RANDOM PROCESS:

Random variables,Statistical averages: Function of random variables, moments, mean, Correlation and Covariance function: Principles of autocorrelation function, cross-correlation functions, Central limit theorem, Properties of Gaussian process.

Practice: (*using Hardware/MATLAB)*

9. Study of Autocorrelation&Cross-correlation using MATLAB

ModuleVI(5 Hours) [3 hrs. Theory + 2 hrs. Practice]

NOISE: Introduction, shot noise, thermal noise, white noise, Noise equivalent bandwidth, Narrow bandwidth, Noise Figure, Equivalent noise temperature, cascade connection of two-port networks, Frequency Domain Representation of Noise, Power Spectral Density, Spectral Components of Noise, Response of a Narrow band filter to noise, Effect of a Filter on the Power spectral density of noise, Superposition of Noises, Noise Bandwidth, Narrow band representation of noise and its PSD.

Practice: (*using Hardware/MATLAB/LABVIEW)*

10. Generation of Gaussian Noise using MATLAB/SCILAB

ModuleVII(3 Hours) [3 hrs. Theory]

NOISE IN CONTINUOUS WAVE MODULATION SYSTEMS:

Introduction, Receiver model, Noise in DSB-SC receivers, Noise in SSB receivers, Noise in AM receivers, Threshold effect, Noise in FM receivers, FM threshold effect, Pre-emphasis and De-emphasis in FM.

- **Text Books:**

1. Communication Systems, Simon Haykins, 5th Edition, John Willey, India Pvt. Ltd, 2009.
2. H. Taub, D. L Schilling, G. Saha,Principles of Communication System, 3rd Edition, 2008, Tata McGraw Hill, India; ISBN: 0070648115. (Selected portions from chapters: 1, 2, 3, 4, 5, 7, 8 and 9)

- **Reference Books:**

1. Modern digital and analog Communication systems B. P. Lathi, Oxford University Press., 4th ed, 2010,
2. Communication System Engineering, Second Edition by MasoudSalehi, John G. Proakis, ISBN: 0130950076
3. Principles of Electronic communication Systems,Louis E. Frenzel,3rd Edition, Tata McGraw Hill.
4. Communication Systems: Analog and digital, Singh and Sapre,TMH, 2nd Ed, 2007.

Digital Communication Systems

Code	Course Title	Credit	T-P-PJ
CUTM1044	Digital Communication Systems	3	2-1-0

Objective

- To impart the fundamentals of modern digital communication system design.
- To evaluate the performance of digital signalling schemes on realistic communication channels.
- Know the techniques of digital communication, information theory, and error control coding.

Course outcome

- Analyze the performance of a baseband and pass band digital communication system in terms of error rate and spectral efficiency.
- Perform the time and frequency domain analysis of the signals in a digital communication system.
- Develop understanding about performance of digital communication systems.
- Calculate bandwidth and power requirements for digital systems.

Course Outline

ModuleI(05 Hours) [03 hrs.Theory + 02hrs. Practice]

Digital Representation of Analog Signal:

Sampling Theorem, Signal Reconstruction from uniform samples,Quantization of Signals, Quantization error, PCM, Electrical representation of binary digits, PCM System, Companding.

Practice: (*using Trainer Kit/MATLAB)*

1. Analysis of Sampling theorem by using Trainer Kit/MATLAB.

2. PCM Modulation and Demodulation using Trainer Kit/ MATLAB.

ModuleII(5 Hours) [03 hrs. Theory + 02hrs. Practice]

Issues in Digital transmission: Line coding, Scrambling, T1Digital System, Multiplexing T1 lines – The T2, T3, T4, E1 lines, Differential PCM: Linear predicted design, Delta Modulation, Adaptive Delta Modulation

(Ref Text Book 1: Chapter 5.1.1, 5.4, 5.5 and 5.6)

Practice: (*usingTrainer Kit/MATLAB)*

3. Differential PCM Modulation and Demodulation using Trainer Kit/ MATLAB

4. Delta Modulation and Demodulation using Trainer Kit/ MATLAB

ModuleIII (6 Hours) [03 hrs. Theory + 03 hrs. Practice]

Digital Modulation Technique:

Generation, Transmission, Reception, Spectrum and Geometrical Representation in the Signal Space of BPSK,QPSK, QASK, M-ary PSK, BFSK, M-ary FSK, and Minimum Shifting Keying (MSK), GMSK, 16-QAM, 64-QAM.

