UNLIMITED POWER FULL LEARNING

UNLIMITED POWERFUL LEARNING

UPL PROSPECTOUS

early break even | best experience

@unlimited powerfull learning

2023

ABOUT COMPANY:

UNLIMITED POWER FULL LEARNING (UPL) aims to solve the challenges and minimize the gap between students with IT industries' expectations. This organization is built by a strong team who are having good academic and industry experience of more than two decades. The founder of this experience G.D. Mallikarjuna has 20+ plus started as a technologist having diverse experience in the education sector as Trainer and Developer.

VISION:

At UPL@SNIPE, we make the best experience in technology learning with career guidance for their life journey

MISSION:

Learn with Live experience and career values.

PROGRAMS OFFERED:

PROGRAMS	DURATION	AMOUNT + GST
CODING BOOT CAMP	4 TO 6 MONTHS	Rs.30000/-
CERTIFICATION COURSE	3 SEMESTERS 1 YEAR COURSE	Rs. 25000/- per semester Rs. 10000/- final semester
CAREER BRIDGE	3 MONTHS	Rs. 50000/-
INDUSTRY READINESS PROGRAM	3 MONTHS	Rs.20000/-

*Terms and Conditions are subjected to further discussion before signing the legal agreement

CODING BOOT CAMP

ABOUT THIS MODEL

- **Category:** Virtual Program
- **Target Audience:** Fresher & Experienced
- **Duration:** 4 To 6 Months
- Cost: Rs. 30,000/Candidate (**Registration:** 10K + GST **After 6 Weeks:** 10K + GST **Live Project:** 10K + GST)
- **Course Coverage:** 2 Months training in a relevant discipline, 1 capstone project & followed by involving in live project for duration 4 months.
- **Outcome:** Build their careers feature strong growth projections & lucrative salaries
- **Career Opportunities:** The best jobs you can secure after completing one of these programs such as, Technical Support Specialist, Digital Marketer, Junior Developer, Data Analyst, Web Developer, Project Manager, User Interface/Xxperience (UI/UX) designer, Application Developer, Product Manager, Software Engineer, Full Stack Developer, Data Scientist, Development Operations (DevOps) Engineer, Back End Eeveloper, Teach Others, also Freelancer

COURSES ARE :

- JAVA FULLSTACK
- FULL STACK C# .NET
- FRONT END DEVELOPER IN (REACT/ANGULAR)
- MEAN STACK
- PYTHON
- DATA-SCIENCE
- AUTOMATION TESTING WITH JAVA
- UI/UX DESIGN
- DIGITAL MARKETING
- JENKINS
- MACHINE LEARNING
- DATASTRUCTURE IN PYTHON
- TABLEOU
- POWER BI
- PSPARK
- DEVOPS

BENEFITS IN THIS PROGRAM :

- Uplsnipe Coding Bootcamp Certificate.
- Program Transcript For The Entire Learning Path.
- Coding Bootcamps Can Open Doors To Exciting Technical Career Opportunities.
- Mastering Programming Languages And Associated Technologies Can Prepare You To Work As A Software Or Web Developer.
- Strong Growth Projections And Lucrative Salaries

FULL STACK C# .NET

A Full-Stack C# .NET Course Typically Covers A Wide Range Of Topics, From Front-End Web Development To Back-End Server-Side Programming, Database Management, And More. Here's An Overview Of The Typical Content Covered In A Full-Stack C# .NET Course:

UNIT_001 : C# PROGRAMMING FUNDAMENTALS: 03 HRS

what is Project? Difference between project, product, platform, technology, front end, backend and architecture, Introduction, User Interface Overview, Design principles, UI, JavaScript, JQuery, AngularJS, Node js, standards, Design

UNIT_002 : INTRODUCTION TO UI TECHNOLOGY

05 HRS

Introduction To Front-End Web Development Technologies.
HTML Markup, CSS Styling, And Javascript Programming Fundamentals.Building Responsive And Mobile-First Web Interfaces Using Bootstrap Or Other UI Frameworks.

