The following colour represents the syllabus revision, skill, employability and entrepreneurship.

Green : Skill

Pink : Employability

Sky : Entrepreneurship

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

Learning 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.

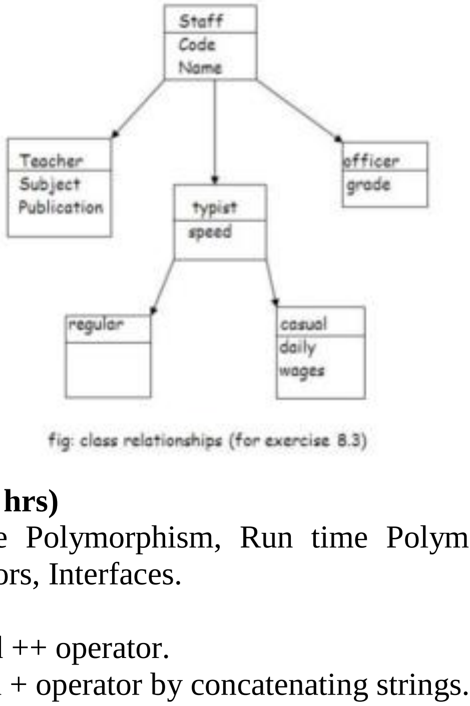


fig: class relationships (for exercise 8.3)

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.

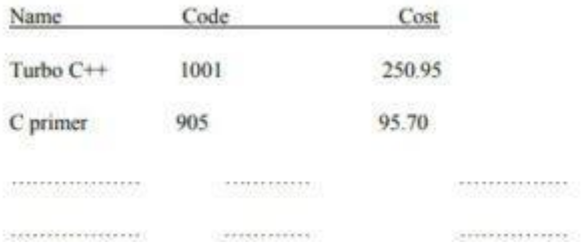
3. Write a program to describe about virtual function.

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

Learning 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

Learning 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 & jQuery(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)

