IT Technology Career Path Lectureflow

Level 1 - Module-1) SE - Overview of IT Industry	5

- Introduction of students
- Career in IT
- Understanding Student Login of TOPS ERP
- Using Lab
- What is Program
- What is programming?
- Types of Programming Language
- World Wide Web
- How Internet Works
- Network Layers on Client and Server
- Client And Servers
- Types of Internet Connections
- Protocols
- Application Security
- Software Applications and its types
- Software Architecture
- Layers in Software Architecture
- Software Environments
- Types of Programming Languages
- Source Code
- Github and introductions
- Student Account in Github
- Types of Software
- Introduction of Software
- Application software
- Software development process
- Software Requirement
- Software Analysis
- System Design
- Software Testing
- Maintenance
- Development
- Web Application
- Designing
- moble application
- DFD
- Desktop Application
- Flow Chart

Level 1 - Module-2) SE- HTML and CSS	6

- What is Internet, HTTP/HTTPS, WWW, Domain name and Top Domain name
- SEO, What is HTML, What is Text Editor, Web Browser, Downloading Text Editor , HTML Structure, First Program in HTML
- 1) HTML Introduction 2) HTML Getting Started 3) HTML Elements 4) HTML Attributes 5) HTML Basic Tags
- 1) HTML Doctypes 2) HTML Layout 3) HTML Head 4) HTML Meta 5) HTML Scripts
- Web Programming Design web pages with HTML structure
- Practical Examples: 1) Create any simple web page to display your name. 2) Importance of meta tag and Doctypes
- Tags and self Closing Tags, Basic Tag , Attribute and Events, Marquee Tag
- HTML - Meta Tags, HTML - Comments, HTML - Images, HTML - Tables, HTML - Lists, HTML - Text Links, HTML - Image Links
- HTML basic tags-P,BR,MARQUEE etc
- HTML Headings HTML Paragraphs HTML Links HTML Text Formatting HTML Styles HTML Images
- Anchor Tag, Img Tag, Image Mapping
- List Tag, Tables, Forms
- PRactical Examples: 1) Create simple Doc and display your name using different heading tag 2) Create link for open google. 3) Create document using all text formatting tags
- Form tags with input tag
- Practical Examples: 1) Create simple table 2) Create time table for your school 3) Create table with colspanrowspan example 4) Create invoice using table 5) Create hotel menu. 6) Create index page for your book. 7) Create list with different categories.
- PRactical Examples: Create registration form with all fields and validation
- 1) CSS 2) In-line CSS Internal Style External Style Sheet @import Style Sheet 3) CSS Class CSS ID
- What is CSS How to Implement CSS Class and ID Width and Height Css Unit Box Model (Margin,padding,Border) and create basic template design
- CSS Selectors , Pseudo Classes and Elements , Float and Clear and Alignment , Font Styling , Opacity and Visibility , Line Height
- Practical example : Create page with difference color text
- 1) CSS Text 2)CSS Font 3) CSS Background 4) CSS Links 5) CSS Lists 6) CSS Display 7) CSS Visibility

Level 1 - Module-3) SE - Fundamentals of Programming	15

- Basic Syntax
- Data Structures
- Variables
- Operators
- Control and looping Structures
- functions
- Arrays and strings

- Introduction to C
- What is Language?
- What is programming and program?
- Fundamental of Algorithms and Flowchart
- Real world problems - get solution via programs
- Practical Example: 1. Write a Flow chart of real problems - Days to month conversion system.
- Data Types and Variables - Data Types, Void Data Types,
- Compiler and interpreter
- environment setup
- Type Modifiers,
- Basic Structure of C Programs
- Importance of C
- Fundamentals of C
- Practical Example : 1. Write a program of scanf 2. Write a program to demonstrate escape sequence 3. Write a program to demonstrate comments
- Comments
- Keywords
- Escape Sequence
- Practical Example: 1. Write a program to print (Hello World). 2. Write a program to print the sum of two numbers. 3. Write a program to exchange values of two variables using the 3rd variable. 4. Write a program to convert days into years and years into days.