FRONT END TECHNOLOGIES

UNIT_003 : HTML

35 HRS

IINTRODUCTION,History,Text-editor,Webbrowser, Html page structure, Html document format, Basic tags list, Html elements and tags,, Types of tags, Html attributes

UNIT_004 : CSS

Introduction, Selector, Style, Background, Border, Display, CSS Float, Font, Text, Button, Table, Form

UNIT_005 : BOOTSTRAP

History,Why bootstrap,Getting started,containers,Grid basic,Typography Colours,Tables,Images,Alerts,Buttons,Progressbar,Spinner,Pagination,List groups,Cards,Dropdowns,Collapse,Navs,Carousel,Modal,Tooltip,Popovers, Toast,Scrollspy,Offcanvas,Utilities,Bootstrap 5 forms

UNIT_006 : JAVA SCRIPT

Introduction,History,Features,Browser,Browser Object Model (BOM),How does 03 Hrs Java script works?,Document Object Model (DOM) ,Data type, undefined, falsy values, conditional and control statements,with statement,looping constructs,string, Arrays,window, forms, events,DHTML

UNIT_007 : JQUERY

JQUERY HOW TO USE JQUERY?, Merits,methods,Introduction,History,What Is Typescript,Why We Use,Features,Ts & Ecmasccript,Components Of Ts,Declaration Files,Environment Setup,Basic Syntax,Compile & Exe ,Example,Classes,Modules

LAB SET FOR HTML ,CSS,BOOTSTRAP,AND JAVA SCRIPT

LAB 1 : HTML BASICS

- Create an HTML page with a header, footer, and navigation menu.
- Add different types of HTML elements such as headings, paragraphs, lists, and images.
- Create a simple form with input fields and a submit button.

LAB 2 : CSS STYLING

- Style the HTML elements using inline CSS, internal CSS, and external CSS.
- Apply different font styles, colors, and backgrounds to elements.
- Create a CSS-based layout for a web page using divs and CSS properties like float and margin.

LAB 3 : RESPONSIVE DESIGN WITH BOOTSTRAP

- Integrate Bootstrap into an HTML page using the Bootstrap CDN.
- Use Bootstrap classes to create responsive layouts and grids.
- Apply Bootstrap components such as navigation bars, buttons, and forms.

LAB 4 : CSS FLEXBOX AND GRID

- Use CSS Flexbox to create a flexible layout for a web page.
- Create a grid layout using CSS Grid to arrange elements in a structured manner.
- Combine Flexbox and Grid to create complex layouts.

LAB 5 : JAVASCRIPT BASICS

- Write JavaScript code to display a popup message or alert on a web page.
- Create a JavaScript function to validate a form before submitting it.
- Manipulate HTML elements dynamically using JavaScript.

LAB 6 : DOM MANIPULATION

- Use JavaScript to access and manipulate the Document Object Model (DOM) of a web page.
- Create interactive features like event listeners and handlers using JavaScript.
- Build a simple slideshow or image carousel using JavaScript.

LAB 7 : FORM VALIDATION WITH JAVASCRIPT

- Write JavaScript code to validate form input fields, such as checking for empty fields or valid email addresses.
- Display error messages dynamically based on user input using JavaScript.
- Implement client-side form validation before submitting data to the server.

LAB 8 : HANDLING EVENTS WITH JAVASCRIPT

- Write JavaScript code to handle different types of events, such as button clicks or mouse movements.
- Implement functionality to show and hide elements based on user actions using JavaScript event handlers.
- Use JavaScript to create interactive elements like dropdown menus or to

LAB 9 : AJAX AND API INTEGRATION

- Use JavaScript and AJAX to make asynchronous requests to a server and update web page content dynamically.
- Integrate an external API into a web page to fetch and display data using JavaScript.
- Implement functionality to search and filter data on a web page using AJAX and JavaScript.

LAB 10 : MINI PROJECT

Design and implement a simple web application that combines HTML, CSS, Bootstrap, and JavaScript.