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

Learning 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

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):

```
111111111122222222223333333333444444444455
0123456789012345678901234567890123456789012345678901
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

This course on courseware:http://courseware.cutm.ac.in/courses/java-technologies/

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

Learning 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°, 90°, 180°).
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.

Learning 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

ModuleI: 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.

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.

Learning 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 deliver problem solving skills on circuits through the application of simulation & programming techniques and principles to common circuit problems.
- To analyze the behavior of the circuit’s response in time domain.

Learning outcome

- Apply the knowledge of basic circuital law and simplify the network using different techniques.
- Analyze the circuit using graphical method and network theorems.
- Infer and evaluate transient response, Steady state response
- 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 **(4 hrs)**

Network Topology

Graph of a network, Concept of tree, Incidence matrix, Tie-set matrix, Cut-set matrix, Formulation and solution of network equilibrium equations on loop and node basis

Practice

- Incidence Matrix Formulation
- Tie-set Matrix Formulation
- Cut-set Matrix Formulation

Module II **(5 hrs)**

Network Theorems

Substitution theorem, Reciprocity theorem, Maximum power transfer theorem, Tellegen’s theorem, Millman’s theorem, Compensation theorem

Practice

- Verification of Reciprocity theorem

- Verification of Tellegen’s theorem
- Verification of Milliman’s theorem
- Verification of Maximum power transfer theorem
- Verification of Compensation theorem

Module III
Coupled Circuits **(5 hrs)**

Theory

Coupled Circuits, Dot Convention for representing coupled circuits, Coefficient of coupling, Series and parallel resonant circuits: Band Width and Q-factor

Practice

1. Self-inductance, mutual inductance and coefficient of coupling to be determined for a 1-Ø transformer representing coupled circuit.
2. Frequency response of a series and parallel resonant circuit by laboratory set up.

Module IV **(5 hrs)**
Network Laplace Transform

Application of Laplace transform: Circuit Analysis (Steady State and Transient)

Practice:

- Analysis of transient characteristics using Matlab
- AC and DC transient response analysis for RL,RC and RLC circuits

Module V **(5 hrs)**
Two Port Network

Z, Y, ABCD and h-parameters, Reciprocity and Symmetry, Interrelation of two-port parameters, Interconnection of two-port networks

Practice:

- Determination of Z parameters
- Determination of Y parameters
- Determination of h parameters
- Determination of ABCD parameters

Module VI **(4 hrs)**
Filters

Brief idea about network filters (Low pass, High pass, Band pass and Band elimination) and their frequency response

Practice:

- Design and frequency response analysis of Low Pass filter
- Design and frequency response analysis of High Pass filter
- Design and frequency response analysis of Band Pass filter
- Design and frequency response analysis of Band elimination filter

Module VII
Fourier Series **(5 hrs)**

Theory

Fourier series, Fourier analysis and evaluation of coefficients, Steady state response of network to periodic signals, Fourier transform and convergence, Fourier transform of some functions

Practice:

- Fourier series expansion of Square wave
- Fourier series expansion of Sine wave

Text Books:

1. *M. E. VAN VALKENBURG- Network Analysis, PHI Publications*
2. *A K Chakraborty, "Network Theory," DhanpatRai Publication*
3. *MAHMOOD NAHVI – Electric Circuits, SCHAUM'S Outlines Fifth Edition*

Reference Books:

1. *Smarajit Ghosh- Network Theory Analysis & Synthesis, MC Graw Hill Publishers*
2. *Dr. B.R.GUPTA-Network Analysis & Synthesis, S.Chand*

Energy Production & Transmission

Code	Course Title	Credit	T-P-PJ
CUTM1051	Energy Production & Transmission	3	2-1-0

Course Objective

- To understand power generation and economics
- To design the transmission line parameters
- To understand the mechanical design of transmission lines

Learning outcome

- Able to understand the different functions of major equipment of the power plants and layout designing of the plants
- Able to understand the economic aspects of power system generation
- Able to design transmission line cables

Course content

Module I: Thermal & Nuclear Power Plants **(6Hours)**