Level 1 - Module-4) OOP Concept	8

- Procedure Oriented And object Oriented Programming
- Basic Concepts of OOP
- OOP - Objects and Classes
- Constructors and Destructors
- Data Abstraction and Encapsulation
- inheritance
- Encapsulation
- Types of polymorphism
- Dynamic Binding
- Array
- Types of constructors
- Compile time
- Types of Array
- Class and arrays : 1) Array within class 2) Array of objects
- Run time
- String
- Practical Example: 1. Write a program to print the score card of two students using an array of objects.
- Difference between constructor and destructor

- Practical Example: 1. Write a program to demonstrate difference between constructor and destructor 2. Write a program to demonstrate copy constructor
- Abstract class
- Practical Examples: 1. Write a program to check whether entered number is even or not using if..else statement in C++ 2. Write a menu - driven program to calculate the area of the circle,rectangle and triangle. 3. Write a program to calculate factorial of given number using for loop 4. Write a program to print the fibonacci series using while loop 5. Write a program to check whether the given number is palindrome using do..while loop. 6. Write a program to demonstrate jumping statements
- Practical Example: 4. Write a program to demonstrate pass object to a function 5. Write a program to demonstrate return object from function
- Class and pointer
- Aggregation
- Class and objects
- Practical Example: 1. Write a program to demonstrate pointer with class 2. Write a program to demonstrate dynamic object using new keyword
- Access modifiers
- Practical Example: 1. Write a program to demonstrate function overloading with different types of arguments 2. Write a program to demonstrate function overloading with default arguments 3. Write a program to show the constructor function overloading
- Member Function
- Types of inheritance 1 - Single level 2 - Multi-level 3 - Multiple 4- Hierarchical 5- Hybrid
- Comparisons of class and object
- Practical Example : Write a program to implement single level inheritance 2. Write a program to demonstrate single level inheritance in private mode 3. Write a program to demonstrate the ambiguity in single level inheritance 4. Write a program to demonstrate multilevel inheritance 5. Write a program to demonstrate multiple inheritance 6. Write a program to demonstrate the hierarchical inheritance 7. Write a program to demonstrate the hybrid inheritance
- Namespace
- Static Keyword
- Practical Example: 1) Write a program to demonstrate constructor invocation in inheritance
- Scope resolution operator

Level 1 - Module-5) SE - Database	**9**

- What is Database
- DBMS and RDBMS
- Types of Database
- Normalization
- algebra
- Primary key
- foreign key
- unique key
- Database Programming Language SQL
- SQL Statements Types

- DDL
- DML
- TCL
- TQL
- Database backup and Restore
- What are Joins
- Types of Joins
- Function
- Procedure
- Trigger
- Curser
- Transaction concepts
- properties of transactions
- rollback and commit savepoint
- ER database schema

Level 2 -Module 1) - Java -Introduction	**1**

- Introduction Lecture
- Introduction of students
- Understanding Student Login of TOPS ERP
- Working on Project and Assignment
- Using Lab
- Career in IT

Level 2 - Module 2) - Java - Core Java	**15**

- Practical Example : 1. Create class named student with variable rno, fname, lname, email, mobile. create 2 methods to get student data and print them.
- Practical Example : 1. Create class box with three variable height, width, depth. Create default, parameterized and copy constructor. Create one method called volume to show width*height*depth. Call all constructor and volume method for all constructor
- Practical Example : 1. One dimensional array(get data by scanner and print it). 2. Array array elements in asceding and desceding order
- Practical Example: 1. Create 2 two dimensional array and perform matrix addition, subtractio and multiplication
- Practical Example: 1. Perform singleinheritance. 2. Multilevel. 3. Hierarchical.
- Practical Example : 1. Perform constructor chaining
- Practical Example: 1. Method overloading. 2. Method overriding 3. Dynamic method dispatch to solve method override
- Practical Example: 1. Create abstract class RBI with one abstract method interest rate and extend this class in three class SBI, HDFC, Kotak to implement abstract method. 2. Create interface with 2 method. Implement this method in two class. 3. Create program for inheritance of interface. 4. Create program to implment static method in interface and call it in a class