ANGULAR

20 HRS

UNIT_008 : INTRODUCTION TO ANGULAR:

Understanding the fundamentals of Angular.
Overview of Angular architecture and its key features.
Setting up the development environment for Angular.

UNIT_009 : TYPESCRIPT:

Introduction to TypeScript, the language used in Angular development.TypeScript syntax, data types, variables, and functions.
Classes, interfaces, and modules in TypeScript.
TypeScript decorators and their usage in Angular.

UNIT_010 : ANGULAR COMPONENTS:

Understanding Angular components and their role in building applications.
Creating components using the Angular CLI (Command Line Interface).
Data binding and event handling in components.
Component lifecycle hooks and their usage.

UNIT_011 : ANGULAR TEMPLATES AND DIRECTIVES:

Angular template syntax and interpolation.
Working with built-in directives (e.g., ngIf, ngFor, ngSwitch).
Creating custom directives in Angular.

UNIT_012 : ANGULAR SERVICES AND DEPENDENCY INJECTION:

Creating and using services in Angular.
Understanding dependency injection and its importance.
Injecting services into components and other Angular constructs.
Sharing data and functionality across components using services.

UNIT_013 : ROUTING AND NAVIGATION:

Implementing client-side routing in Angular.
Configuring routes and route parameters.
Implementing nested routes and child components.
Using router guards for authentication and authorization.

UNIT_014 : ANGULAR FORMS:

Building forms in Angular using template-driven and reactive approaches.
Form validation and error handling.
Working with form controls and form groups.
Implementing custom form validators.

UNIT_015 : ANGULAR MODULES AND LAZY LOADING:

Understanding Angular modules and their role in organizing the application.
Creating and configuring modules in Angular.
Implementing lazy loading for optimizing application performance.

UNIT_016 : ANGULAR HTTP AND API INTEGRATION:

Making HTTP requests using the Angular HttpClient module.
Consuming RESTful APIs and handling responses.
Error handling and interceptors in HTTP requests.
Mocking HTTP requests for testing purposes.

UNIT_017 : ANGULAR TESTING AND DEBUGGING:

Writing unit tests for Angular components, services, and directives.
Using testing utilities and frameworks (e.g., Jasmine, Karma).
Debugging Angular applications using browser developer tools.
Performance optimization and debugging techniques.

UNIT_018 : ANGULAR DEPLOYMENT AND BUILD OPTIMIZATION:

Building and bundling Angular applications for production.
Optimizing application size and performance.
Deploying Angular applications to various hosting platforms.
Continuous integration and deployment (CI/CD) for Angular projects.

UNIT_019 : DEVOPS AND DEPLOYMENT:

Understanding the principles and practices of DevOps and continuous integration/continuous deployment (CI/CD).
Setting up build and release pipelines using tools like Visual Studio, Jenkins, or Azure DevOps.
Deploying applications to cloud platforms like Azure or AWS

UNIT_020 : TESTING AND QUALITY ASSURANCE:

Understanding the importance of software testing and quality assurance.
Introduction to automated testing frameworks like NUnit, xUnit, or MSTest.
Writing unit tests and integration tests to ensure code quality and maintainability.

UNIT_021 : PROJECT MANAGEMENT AND TEAM COLLABORATION

Introduction to agile project management methodologies like Scrum or Kanban.
Effective collaboration and communication within a development team.
Source control and versioning using Git and other collaboration tools.

LAB SET ANGULAR

LAB 1 : SETTING UP ANGULAR DEVELOPMENT ENVIRONMENT

- Install Node.js and Angular CLI.
- Create a new Angular project using Angular CLI.
- Run and test the default Angular application in the browser.

LAB 2 : COMPONENTS AND TEMPLATES

- Create a new component and template.
- Use data binding to display dynamic content in the template.
- Implement event binding to handle user interactions.

LAB 3 : SERVICES AND DEPENDENCY INJECTION

- Create a service to handle data retrieval and manipulation.
- Inject the service into a component and use it to fetch and display data.
- Implement dependency injection to provide the service to the component.