Introduction: Statistics of generation of electric power from Conventional and non conventional sources of energy, Thermal & Nuclear power station: Schematic arrangement, Types of prime movers, types of reactors, speed control & auxiliaries, Environmental aspects for selecting the sites and locations, Hazards.

Practice:

1. Schematic Layout design Thermal Plant using AutoCad
2. Schematic Layout design Nuclear Plant using AutoCad

Module II: Hydro & Wind power plants **(4Hours)**

Hydro power station: Schematic arrangement, Hydro turbines, Environmental aspects for selecting the sites and locations of hydro power stations, small hydro for irrigation, Wind power generation.

Practice:

3. Schematic Layout design of Hydro Plant using AutoCad

Module III: Power Generation Tariffs **(5Hours)**

Tariff and Economic aspects in power Generation: Terms commonly used in system operation, various factors affecting cost of generation: Load curves, load duration curves, Connected load, maximum load, Peak load, base load and peak load power plants, load factor, Plant capacity factor, Plant use factor, Demand factor, diversity factor, Cost of power plant.

Practice: MATLAB

4. Preparation of Load calculator using MATLAB

Module IV: Transmission systems **(4 Hours)**

Supply System: Different kinds of supply system and their comparison, choice of transmission voltage.TransmissionLines:Configurations, types of conductors, resistance of line, skin effect, Kelvin’s law, Proximity effect,

Module V: Transmission line Parameters **(5 Hours)**

Calculation of inductance and capacitance of single phase, three phase, single circuit and double circuittransmission lines, Representation and performance of short, medium and long transmission lines, T & Pi networks, ABCD parameters, Ferranti effect, Surgeimpedance loading.

Practice:

5. Designing of transmission line parameters using MATLAB

Module VI: Transmission line operation & Insulators **(5Hours)**

Phenomenon of corona, corona formation, calculation of potential gradient, corona loss, factors affectingcorona, methods of reducing corona and interferenceElectrostatic and electromagnetic interference with communication lines.

Overhead line Insulators:

Type of insulators and their applications, potential distribution over a string of insulators, methods ofequalizing the potential, string efficiency.

Practice:

6. Designing of Insulators and calculation of voltages using MATLAB

Module VII: Design of cables **(4 Hours)**

Calculation of sag & tension, affects of wind and ice loading, sag template, vibrationdampers.Under Ground Insulated cables:Type of cables and their construction, dielectric stress, grading of cables, insulation resistance, capacitance ofsingle phase and three phase cables, dielectric loss, heating of cables, Transmission line tower designs

Practice:

7. Designing of Cables

Text Books:

1. Electrical power Generation, Transmission and Distribution S.N. Singh PHI 2nd Edition, 2009

Reference Books:

1. A Text Book on Power System Engineering A.Chakrabarti, DhanpathRai 2nd Edition

Substation Switch Gear &Protection

Code	Course Title	(Credit)	T-P-PJ
CUTM1052	Substation Switch Gear &Protection	4	2-1-1

Objective

- To understand the different components of substation.
- To understand the protection of different equipment in power system.

Learning outcome

- Able to understand the performance of different protection methods of different equipments.
- Able to understand the different components of substation and their operation.
- Able to design the power system switchgear.

Course content

Module I: Substation Systems **(6 Hours)**

Introduction to Substation System:Definition of substation, necessity of substation, essential features, types of substation, single line diagram of substation, List and functions of each component of substation. Auxiliary systems, Over head earthing screen, Sub-station earthing system.

Practice:

1. Layout Designof 220KV substation using MATLAB
2. Layout Design of 400KV & 750KV substation using MATLAB as per IEEE standards

Module II: Operation & Maintenance of Substations (10 Hours)

Testing and maintenance of Bus Bars, and Isolators:Types and ratings – Bus bar configuration, Tests on Bus bars. Types of isolators and ratings,Load Break switches, Maintenance of isolators,testing and maintenance of Power Transformers, Current and Voltage Transformers and Insulators:Preliminary tests, Final tests, Impulse test, Partial discharge test, Transformer maintenance. Current Transformer tests, Potential Transformer tests. CT and PT maintenance, Tests and maintenance of insulator

Practice:

3. Maintenance tests of CT & PT of substation as per manufacturers Hand book
4. Maintenance tests of Lightening Arrestor & Circuit Breaker of substation as per manufacturers Hand book
5. Maintenance tests of transformer as per manufacturers Hand book

Module III: Protection& System components (5 Hours)