- Practical Example: 1. Write a program to show the use of this keyword in assigning values to variable, as argument in constructor and method, call the default constructor in parameterized constructor using this, and call the method using this. 2. Write a program to demonstrate the difference between static and non static variable. 3. Write a program to to create a static method and static block. 4. Demonstrate teh use of final variable, mthod and class. 5. Access the variable, methods and constructor from d
- Practical Example: 1. String class &amp; its method 2.Perform StringBuffer class methods
- Practical Example: 1. Demonstrate the devide by zero, input mismatch exception and arrayindexoutofbounds exception in a multi catch and multi try statement. 2. Create a method called demo and enter user defind integer value at runtime, if user enters negative value ask again to put value using recursion others wise throw an exception and handle it. 3. Create above program using throws clause without recusrion. 4. Demonstrate the finally block. 5. How to use exception in method override
- Practical Example: 1. Create custome excepton insufficientfund. Create class named bank and create two methods deposite and withdraw. If withdrawal amount is greater than balance then throw user defined exception and handle it.
- Practical Example: 1. Write a progra to write 1 string data into the file using FileOutputStream and read that file using FileInputStream.
- Practical Example: 2. Do above operation using FileWritter &amp; FileReader
- Practical Example: 3. Create one class name student with rno, fname, lname &amp; email and store values of variable into object and then write that object into file and read it.
- Practical Example: 4. Print all the basic property of file that is available in your c:\ drive. You create tops1.txt and put some text into it.
- Practical Example: 1. Pass the 2 integer values through command line and print the maximum number from this.
- Practical Example : 1. Print the current thread that is by default available and then change it's name and again print it. 2. Create a thread usinf Runnable interface. 3. Create a thread using Thread class. 4. Create multiple thread and execute it in main method. 5. Create multiple thread and execute them simultaneouly and achive synchronization. 6. Create two synchronized thread and perform deadlock
- Practical Example: 1. Make a ArrayList with different type of data and perform it's different method. 2. Iterate ArrayList data in both direction from first to last and last to first. 3. Demonstrate HashSet with it;s method. 4. Demonstrate HashMap and iterate it's data. 5. Perform enumeration with Vector class. 6. Create generic method to print different types of array of diffent wrapper classes. 7. Demonstrate Comparator 8. Demonstrate Comparable.
- Practical Example: 1. Create swing GUI with id, fname, lname &amp; email and perform CRUD operation with mysql datanase.
- Conditional Statements (If, If Else, Nested If Else If)
- Introduction of Core Java
- Practical Example : 1. Odd-Even, 2. Prime Number, 3. Max out of three, 4. Student's grade system
- Eclipse IDE
- (Switch Case)
- JVM,JDK,JRE
- Practical Example : 1. Mini Calculator
- Class, Object Constructor

- Loops (While, Do While, For)
- Class, Object, Method
- Practical Example : 1. Sum of n numbers, 2. patterns, 3. prime numbers for a range
- Constructor
- Break and Continue
- Garbage Collection
- Practical Example : 1. exit or continue from loop using break &amp; continue
- Finalize
- SDLC Process
- Project Analysis
- Source File Layout
- Analysis In Details
- DFD (with practical)
- Package Management, Modifiers- Public, Private, Protected, Default
- Introduction of DFD
- Import Statement
- Rules for Drawing DFD
- Context Level
- Data types
- First Level
- Primitive Types
- Second Level
- Reference Types
- Array Introduction
- Data Dictionary
- Modifiers - Public, Private, Protected, Default
- Why Array? Advantages
- Flow Chart
- Types of Array
- Resizing Array
- Copying Array
- Primitive types and Reference type Arrays
- Encapsulations
- Advantages of Inheritance
- Types of Inheritance
- Practical of Inheritance
- Practical of Inheritance with Constructor
- Polymorphism
- Types of Polymorphism
- Method Overloading and Method Overriding
- Abstract and Interface - Introduction and Difference
- Keywords - This, Static, Final, Super
- Classes
- Object Class(only Important Methods)