(Ref Text Book 1: Chapter 6)

Practice :(*using MATLAB)*

5. PSK Modulation and Demodulation using Trainer Kit/MATLAB.
6. FSK Modulation and Demodulation using Trainer Kit/MATLAB.
7. ASK Modulation and Demodulation using Trainer Kit/MATLAB.

Module IV (5 Hours) [03 hrs. Theory + 02 hrs. Practice]

Noise in PCM and DM: Calculation of Quantization Noise, Output Signal Power, and the Thermal Noise, Output SNR in PCM, Quantization noise in Delta Modulation, output signal power, output SNR, Comparison with PCM and DM

(Ref Text Book 1: Chapter 12.1 -12.3)

Practice: (*using MATLAB)*

8. SNR study of PCM using MATLAB
9. SNR study of Delta Modulation using MATLAB

Module V (3 Hours) [03 hrs. Theory]

Principle of Digital Data Transmission:

Digital Communication Systems – Source, Line coder, Multiplexer, Regenerative repeater; Line Coding: PSD of various line codes, polar signaling, constructing a DC Null in PSD by pulse shaping, On Off signaling, Bipolar signaling; Pulse shaping – ISI and effect, Nyquist first criterion for zero ISI; Scrambling, Digital receiver and regenerative repeaters; Equalizers, Timing extraction, Detection error, Eye Diagram

(Ref Text Book 2: Chapter 7.1, 7.2, 7.3.1, 7.3.2, 7.4, 7.5, 7.6)

Module VI (4.5 Hours) [03hrs.Theory + 1.5 hrs. Practice]

A base band signal Receiver, Peak signal to RMS noise output voltage ratio, probability of error, optimum threshold, optimum receiver for both base band and pass band: calculation of optimum filter transfer function, optimum filter realization using Matched filter.

(Ref Text Book 1: Chapter 11.1 – 11.3)

Practice:(*using MATLAB)*

10. Study of optimum filters realization using MATLAB.

Module VII (04 Hours) [02 hrs.Theory + 02 hrs. Practice]

Discrete Messages and information content:

Information theory: entropy, mutual information and channel capacity theorem. Shannon coding, Hoffman coding. Fundamentals of error correction, hamming codes.

(Ref Text Book 1: Chapter 13.1 – 13.3)

Practice:(*using MATLAB)*

11. Study of Hamming codesusing MATLAB.

12. Study of error correction using MATLAB.

1. Reference

- **Text Books:**

1. Principles of Communication Systems by H Taub, D. L. Schilling and G Saha, 3rdEdition 2008, TMH Education Pvt. Ltd, New Delhi,

- **Reference Books:**

1. Modern Digital and Analogue Communication Systems by B.P. Lathi and Z Ding, 4th Edition 2010, Oxford University Press, New Delhi
2. Communication Systems by SimonHaykin, 4thEdition, John Wiley & Sons, Inc.
3. Digital Communications, Proakis and Salehi, 5th Edition, Pearson Edu.

- **Online Source:**

1. http://nptel.ac.in/courses/117101051/Digital Communication by Prof.Bikash Kumar Dey, IIT,Bombay, Video Course.
2. https://www.udemy.com/course/digital-communication-information-theory/
3. https://www.udemy.com/course/digital-analog-introduction-to-modulation-in-communication-systems/

Principles of Wireless Communication

Code	Course Title	Credit	T-P-PJ
CUTM1047	Principles of Wireless Communication	3	2-1-0

Objective

- To study the characteristic of wireless channel
- To understand the design of a cellular system
- To study the various digital signaling techniques and multipath mitigation techniques
- To understand the concepts of multiple antenna techniques

Course outcome

Identify and discuss the fundamental operational and design problems of wireless communication systems
Apply basic techniques to design radio links and basic communication systems
Apply basic mathematical and scientific principles to solve engineering design problems
Develop abilities to setup experiments and analyse system performance using wireless systems, hardware and software

Course content

Module 1: Introduction and History of Cellular Communication Systems (3 hours)
Cellular Communication, First to Second Generation (1G to 2G), Third Generation System (3G), Fourth Generation System, (4G), Future Cellular System.