Need for protective schemes, Nature and Cause of Faults, Types of Fault, Effects of Faults, Zones of Protection, Primary and Backup Protection, Essential Qualities of Protection, Classification of Protective Relays, Automatic Reclosing, Current Transformers for protection, Voltage Transformers for Protection.

Module IV: Relays (4 Hours)

Introduction,-Electromechanical Relays,Static Relays – Merits and Demerits of Static Relays, Numerical Relays, Comparison between Electromechanical Relays and Numerical Relays.

Practice:

6. Designing of a Digital Relay

Module V: Relay Operations (7 Hours)

Introduction, Time – current Characteristics, Current Setting, Time Setting, Overcurrent Protective Schemes, Reverse Power or Directional Relay, Protection of Parallel Feeders, Protection of Ring Mains, Earth Fault and Phase Fault Protection, Combined Earth Fault and Phase Fault Protective Scheme, Phase Fault Protective Scheme, Directional Earth Fault Relay, Static Overcurrent Relays, Numerical Overcurrent Relays.

Module VI: Protection control systems (5Hours)

Pilot Relaying Schemes:Introduction, Wire Pilot Protection, Carrier Current Protection
Numerical Differential Relays: Simple Differential Protection, Percentage or Biased Differential Relay, Differential Protection of 3 Phase Circuits, Balanced (Opposed) Voltage Differential Protection.
Rotating Machines Protection:Introduction, Protection of Generators

Transformer and Buszone Protection:Introduction, Transformer Protection, Buszone Protection, Frame Leakage Protection

Practice:

7. Design of Protection Scheme using MATLAB

Module VII: Circuit Breakers **(8 Hours)**

Introduction, Fault Clearing Time of a Circuit Breaker, Arc Voltage, Arc Interruption, Restriking Voltage and Recovery Voltage, Current Chopping, Interruption of Capacitive Current, Classification of Circuit Breakers, Air – Break Circuit Breakers, Oil Circuit Breakers, Air – Blast Circuit Breakers, SF6 Circuit Breakers, Vacuum Circuit Breakers, High Voltage Direct Current Circuit Breakers, Rating of Circuit Breakers, Testing of Circuit Breakers.

Practice:

8. Design & simulation of Circuit breaker using 3DS Tools

Project:

1. Analysis of critically operating power system using MATLAB/DYMOLA

Text Books:

1. Fundamentals of Power System Protection Y.G.Paithankar S.R. Bhide PHI 1 st Edition, 2009

Reference Books:

1. Power System Protection and Switchgear, BhuvaneshOza et al McGraw Hill 1 st Edition, 2010

System Modeling & Control

Code	**Course Title**	**(Credit)**	**T-P-PJ**
CUTM1053	System Modeling and Control	4	3-1-0

Objective

- To teach how to convert a physical systems consist of mechanical and electrical system into a mathematical model.
- Analysis of a live system in time domain and frequency domain and application of

controllers to get the desired response.

Learning outcome

- Students will understand the basics of a system.
- Student gain knowledge on stability of a system.
- Student will analyze the system and controller.
- Students will develop skill of designing automatic control system and controller for a particular application.

Course content

Module I: Introduction **(6 Hours)**

Theory

Introduction to Control Systems: Basic Concepts of Control Systems, Open loop and closed loop systems; Servomechanisms,Laplace transform, Transfer functions, Concept of Pole and Zero.

Practice: Hardware/MATLAB

1. Study of Temperature control system
2. Using MATLAB, find the poles, zeros, gain and draw the pole-zero plot of the transfer function.

Module II: System Dynamics **(10 Hours)**

Theory

Mathematical Models of Physical Systems: Differential Equations of Physical Systems, Mechanical Translational Systems, Rotational systems, Electrical Systems, Analogy between Mechanical and electrical quantities, Derivation of Transfer functions, Block Diagram Algebra, Signal Flow Graphs and Mason’s Gain Formula.

Practice: MATLAB

3. Using MATLAB, find the transfer function from given block diagram.

Module III: Time Response Analysis **(8Hours)**

Theory

Time Response Analysis: Type Test Signals, Time response of first order systems to unit step and unit ramp inputs, Time Response of Second order systems to unit step input, Time Response specifications, Steady State Errors and Static Error Constants of different types of systems.

Practice: MATLAB/DYMOLA

4. Standard Test Signals
5. Time response of first order systems to unit step and unit ramp inputs
6. Time Response of Second order systems to unit step input
7. Using MATLAB, determine the steady state error of the given system.

Module IV: Stability in Time Domain **(4 Hours)**

Theory

Stability in Time Domain: Stability and Algebraic Criteria, concept of stability, Necessary conditions of stability, Hurwitz stability criterion, Routh stability criterion and Application of the Routh stability criterion to linear feedback system