- String Class (Only Important Methods)
- String Buffer &amp; String Builder
- Wrapper Classes
- Exceptions
- Introduction - Why Exceptions
- Types of Exceptions
- Try catch and Finally Block
- Multi catch Exceptions
- Throw and Throws keywords
- Method Overriding with Exceptions
- Custom Exceptions
- FILE I/O
- What is Stream and Types of Stream
- File Input Output Streams and Its Methods
- File class
- Command Line Arguments
- Thread-Introduction
- Thread Life Cycle
- Creating Threads
- Thread Class Methods (Only Important Methods)
- Runnable Interface
- Synchronized block and Synchronized Methods
- Collection Framework - Introduction
- Collection API
- Hierarchy of Collections
- List and Set and Map Collections
- Array list, vector and Other Classes
- Generics
- Comparator and Comparables
- JAVA GUI
- AWT (Introduction only) &amp; Swing (in Details)
- Components, Containers, Frame, Window, Panel, Layout
- All Components
- Events, Event Handling

Level 2 - Module 3) Java - RDBMS &amp; Database Programming With JDBC	**5**

- Database
- DBMS and RDBMS
- Introduction MYSQL
- Mysql IDE
- Query Types
- DDL, DML, DQL, DCL

- Constraints : Primary Key, Foreign Key, Unique Key
- Normalizations: 1NF 2NF 3NF
- Joins: All Joins Types
- Advance Database: Indexers Views Procedures Functions Cursor, Triggers
- JDBC (Insert, Update, Select, Delete)
- Introduction of JDBC
- Driver Types
- Steps for Creating Connections
- Types of Statements (Statements, prepared Statements and Callable Statements)
- Result Set Interface
- Database Metadata
- Result Set Metadata
- Practical Examples: SQL Queries
- Practical Example : 1. Create swing GUI with id, fname, lname &amp; email and perform CRUD operation with mysql datanase. 2. Demonstrate callble statement in &amp; out parameter.

Level 2 - Module 4) Java - Web Technologies In Java	**12**

- Practical Example: 1. Perform server side validation using filter.
- Action JSTL Custom Tags
- Comments
- Declaration Implicit Objects
- Directives - Scriplets
- Expression
- JSP Life Cycle
- JSP Translation
- Practical Example: JSP Translation JSP Life Cycle Comments Directives Scrip lets Expression Declaration Implicit Objects Action JSTL Custom Tags
- Cookies Session
- Hidden Form Fields
- Session Management - Introduction
- Session Tracking Technique
- URL Rewriting
- What are needs?
- Practical Example: 1. Create registration form, after validation insert data to database and redirect to login form, if successfull login manage session data and logout. 2. Create complete CRUD operation for user profile management.
- Design Pattern MVC Design Pattern with Example
- Practical Example: 1. Perform MVC CRUD operation
- AJAX Programming With Example
- Practical Example: 1. Perform dynamic search operation in project using AJAX. 2. Register user with unique email uing AJAX
- Introduction to Distributed Technologies RMI , EJB and WEB Services Introduction Types of Web Services What is Restful Web Services? Restful Web Services Annotations Restful Web Services

with Example

- Practical Example: 1. Restfull web service CRUD operation
- HTML UL, Tag LI, Tag a, Tag IMG, tag Table, TR, TD, tag
- Form tags with Attributes
- All input tags CSS
- Types of CSS Pseudo- Classes Margins and Puddings
- CSS background
- CSS using ID and Class
- JavaScript Events
- Validations with Regular Expressions
- Firebug Template Integration
- Practical Example: 1. Basic HTML Tags 2. Create Registration form and perform required and regular expression validation for firstname(only alphabets allowd), email(standard email id), mobile number(only 10 digits). 3. Perform all type of css, class &amp; id, pseudo code.
- Introduction of Client Server Architecture
- HTTP Protocol overview with Request and Response header explanation
- J2EE Architecture Overview
- Web Component Development In Java CGI Programming Process Advantage and Disadvantage
- Servlet Programming Introductions Advantage and Disadvantage
- Servlet Versions, Types of Servlets
- Difference between HTTP Servlet an Generic Servlet
- Servlet Life Cycle
- Creating Servlets Servlet Entry in web.xml
- Logical URL Servlet Config Interface
- Request Dispatcher Interface Forward and Include Methods
- Request Dispatcher Interface
- Servlet Context Interface Web Application Listener Scope of Objects, Request and Response Application (Context)
- Practical Example: 1. Fetch data from web.xml to particular servlet using ServletConfig. 2. Fetch data from web.xml to multiple servlet using ServletContext. 3. Create one registration form in jsp and send data to servlet, from servlet again send data to jsp using RequestDispatcher. 4. Create login form in jsp and after login send uname &amp; password to servlet, check data if not blank go forward and if blank then include login.jsp page to servlet.
- Java Filters - Introduction What are the needs Filter Life Cycle Process of Execution Filter Applying Filter Entry in web.xml URL Pattern with Filter