Module 2: Principles of Wireless Communication Theory (4 hours)
How can we represent information into a binary format? How can information be transferred? , How does wireless digital MODEM work? How can high-rate data be delivered reliably? How can many users access simultaneously
Practice:

- Describe the location of where you experienced the largest number of Neighbor Base Stations, and how many were there. In addition, describe the PCI, RSSI, RSRP, RSRQ, RSSNR, ASU, and CQI values measured. (Using Smartphone)

Module 3: Wireless Channels (6 hours)
Large scale path loss – Path loss models: Free Space and Two-Ray models -Link Budget design – Small scale fading- Parameters of mobile multipath channels – Time dispersion parameters- Coherence bandwidth – Doppler spread & Coherence time, fading due to Multipath time delay spread – flat fading – frequency selective fading – Fading due to Doppler spread – fast fading – slow fading.
Practice

- Understanding of Pathloss
- Pathloss with Shadowing
- Horizontal and Vertical Beam Pattern

- Calculation of Boundary Coverage Probability

Module 4: Cellular Architecture (6 hours)

Multiple Access techniques – FDMA, TDMA, CDMA – Capacity calculations–Cellular concept- Frequency reuse – channel assignment- hand off- interference & system capacity Trunking & grade of service – Coverage and capacity improvement.

Practice:

- Frequency Reuse: Co-Channel Cells & Cell Cluster
- Handoff
- Sectoring

Module 5: Digital Signaling for Fading Channels (6 hours)

Structure of a wireless communication link, Principles of Offset-QPSK, p/4-DQPSK,Minimum Shift Keying, Gaussian Minimum Shift Keying, Error performance in fading channels, OFDM principle – Cyclic prefix, Windowing, PAPR.

Practice

- Flat Fading
- Frequency Selective Fading

Module 6: Multipath Mitigation Techniques (5 hours)

Equalization – Adaptive equalization, linear and Non-Linear equalization, Zero forcing and LMS Algorithms. Diversity – Micro and Macro diversity, Diversity combining techniques, Error probability in fading channels with diversity reception, Rake receiver.

Practice

- Calculation of SINR including Beam Tilt: Uplink, Downlink

Module 7: Multiple Antenna Techniques (4 hours)

MIMO systems – spatial multiplexing -System model -Pre-coding – Beam forming –transmitter diversity, receiver diversity- Channel state information-capacity in fading and non-fading channels.

Text Books:

1. Wireless Communications: Principles and Practice, Rappaport, Pearson Education India, 2009.

Reference Books:

2. Principles of Modern Wireless Communication Systems by Aditya K. Jagannatham, Pearson Education India.

Digital Signal & Image Processing

Course Title	Code	Credit	T-P-PJ
CUTM1048	Digital Signal &Image Processing	3	2-1-0

Objective

- To teach students the time domain to frequency domain conversion for discrete time signal and digital filter design techniques
- To provide knowledge on basic concepts of image and its processing techniques
- To provide knowledge on Enhancement, Restoration, Segmentation techniques
- To provide hand on experience of signal & image processing techniques using MATLAB

Course outcome

- Students will gain knowledge on fundamental concepts of a digital signal and image processing System.
- Students will develop skill of developing new algorithms in signal and image processing Applications.
- Student will develop skill on MATLAB implementation of different signal and image processing techniques.

Course Outline

Module I: FUNDAMENTALS OF DIGITAL SIGNAL PROCESSING (2 Hours)

Theory Characterization and classification of signals, Z-Transform: Direct Z-Transform, inverse Z-Transform, Properties of The Z Transform, Linearity, Time Shifting, Scaling, Time Reversal, Differentiation, Convolution, Correlation, Accumulation, System Function of a Linear Time-Invariant System

Practice (1 Hour)

1. Signal generation using MATLAB

2. Analysis of LTI system and Z-transform of signal using MATLAB

Module II: THE DISCRETE FOURIER TRANSFORM & FAST FOURIER TRANSFORM (3 Hours)

Theory: DTFT and DFT Relationship, Discrete Fourier transform (DFT), Properties of the DFT: periodicity, linearity, and symmetry properties, relationship of the DFT to other transforms, DFT as a linear transformation, multiplication of two DFT and circular convolution, Efficient Computation of the DFT, FFT Algorithms: Radix-2 FFT Algorithms: Decimation in-Time (DIT), Decimation-in-Frequency (DIF)

Practice (1 Hour)

3. MATLAB simulation for DFT & IDFT.

4. DIT and DIF FFT by MATLAB simulation

Module III: DESIGN AND REALIZATION OF DIGITAL FIR FILTERS (3 Hours)

Theory: FIR Filter Structure: Direct Form-I, Direct Form-II, Linear Phase FIR Filter, Liner Phase FIR Filter, Design of FIR Filters Using Windowing Techniques, Design of FIR Filter by Frequency Sampling Technique

Practice (2Hours)

5. MATLAB Simulation of FIR filters using windows technique (Rectangular, Hamming and Hanning)

6. MATLAB simulation of LPF and high pass filter by FIR filter

Module IV: DESIGN AND REALIZATION OF DIGITAL IIR FILTERS (2 Hours)

Theory: Design of IIR Filters from Analog Filters(Butterworth Approximation): IIR Filter Design by Impulse Invariance, IIR Filter Design By The Bilinear Transformation, Realization of Digital Filter by using Direct Form-I, Direct Form-II, Cascade Form and Parallel Form Structures.