Module V:Root Locus Technique **(5 Hours)**

Theory

Root Locus Technique: Root locus concepts, Rules of Construction of Root locus and Determination of Roots from Root locus for a specified open loop gain.

Practice: MATLAB

8. Construct the root locus for 2nd&3rd order system and analyze its stability (Gain)

Module VI: Frequency Response Analysis **(6Hours)**

Theory

Frequency Response Analysis: Frequency domain specifications, correlation between Time and Frequency Response with respect to second order system, Bode plot, Determination of Gain Margin and Phase Margin from Bode plot.

Practice: MATLAB

9. Construct the bode plot for 2nd and 3rdorder system and analyze its stability (PM & GM)

Module VII: Controllers **(4 Hours)**

Theory

Controllers: Concept of Proportional, Derivative and Integral Control actions, P, PD, PI and PID controllers.

Practice: MATLAB/DYMOLA

10. Design of P,PD, PI and PID Controller for 2ndor 3rdorder system

Text Books:

1. Saeed S. Hasan, "Automatic Control Systems,"Kataria Publication, 9th Edition-2017.

Reference Books:

2. Nagrath J. and Gopal M., "Control Systems Engineering," New Age International Publishers, 6th Edition-2017.

Electrical Machines Operation and Control

Code	Course Title	T-P-PJ	Prerequisite
CUTM1054	Electrical Machines Operation and Control	3-1-0	Basic Electrical Engineering

Objective

- To introduce the students about principles of electromagnetism applied to alternating machines.
- To familiarize the students about the fundamental laws that governs the operation of machines and to extend its application to synchronous generator and motors.
- To introduce the students about the constraints associated with starting of Induction

motors.

- Develop selection skill to identify the type of generators or motors required for particular application.
- Highlight the importance of transformers in transmission and distribution of electric power.

Learning Outcome

- Distinguish the constructional similarity and dissimilarity between various machines.
- Perform different tests on various machines.
- Understand electromagnetic and electromagnetic induction
- Understand DC Machines
- Understand single and three phase A.C circuits, and Understand AC machines

Course content

Module-I: D.C. Machines (5 Hrs)

Theory

Construction, Classification and Principle of operation of DC machines.

Theory & testing:-EMF equation of DC generator, DC Motor Characteristics, Speed Equation of DC Motor. Characteristic for Speed Armature Current, Torque Armature Current and Speed Torque of (i) Separately Excited DC Motor, (ii) DC Shunt Motor, (iii) DC Series Motor, and (iv) DC Compound Motor, Comparison between Different types of DC Motors

Application- DC Generator, DC Motor-Types

Practice:

1) Determination of OCC (Open Circuit Characteristics) of D.C Shunt Generator.

2) Starting & Speed Control of D.C Shunt motor by (i) Field flux control method & (ii) armature voltage control method.

3) Starting & Speed Control of D.C Series motor by (i) Field flux control method & (ii) armature voltage control method.

MODULE II: Stepper Motors (Precision Machines) (6 Hrs)

Theory

Stepper motor drive, basic principles involved in stepper motor control, stepper motor specification, operation and commercial driver chips and packages, Brushless DC Motors, Reluctance Motor, Hysteresis Motor

Application in Medical, Automobile, Civil, Electrical etc

Practice:

4) Motor Voltage and Current Measurement.

5) ON-Load Tap changer

Module-III: Induction Motors (8 Hrs)

Theory

Principles of operation of induction motors, both single and 3-phase types. Torque-speed curves, Different types of single phase motors

Three Phase Induction Motor

Equivalent Circuit and Phasor Diagram, No-Load and Blocked Rotor tests, Determination of Parameters, Slip-Torque Characteristics Losses and Efficiency, Effect of rotor resistance, Starting and speed control methods, Cogging, Crawling and Electrical Braking of Induction Motors.
Applications of three & single phase motors which will assist in picking the right one for an application.

Practice:

5) Determination of parameter of a single phase induction motor and study of (a) Capacitor start induction motor (b) Capacitor start and capacitor run induction motor
6) Determination of Efficiency, Plotting of Torque-Slip Characteristics of Three Phase Induction motor by Brake Test.
7) Load test of a 3 phase slip ring induction motor.

Module-IV: Three Phase Synchronous Generators (7 Hrs)

Theory

Construction, Principle, Coil Pitch, Distributed Windings in A.C. Machines, The Equivalent Circuit of a Synchronous Generator (Armature Reaction Reactance, Synchronous Reactance and Impedance). The Phasor Diagram of a Synchronous Generator, Power and Torque in Synchronous Generators (Power Angle Equation and Power Angle Characteristic)