LAB 4 : ROUTING AND NAVIGATION

- Configure routing in the Angular application.
- Create multiple components and associate them with different routes.
- Implement navigation between components using router links and programmatically.

LAB 5 : FORMS AND VALIDATION

- Create a form with input fields and submit functionality.
- Implement form validation using Angular's built-in validators.
- Display validation errors and provide user feedback.

LAB 6 : HTTP REQUESTS AND APIS

- Use Angular's HttpClient module to make HTTP requests to an API.
- Fetch data from an external API and display it in the application.
- Implement CRUD operations (create, read, update, delete) using HTTP requests.

LAB 7 : ANGULAR MATERIAL

- Install and configure Angular Material in the project.
- Use Angular Material components like buttons, cards, and forms.
- Apply styles and themes provided by Angular Material.

LAB 8 : STATE MANAGEMENT WITH NGRX

- Set up NgRx for state management in the Angular application.
- Create actions, reducers, and effects to manage application state.
- Use selectors to retrieve and display data from the store.

LAB 9 : AUTHENTICATION AND AUTHORIZATION

- Implement user authentication using a mock or real authentication service.
- Protect routes and components based on user authentication status.
- Restrict access to certain features based on user roles or permissions.

LAB 10 : DEPLOYMENT AND PRODUCTION BUILD

- Build the Angular application for production.
- Deploy the application to a hosting platform or server.
- Configure routing and server-side rendering for SEO optimization.

BACKEND TECHNOLOGY

ASP.NET CORE:

40 HRS

UNIT_022 : INTRODUCTION

Introduction, What is ASP.NET Core?,ASP.NET Core Features , Advantages of ASP.NET Core MVC Pattern, Understanding ASP.NET Core MVC, SP.NET Core vs. ASP.NET MVC vs. ASP.NET Web Forms, ASP.NET Core Environment Setup, ASP .NET Core First Application, project Layout, Understanding Life Cycle of ASP.Net Core Request. Controllers & Action Methods : controllers Overview, Action Methods, and IActionResult object Passing data from Controller to View, Understanding Action Selectors, Action Filters, Building Custom Action Filters, Middleware, Asynchronous Action Methods, Views : Introducing Razor View , Advantages of Razor View ,Razor Syntax , Types of Views, Partial Views, Layout Pages , Special Views , View Categorization based on Model,Validations &

UNIT_022 : DATA ANNOTATIONS :

Data Annotations and Validations Overview , Validations with Data Annotation, Server Side and Client Side Validation, Custom Server side validation, Custom unobstrive Client side Validation, Remote Validation.

UNIT_023 :

ASP.NET MVC : INTRODUCTION TO WEB PROGRAMMING:

Communication between Web , Server and Web Browser, HTTP Protocol , HTTP Request Life Cycle , Request and Response Structure, About Get and Post methods, Creating an ASP.NET Website, Using ASP.NET WebForm Controls , ASP.NET Architecture, Understanding Page Controller, Architecture, Navigating between Pages, Overview of Master Pages, What is MVC Architecture? , What is ASP.NET MVC?, Understanding Model, Understanding View Understanding Controller, Advantages of MVC based Web Application , Features of ASP.NET MVC Framework,

UNIT_024 : EXPLORING CONTROLLER'S :

Exploring Controllers and Controller Base class Passing data from Controller to View using View Data/View Bag , Types of Action Methods, Action Method Parameters, Action Selectors ,Action Filters Overview, Authentication Filters in MVC 5, Building Custom Action Filters , Filter Override features, Asynchronous Controllers

UNIT_025 : EXPLORING RAZOR VIEWS:

Types of Views, Introducing Razor View, Razor Syntax Fundamentals ,Enum Support, Layout view Razor (Master Pages) ,Significance of _ViewStart.cshtml,Working with Sections, Working with Partial Views ,Bootstrap support for editor templates

UNIT_026 : ANNOTATIONS AND VALIDATIONS:

Overview of Data Annotations, Annotations and Validation Attributes, How Validation Works ,Explicit server side validations of Models ,Custom Validations using IValidatableObject ,Developing Custom Unobtrusive Client Side Validation ,Applying Annotations to Model classes using Metadata class

UNIT_027 : WEB CONFIGURATION FILE AND GLOBAL APPLICATION CLASS:

Architecture of Config File ,Machine.config and Web.Config ,AppSettings , system. Web Section, Exception Handling ,Location Section ,Encrypting Connection String

UNIT_028 : STATE MANAGEMENT TECHNIQUES :

Cookies , Authentication in Cookies , Cookie Dictionary , Sessions Management

UNIT_029 : CRUD OPERATIONS USING ENTITY FRAMEWORK :

Basic CRUD Operations using Scaffold Templates , Separation of work using BO Classes Using Single Database Context Object across all Business Objects ,Caching in Repository

UNIT_030 : AUTHENTICATION AND AUTHORIZATION :

Overview of Authentication and Authorization , Types of Authentication , [Authorize] and [AllowAnonymous] attributes, Windows Authentication Implementation, Forms Authentication Implementation, Roles Based Security

UNIT_031 : ASP.NET IDENTITIES:

ASP.NET Identity Introduction, OWIN, and Katana , Customizing Template Generated Code, Extending Identity Model and using Integer Key instead of String Key, OAuth, and Social Authentication, Implementing Email Confirmation , Two Factor Authentication , User and Role Management

LAB SET ASP.NET CORE, ASP.NET MVC

LAB 1 :

Install the .NET SDK: Begin by installing the .NET SDK, which includes the necessary tools and libraries for developing and running ASP.NET Core applications. You can download the SDK from the official Microsoft website (https://dotnet.microsoft.com/download).

LAB 2 :

Choose an Integrated Development Environment (IDE): There are multiple IDE options for ASP.NET Core development. Visual Studio, both the full version and the free Community edition, provide a comprehensive development environment. Visual Studio Code, a lightweight and extensible editor, is another popular choice. Install your preferred IDE.

LAB 3 :

Create a New ASP.NET Core Project: Open your IDE and create a new ASP.NET Core project. You can choose from various project templates, such as MVC (Model-View-Controller), Web API, Razor Pages, or Blazor, depending on your application needs.

LAB 4 :

Configure Version Control: Initialize a version control repository for your project using Git. This allows you to track changes, collaborate with others, and manage different versions of your code. You can use the Git command-line interface or IDE-integrated Git tools.

LAB 5 :

Set Up a Database: If your application requires a database, you'll need to install and configure a database server. Microsoft SQL Server is a popular choice, but you can also use other options like MySQL or PostgreSQL. Configure your database connection string in your ASP.NET Core project.

LAB 6 :

Install NuGet Packages: NuGet is the package manager for .NET, and it allows you to add third-party libraries and dependencies to your project. Use the NuGet Package Manager Console or the IDE's integrated NuGet package manager to search for and install the required packages.

LAB 7 :

Develop and Test Your Application: Start writing your ASP.NET Core application code, including controllers, models, views, or APIs. Utilize the features and frameworks provided by ASP.NET Core, such as Entity Framework Core for database access, authentication middleware, or dependency injection.

LAB 8 :

Set Up Testing and Continuous Integration: Implement unit tests for your code using testing frameworks like xUnit, NUnit, or MSTest. Set up a continuous integration (CI) pipeline using tools like Azure DevOps, Jenkins, or GitHub Actions to automatically build and test your application on code commits.

LAB 9 :

Deploy and Host Your Application: Choose a hosting provider or deployment strategy for your ASP.NET Core application. Options include Azure App Service, Azure Kubernetes Service (AKS), or self-hosting on IIS or Docker containers. Configure the necessary deployment settings and publish your application.

LAB 10 :

Monitor and Debug: Implement logging and monitoring in your ASP.NET Core application using frameworks like Serilog or Application Insights. Set up debugging configurations in your IDE to help troubleshoot issues during development.