Practice (2 Hours)

9. Design of IIR Butterworth filter from filter specification (both programming & and by using FDA tool box)

10. Design of IIR low pass Butterworth filter using impulse invariant transformation from filter specification

Module V: DIGITAL IMAGESFUNDAMENTALS

Theory (2 Hours)

Image fundamental, Steps of Image Processing, Types of Images, A simple Image Model, Sampling and Quantization, Pixel Relationship (Neighbor and Adjacency)

Practice (1 Hr 30 Min)

1. Matrix and Array operation in MATLAB
2. Image read and writes operation
3. Reading an RGB Image and extract the color components

Module VI: IMAGEENHANCEMENT

Theory (4 Hours)

Spatial Domain Enhancement, Brightness and Contrast Enhancement, , Basic Gray Level Enhancement-Image Negative, Histogram Equalization, Basic Filtering Operation for Smoothing and Sharpening Filter (Use of Filter Kernel), 2D Fourier Transform and Filtering in Frequency Domain, Ideal Low pass and High Pass Filter for Frequency domain Smoothing and Sharpening.

Practice (3 Hrs)

4. Brightness and contrast enhancement of an image using MATLAB.
5. Simulation of Image negative using MATLAB.
6. MATLAB Simulation of Image smoothing and sharpening using different mask.

Module VII: IMAGE RESTORATION

Theory (4 Hours)

Image Restoration, Model of Image Degradation / Restoration process, Gaussian and Salt and Pepper Noise, Restoration using Mean Filters and Order Statistic Filters (Median and Min-Max Filtering),

Practice

7. MATLAB Simulation of Image noising using different noise distribution.
8. MATLAB Simulation of Image De-noising using Arithmetic mean and median filter.
9. MATLAB Simulation of Image De-noising using Order Statistics Filter (Median, Min-Max Filter).

Reference (Books)

Text Books

1. 1. V. K. Ingle and J.G. Proaksis, J.G, "Digital Signal Processing-A MATLAB Based Approach", Cengage Learning Publisher
2. S. Salivahanan, A. Vallavaraj and C. Gnanapriya, "Digital Signal Processing", McGraw-Hill Publication
3. Gonzalez,RafaelC.,andRichardE.Woods,"DigitalImageProcessing"2ndEdition,Pearson Education, 2002
4. GonzalezRC,WoodsRE,"DigitalImageProcessingAddison-Wesley.Reading",Ma. 1992
5. Sridhar S. Oxford university publication. Digital Image Processing. 2001.

ReferenceBook

Tarun K.Rawat, "Digital Signal Processing", Oxford University Press India

Gonzalez,RafaelC.,andRichardE.Woods,StevenLEddins"DigitalImageProcessing usingMATLAB", Pearson Education, 2009

Antennas Analysis & Design

Code	Course Title	Credit	T-P-PJ
CUTM1049	Antennas Analysis & Design	3	2-1-0

Objective

- To understand the theory and fundamentals of antenna design.
- This course helps the students to learn key aspects of practical antenna design.
- A broad range of antennas such as dipole, loop, microstrip patch, horn, smart etc are studied during the course

Course outcome

- Design and analyze antenna arrays
- Design and analyze wire and aperture antennas
- Identify the characteristics of radio-wave propagation

Course content

Module I: Fundamental Concepts: 8 Hours

Physical concept of radiation, Radiation pattern, near- and far-field regions, reciprocity, directivity and gain, effective aperture, polarization, input impedance, efficiency, Friis transmission equation, radiation integrals and auxiliary potential functions.

Module II: Radiation from Wires and Loops: (7 Hours)

Infinitesimal dipole, finite-length dipole, linear elements near conductors, dipoles for mobile communication, small circular loop. Polarization – Linear, Circular and Elliptical, Radiated Fields, Radiation resistance, Field regions & Directivity, Current distribution, Radiated Fields.