Practice:

8) Plotting the open circuit and short circuit characteristics of alternator.
9) Calculating the voltage regulation by synchronous impedance method.
10) Calculating the voltage regulation by zero power factor method.

Module-V: Parallel Operation Of Three Phase AC Synchronous Generators (4 Hrs)

Theory

Synchronous condenser, Hunting, Paralleling-Conditions, Procedure, Operation of Generators in Parallel with Infinite bus bar, Effect of excitation, effect of unequal voltage and steam power supply.

Practice:

11) Connection & verifying the conditions of parallel operation of alternators.
12) Verification of direct axis reactance, quadrature axis reactance
13) Load Sharing during parallel operation using Dymola.

Module-VI: Three Phase Synchronous Motors (6 Hrs)

Theory

Basic Principles of Motor operation, Construction, Starting Synchronous Motors, Synchronous Motor Ratings, Equivalent circuit & phasor diagram, Effect of excitation on varying load, power developed in a synchronous motor.
Applications of synchronous motors

Practice:

14) Study of universal motor and shaded pole motor.
15) Use of synchronous motor as a synchronous condenser for p.f improvement.

Module-VII: Single-Phase Transformers (9 Hrs):

Theory

Construction and principle of operation, EMF Equation, Transformation ratio, Practical and Ideal transformers

Three Phase Transformers: Three-phase transformer connections- Star-star, Two Single-Phase Transformers connected in Open Delta (V-Connection) and their rating, Delta-star, Zig-zag connections. Scott connection, Open delta connection. Auto Transformers, Welding Transformer.
Application of Single & Three Phase transformer

Practice:
16) Prescribed tests of single phase and three phase Transformer.
17) Load balancing in a three phase distribution Transformer.
18) Simulation of open delta condition of Transformer.

TEXT BOOK:

1. Electrical Machines – D P Kothari and I J Nagrath, Fourth Edition – Tata McGraw Hill.

REFERENCE BOOKS:

1. Electrical Machinery – P S Bimbhra – Khanna Publishers.
2. Electrical Machines - P. K. Mukherjee, S. Chakravarti, Dhanpat Rai & Sons

Industrial Power Electronics

Code	Course Title	Credit	T-P-PJ
CUTM1055	Industrial Power Electronics	4	2-1-1

Course Objective

- Theymust meet industrial requirement for power electronic engineers.
- They must be gaining adequate practical knowledge on power semiconductor devices, converters and their control techniques.
- They should know the typical applications to motor drives.

Learning outcome

- They will apply their knowledge of the electrical characteristics of power semiconductor devices.
- They will know how to select power semiconductor devices for a range of applications.
- They will understand the basic topology of converters, inverters and power supplies and design calculations for drive
- They will learn the power converter applications, and understand the approximations used.

Course content

Module I **(6hrs.)**

Power Semiconductor Devices

Introduction to power electronics, uncontrolled switches, semi-controlled switches,fully controlled switches, constructional features, operating principle, characteristics and specification of power semiconductor devices,hard and soft switching of power semiconductors switches.

Practice

1. Simulation of V-I characteristics of power diode&power transistor.
2. Simulation of V-I characteristics of MOSFET& IGBT.
3. Simulation of V-I characteristics of silicon-controlled rectifier.

Module II (3 hrs.)

Triggering Circuits

R- Triggering, R-C triggering, UJT triggering, design of UJT triggering circuit.

Practice

4. Simulation of R and RC triggering.
5. Simulation of UJT triggering

Module III (8 hrs)

AC to DC Converter

Overview of rectifiers, half wave uncontrolled rectifier with R load and R-L load, use of freewheeling diode, half wave rectifier R-L load with FWD, full wave bridge uncontrolled rectifier, half wave controlled rectifier with R load,R-L load and R-L load with free-wheeling diode, half controlled bridge rectifier, fully controlled bridge rectifier, effect of source inductance on the performance of ac to dc converters, power factor improvement,harmonic reduction, filter design.

Practice

6. Simulation of single phase half-wave and full-wave diode rectifier using R & L load.
7. Simulation of single phase fully controlled converter using R & L load.
8. Simulation of single phase semi converter using R-L load.
9. Simulation of 3-phase semi converter with R, R-L and dc motor load with/without freewheeling diode.
10. Simulation of 3-phase bridge converter with R, R-L and dc motor load with/without freewheeling diode.

Module IV: (10 hrs.)

DC to DC Converter