UNIT_028 : STATE MANAGEMENT TECHNIQUES :

Cookies , Authentication in Cookies , Cookie Dictionary , Sessions Management

UNIT_029 : CRUD OPERATIONS USING ENTITY FRAMEWORK

Basic CRUD Operations using Scaffold Templates , Separation of work using BO Classes Using Single Database Context Object across all Business Objects ,Caching in Repository

UNIT_030 : AUTHENTICATION AND AUTHORIZATION :

Overview of Authentication and Authorization , Types of Authentication , [Authorize] and [AllowAnonymous] attributes, Windows Authentication Implementation, Forms Authentication Implementation, Roles Based Security

UNIT_031 : ASP.NET IDENTITIES:

ASP.NET Identity Introduction, OWIN, and Katana , Customizing Template Generated Code, Extending Identity Model and using Integer Key instead of String Key, OAuth, and Social Authentication, Implementing Email Confirmation , Two Factor Authentication , User and Role Management

C# PROGRAMMING:

UNIT_032 : C# LANGUAGE SYNTAX : 05 HRS

Why do we need a programming Language, What are the Data Types we have in C# and How to declare a Variable, How Data Types are Categorized into Value Type and Reference Type What is Implicit Casting and Explicit casting and How to handle Overflow checks , Difference between string and string Builder ,What is Boxing ,What is Unboxing ,What is Type Inference , What are constants and Enums ,What are the Operators we have in C#, How the if, while, do while, switch condition will works

UNIT_033 : OOPS – CONCEPTS: 05 HRS

Introduction to OOPS and its principles, What is a class, What is an object , What is a component ,What is Encapsulation and Data Abstraction, What is an inheritance and advantages of inheritance, What is a polymorphism

UNIT_034 : OOPS - PROGRAMMING ENCAPSULATION: : 04 HRS

How to create a WindowsForms application, How to create a class and How to declare field members in it , How to Design GUI using Controls in the ToolBox, How button click event works , How Garbage collector will destroy the objects and What are the generations in Garbage Collector ,What is an instance Method and What is the use of this keyword inside a method , What are properties and What does a get and set block do ,What is the difference between constructor and Destructor ,Where the static members allocate memory, When the memory is allocated for static members

UNIT_035 : OOPS - INTERFACE AND POLYMORPHISM : 05 HRS

What is an interface ,How does multiple inheritance is working with interfaces ,How to solve if two interfaces having same method name ,What is publicly implemented and Explicitly implemented ,Why does the .net doesn't support multiple inheritance using classes

UNIT_036 : EXCEPTION HANDLING : 05 HRS

What is an Exception and types of Exceptions, How to handle Exception using try and catch blocks ,How to throw an Exception using throw ex and throw, What is finally Block

UNIT_037 : EXTENDED C# LANGUAGE FEATURES : 05 HRS

What is Operator Overloading ,What is the partial class, partial methods ,What are Extension Methods , What are Anonymous Types ,What are Tuples ,What is caller Information

UNIT_038 : WPF 05 HRS

Environment setup, Element Tree, Dependency Properties, Controls, Layouts, Resources, Data binding, Styles, Triggers, Custom controls, Exception Handling, 2D and 3D Graphics

UNIT_039 : WCF 05 HRS

Overview and Setup, Developer Tools, Creating, Hosting WCF Service, Consuming WCF Service, Service Binding, Transactions , RIA Framework, Security

UNIT_040 : SQL SERVER 05 HRS

Introduction, DDL, DML Statements, Working with Queries, DCL, Working with stored procedures, functions and Triggers

UNIT_040 : WEB API 05 HRS

Programming the API controller: Mapping CRUD operations to HTTP verbs GET, POST, PUT / PATCH / MERGE, DELETE, Connecting service URLs to operations with routing, Web API clients and hosting , Accessing Web API services from .NET, web and mobile clients, Retrieving XML, JSON, and Atom formatted data, Hosting Web API services in IIS and custom applications, Exploiting Web API service features, Integrating Web API services with Entity Framework, Validating with model binding and data annotations, Securing Web API services, Tracing, Logging, and API Documentation, Optimization and Performance