Practice:

Design of Half wave Dipole Antenna

Design of Monopole Antenna

Module III: Module 3: Aperture Antennas:(4 Hours)

Huygens ' Principle, radiation from rectangular and circular apertures, design considerations, Babinet's principle, Radiation from sectoral and pyramidal horns, design concepts.

Practice:

Design of Horn Antenna

Design of Parabolic Antenna

Module IV: Module 4: Broadband Antennas: (3 Hours)

Broadband concept, Log-periodic antennas, frequency independent antennas,

Antennas For Satellite communication.

Practice

Design of Circular antenna Simulation for UWB

Design of Log Periodic Dipole Antenna

Module V: Microstrip Antennas: (3 Hours)

Basic characteristics of microstrip antennas, feeding methods, methods of analysis, design of rectangular and circular patch antennas

Practice

Design of Microstrip Antenna Array Simulation

Design of Microstrip Antenna Simulation

Module VI: Module 6: Antenna Arrays: (6 Hours)

Analysis of uniformly spaced arrays with uniform and non-uniform excitation amplitudes, extension to planar arrays.

Practice

Design of Helix Antenna Simulation

Module VII: Basic Concepts of Smart Antennas: (3 Hours)

Concept and benefits of smart antennas, fixed weight beamforming basics, Adaptive beamforming

Practice

Design of 5G phased array antenna design and beamforming

Text Books:

1. C. A. Balanis, "Antenna Theory and Design", 3rd Ed., John Wiley & Sons., 2005.

2. W. L. Stutzman, and G. A. Thiele, "Antenna Theory and Design", 2nd Ed., John Wiley & Sons., 1998.

3. R. S. Elliot, "Antenna Theory and Design", Revised edition, Wiley-IEEE Press., 2003.

Reference Books:

1. G.S.N. Raju, “Antennas and Wave Propagation”, Person Education.

Source of reference

https://volakis.eng.fiu.edu/teaching

https://www.udemy.com/course/horn-antennas-design-simulation-optimization

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-661-receivers-antennas-and-signals-spring-2003/lecture-notes

https://nptel.ac.in/content/syllabus_pdf/117107035.pdf

Wireless Communications for IIoT

Code	Course Title	Credit	T-P-PJ
CUTM1050	Wireless Communications for IIoT	3	2-0-1

Objective

- To understand the physical aspects of Wireless Communication for IIoT Applications

Course outcome

- Network layers involved in wireless communications
- Protocols used in wireless transmissions
- New Communication tools involved for Industrial applications

Course content

Module 1: Layered Architectures (5 Hours)

Layered Architecture and OSI Model, OSI Unified View of Protocols and Services, TCP/IP: Architecture and Routing Examples Socket API Digital Transmissions: Berkeley Sockets API - I, Berkeley Sockets API – II, Error Control: Error Control - Parity Checks, Error Control – Polynomial Codes (CRC), CRC Capability, Internet Checksum

Module 2: Peer-to-Peer Protocols: 5 Hours

Peer-to-Peer Protocols and Services, Stop-and-Wait ARQ, S& W Performance, and Go-Back-N ARQ, Go-back-N and Selective-Repeat ARQ, Reliable Services and Data Link Controls: TCP Reliable Stream and Flow Control, Framing and PPP, HDLC, Multiplexing.

Module 3: Medium access control: (4 Hours)

Medium Access Control, MAC Random Access: Aloha, Random Access: CSMA and CSMA/CD, Scheduling Approaches

Module 4: Frame Switching and Packet Switching: (4 Hours)

Bridges and Data Link Layer Switching, Network Layer Services and Topology, Packet Switching: Datagrams.

Module 5: Routing in Packet Networks: (4 Hours)

Packet Switching: Virtual Circuits, Routing in Packet Networks, Shortest Path Routing- Distance Vector

Module 6: Traffic Management: (5 Hours)
Packet level – Scheduling and QoS, Packet level – Fair Queueing and RED, Flow level – Leaky, Bucket Policing, Traffic Shaping by Token Bucket

Module 7: Wireless Technologies: (6 Hours)

Wi-Fi , Bluetooth, ZigBee, Lorawan

Case Studies:

- Built a Wireless network using Packet Tracer and analyzing parameters.
- Smart Phone and mobile network Project.
- Bluetooth Scan Project
- Wi-Fi Analysis Project

Text Books:

1. Industrial IoT: Challenges, Design Principles, Applications, and Security By Ismail Butun

Reference Books:

1. Wireless Powered Communication Networks: From Security Challenges to IoT, By Abbas Jamalipour, Ying Bi