Introduction to chopper (Type A,B, C, D, E), switching techniques, step down dcchopper with R load, R-L-E load, step up dc chopper with R, R-L, R-L-E load, buck regulator, boost regulator, Buck-boost regulator, CUK and SEPIC converter, commutation of thyristor based circuits part-I, commutation of thyristor based circuits part-II, introduction to SMPS circuits, fly back type SMPS, forward type SMPS, design of transformer for SMPS circuits.

Practice

11. Simulation of buck converter.
12. Simulation of boost converter.
13. Simulation of buck boost converter.

Module V (6 hrs.)

DC to AC Converter

Introduction to inverters, importance and application of inverters, single phase half bridge inverter with R and R-L load, single phase bridge inverter with R and R-L load, three phase inverters, control techniques of inverter, single/multiple pulse width modulation, sinusoidal pulse width modulation and its realization, CSI, load-commutated CSI, industrial inverter.

Practice

14. Simulation of single-phase inverter&three phase inverter.

Module VI (3 hrs.)

AC to AC Converter

AC voltage controller: Single phase bi-directional controllers with R and R-L load, single phase cyclo-converters.

Practice

15. Simulation of single phase AC voltage controller&cyclo-converter.

Module VII (9 hrs)

Application of Power Electronics Converters

Analysis of converter fed dc drives, analysis of chopper fed dcdrives,analysis of VSI,CSI fed induction motor drives, automotive & traction system, industries as rolling mills, pumps, elevators, utility systems as FACTs, smart grid, and renewable energy as wind turbine.

Practice

16.Simulation of converter fed dc drives (Wind Turbines).

17. Simulation ofchopper fed dcdrives (PV Systems).

18. Simulation of induction motor drives (e-Vehicle).

19.Simulation of railway electrification system using Dymola

Project

1. 500 VA Sine wave Inverter
2. Industrial Battery Charger using SCR
3. Precision Illumination control of Lamp
4. Dual Converter using Thyristors

Text Books:

1. *M. H. Rashid,"Power Electronics: Circuits, Devices and Applications," 4th Edition, Pearson,2017*
2. *M. D. Singh & K. B. Khanchandani, "Power electronics", 2ndnd Edition, Tata McGraw-Hill,2008*

Reference Books:

1. *J. Vithayathil,"Power Electronics: Principles and Applications",2nd Edition*

TMH Edition,1995

2. *Mohan, Undeland and Robbins,"Power Electronics: Converters, Applications and Design" 3rd Edition Edition, 2007*

Digital Measurement and Instrumentation

Code	Course Title	Credit	T-P-PJ
CUTM1056	Digital Measurement and Instrumentation	3	2-1-0

Objective

- The main objective of this course is to explain the operation, performance and application of Digital Measuring Instruments to the students.

Learning Outcome

- Understand the construction, principle and characteristics of different types of digital measuring instruments
- Apply the knowledge about different instruments and can identify the best suitable instrument for a required typical measurement

Course Content

Module I **(6 hrs)**

Philosophy of digital measurements

Time Measurement Techniques: Error analysis in digital measurement, Measurement of time interval between two events, Error in time interval measurement, Vernier technique for small time measurement, Measurement of time interval with constraints, Measurement of periodic time, phase, Quality factor of ringing circuit, Decibel meter, Software controlled measurement.

Practice

- Error analysis of digital measurement using Matlab
- Simulation of Quality Factor of ringing circuit

Module II **(5 hrs)**

Digital frequency measurement techniques

Measurement of frequency, Ratio of two frequencies, Product of two frequencies, High frequency, average Frequency difference, Deviation of power frequency, Peak frequency. Fast low-frequency measurement, Digital Tachometer.

Practice

- Addition and product of different frequencies using Matlab
- Simulation of digital tachometer using Matlab

Module III **(5 hrs)**

Digitally Programmable Circuits

Single mode switching, Group mode switching, Resistors, Potentiometers, Amplifiers, Schmitt trigger, Dual polarity gain amplifiers. Programmable gain amplifier with dual output, Two stage programming, Programmable Biquads.

Practice:

- Analysis of switching using Matlab
- Simulation of programmable biquads using Matlab

Module IV **(4 hrs)**

Digital to Analog Converters

Output Input relation, DACs derived from programmable gain amplifiers, Weighted-resistor DAC, Weighted current DAC, Weighted reference voltage DAC, Ladder DAC, Switches.

Practice:

- Simulation of programmable gain amplifier using Matlab
- Simulation of DAC using Matlab

Module V (5 hrs)

Digital Voltage Measurement Techniques

Sampling theorem, Time-division multiplexing, Quantization, Indirect type A/D converters, Direct type A/D converters, Input circuitry of a digital voltmeter.

Practice:

- Simulation of Digital voltmeter using Matlab
- Analysis and simulation of digital multi-meter.

Module VI (4 hrs)

Digital Instrument