UNIT_041 : REST API 05 HRS

What is Webservices? ,Why Webservices so Popular? , Overview of SOAP Webservices and REST Webservices ,What is REST API? , How is different from SOAP Webservices? Base URL and REST Resources, Understanding of GET, POST, PUT, DELETE

UNIT_042 : ENTITY FRAMEWORK 05 HRS

Entity Framework Introduction, Entity Framework Introduction, and First Example, CRUD Operations, CRUD Operations with BO Class , Stored Procedure Execution, Querying Database, Entity SQL , Eager and Lazy Loading Queries, Additional Features , Inheritance of Entities , Modelling Techniques , Modelling Techniques

UNIT_043 : MICROSERVICES 05 HRS

Monolithic vs Microservices Architecture, Microservices Design Principles, Communication between Microservices, Setup of Microservice Based Application, Creating a Solution and Project Layout, Implementing a CRUD microservice, Writing Domain Classes and Controllers, Data Context Class and Data Seeding Using Repository Classes ,Swagger and Swashbuckle Integration

UNIT_040 : SECURITY 05 HRS

ISecurity threats, different types of security, Core concepts of security, Role based security, Cryptographical security, Secure coding guidelines,

UNIT_040 : DESIGN PRINCIPLES AND DESIGN ARCHITECTURE 05 HRS

SOLID Principles, creational, structural, behavioral design patterns

UNIT_041 :

CAPSTONE PROJECT

LAB SET C PROGRAMMING

LAB 1: BASIC SYNTAX AND VARIABLES

- Write a C program to display "Hello, World!" on the screen.
- Write a program to calculate the sum of two numbers entered by the user.
- Write a program to convert temperature from Celsius to Fahrenheit.

LAB 2: CONTROL STATEMENTS AND LOOPS

- Write a program to find the largest among three numbers using if-else statements.
- Write a program to check whether a number is prime or not using a for loop.
- Write a program to print the Fibonacci series using a while loop.

LAB 3: ARRAYS

- Write a program to find the sum and average of elements in an array.
- Write a program to search for an element in an array and display its index.
- Write a program to sort an array in ascending order using the bubble sort algorithm.

LAB 4: FUNCTIONS AND POINTERS

- Write a program to find the factorial of a number using a recursive function.
- Write a program to swap two numbers using call by reference.
- Write a program to find the length of a string using pointers.

LAB 5: FILE HANDLING

- Write a program to read and display the contents of a text file.
- Write a program to count the number of characters, words, and lines in a text file.
- Write a program to copy the contents of one file to another file.

LAB 6: STRUCTURES AND DYNAMIC MEMORY ALLOCATION

- Write a program to store and display the details of a student using a structure.
- Write a program to add and display the marks of students using dynamic memory allocation.
- Write a program to find the sum and average of marks using a structure array.

LAB 7: RECURSION AND ADVANCED CONCEPTS

- Write a program to calculate the power of a number using recursion.
- Write a program to reverse a string using recursion.
- Write a program to implement a simple calculator using switch case statements.

LAB 8: ADVANCED DATA STRUCTURES

- Write a program to implement a stack and perform push, pop, and display operations.
- Write a program to implement a queue and perform enqueue, dequeue, and display operations.
- Write a program to implement a linked list and perform insertion, deletion, and display operations.

LAB 9: MINI PROJECT

- Design and implement a program that simulates a library management system.
- Develop a program that performs basic banking operations such as deposit, withdrawal, and balance inquiry.
- Create a program that manages a student record system with functionalities like adding, searching, and deleting student record

THANK YOU

VISIT US @

+91 63635 07858

www.uplsnipe.com

D0WLOARD THE APP

GET IT ON Google Play

/uplsnipe

uplsnipe

#123, 70Th Cross, 2Nd Floor, SVA Arcade, 5Th Block Rajajinagara, Bengaluru - 560010