Need for digital instruments, Advantages of digital instruments, Essentials of digital instrument, Performance characteristics of digital instrument.

Digital Recording Systems

Input Conditioning Equipment, Digitizer, Multiplexer, Programme Pinboard, Linearizer, Digital Clock, Limit Detectors, Output Devices

Practice:

- Data fetching using controllers
- Simulation of digital clock using Matlab

Module VII: (4 hrs)

Signal Generator, Analyzersand Oscilloscopes:

Function Generator, Pulse Generator, RF Signal Generator, Harmonic Distortion Analyzer, Spectrum Analyzer, Digital Storage CROs

Practice:

- To study block wise Construction of a Function Generator
- Measure Voltage, Frequency, Phase and Modulation Index (Trapezoidal Method) using CRO
- Demonstrate features of Digital Storage Oscilloscope
- Measure unknown Frequency using Lissajous Patterns

Text Books:

1. T. S. Rathore- Digital Measurement Techniques, Alpha Science International Ltd
2. David A. Bell - Electronic Instrumentation and Measurements, Oxford Univ. Press, 1997
3. A. K. Sawhney – A Course in Electrical and Electronic Measurements and Instrumentation, Dhanpat Rai & Co

Reference Books:

1. PrithwirajPurkait- Electrical and Electronics Measurement and Instrumentation, MC Graw Hill Publishers
2. H.S. Kalsi-Electronic Instrumentation, Tata McGraw-Hill, New Delhi, 2010
3. R. K. Rajput- Electrical & Electronic measurement and Instrumentation, S. Chand Publication

4. K. Lal Kishore- Electronic Measurements and Instrumentation, Pearson Education 2010

Basic Electrical Engineering

Code	Course Title	(Credit)	T-P-PJ
CUTM1057	Basic Electrical Engineering	2	1-1-0

Objective

- In this course, student will come to know about the Basics of Electrical Engineering, Currents and Voltages across various Electrical elements.
- Their behavior in both Alternating Current and Direct Current circuits.
- Analysis of 1-phase and 3-phase AC wave forms.

Learning outcome

- Student will be exposed to the breadth of electrical engineering, able to gain knowledge in Electrical Circuits (AC and DC).
- Acquire knowledge on various parameters of electrical engineering and their properties with hands-on-practice of basic electrical experiments.

Course content

Module I: Basic Concepts and Basic Laws (4hrs)

Theory

Essence of Electricity, Electric Field; Electric Current, Potential and Potential Difference, E.M.F., Electric Power, Ohm's Law, Basic Circuit Components, Ideal and Practical Sources, Source Conversion.

Practice:

1. Design and Analysis of Basic electrical circuits using Dymola. Plotting the V-I Characteristics of Incandescent lamp using Dymola.

Module II: Methods of Analysis (4hrs)

Theory

Network Analysis using Series and Parallel Equivalents, Voltage and Current Divider Circuits, Nodal Analysis, Mesh Analysis, Delta-Star & Star-Delta conversion.

Practice :

2. Verification of KCL and KVL in series and parallel circuits using Dymola.

Module III: DC Network Theorems (3hrs)

Theory

Analysis of Superposition, Thevenin's and Norton's theorem.

Practice:

3. Verification of Superposition, Thevenin's and Norton's theorem using Dymola.

Module IV: Introduction to Electromagnetism (4hrs)

Theory

Magnetic Circuits, B-H curve, Permeability, Reluctance, Solution of simple magnetic circuits, Hysteresis and Eddy current loss. Methods of preventing such losses. Solenoids and field coils. Application of solenoids in different circuits in Automobiles and in electrical circuit.

Practice (Hardware):

4. Observation of generation of magnetic flux for different input current in a coil and plotting B-H Curve.

Module V: Single-Phase Transformer (2hrs)

Practice (Hardware):

5. Study of Transformers, Linear Transformer Model, Ideal Transformer Model, No-load Loss and Load-loss Calculation.

Module VI: AC Circuit Analysis (3hrs)

Theory

Single-phase EMF Generation, Waveform and Phasor Representation, Average and Effective value of sinusoids, Peak factor & Form factor, Complex Impedance and Power using j-operator, Power factor.

Practice:

6. Calculation of current, voltage, power & power factor of series RLC circuit excited by 1-Ø A.C Supply using Dymola.

Module VII: Phasor Analysis (3hrs)

Theory

Three-Phase AC Circuits: Comparison between single-phase and three-phase systems, Three-phase EMF Generation, Line and Phase quantities in star and delta networks, Power and its measurement in three-phase balanced circuits.

Practice

7. Measurement of power and power factor in a 3-Ø AC circuit by (one, two and three) wattmeter using Dymola.

Recommended Books:

1. P. K. Sathpathy, "Basic Electrical Engineering," 3rd Edition, Oxford.
2. B. L. Thereja, "Electrical Technology", Volume-I,2005 Edition(24th Revised Edition)
3. Hughes, "Electrical & Electronic Technology", Ninth Edition (Revised by J Hiley, K Brown, and I Smith), Pearson Education