Level 2 - Python ----&gt; Module 1) Python - Fundamentals of python language	**6**

- Introduction of students
- Understanding Student Login of TOPSERP
- Career in IT
- Using Lab
- Introduction of Python

- Programming Style
- Core python concepts
- Conditional Statements
- If- else Nested if-else
- Practical Examples: 1) How to the python code Structure work? 2) How to create variable in python? 3) How to take user input? 4) How to check the type of variable dynamically. 5) W.A.P to find greater and less than number using If_else 6) W.A.P to find prime number using if_else 7) W.A.P to find the grade according to percentage using if_else ladder. 8) W.A.P to find that who can donate the blood using Nested if.
- Looping For , While
- Nested loops
- Control Statements
- 1) WAP to print each fruit in list using simple for loop. List1 (apple,banana,mango) 2) WAP to find the length of string using simple for loop List1 (apple,banana,mango) 3) WAP to find particular string using simple for loop and simple if condition. 4) Print this pattern using nested for Loop.
- Break
- Continue
- Pass
- Practical Example: 1) W.A.P to skip the (Banana) from the list using Continue Statement List1 - (apple,banana,mango) 2) W.A.P to break the for loop when (Banana) get in if Condition.
- String Manipulation
- Accessing Strings
- Basic Operations
- String slices
- Function and Methods
- 1) W.A.P to print (Hello) using string 2) W.A.P to allocate the string to a variable. 3) W.A.P to print String using three quotes 4) W.A.P to access the 1st position character using index value. 5) W.A.P to Access the string after the index value 1. 6) W.A.P to Access the string before the index value 5. 7) W.A.P to Access the String between the index value 1 to 4 8) W.A.P to print the string from the last index value. 9) W.A.P to print the String alternate character after the index value 1. 10) W.

Level 2 - Module 2) Python - Collections, functions and Modules in Python	**5**

- Accessing list
- Operations
- Working with List
- Function and Method
- Practical Example: 1) W.A.P create the list of multiple datatype element. 2) W.A.P to find the length of the list. 3) W.A.P to update the list using the insert() and append() 4) W.A.P to remove the element using the pop() and remove()
- Tuple
- Accessing Tuples
- Operations Working

- Functions and Method
- Dictionaries
- Accessing value in dictionaries
- Working with dictionaries
- Property
- Practical Example: 1) W.A.P to access value on index value in the list 2) W.A.P to access the value after the index value 1. 3) W.A.P to access the value between 1 to 5 4) W.A.P to access the value till index 5. 5) W.A.P to update the list using the index value. 6) W.A.P to irate the list using for loop. 7) W.A.P to insert the value in empty list using for loop and append(). 8) W.A.P to delete the element using del() 9) W.A.P to sort the list using sort() and sorted()
- 10) W.A.P to round the value in list using round() and for loop. 11) W.A.P to convert the list into tuple. 12) W.A.P to create tuple with multiple data type. 13) W.A.P to concate the two tuple into one tuple. 14) W.A.P to access the value of index value 1st in tuple. 15) W.A.P to access the value from last in tuple. 16) W.A.P to access the value between index 1st to 5th from the tuple. 17) W.A.P to access the alternate value between index 1st to 5th.
- 18) W.A.P to create the dictionary of having 6 key and value pair. 19) W.A.P to access the value using the key from dictionary. 20) W.A.P to update the value on particular key. 21) W.A.P to separate the key and value from dictionary using keys() and values() of dictionary. 22) W.A.P to convert the two list into one dictionary using for loop. 23) W.A.P to convert the list using zip() of dictionary. 24) W.A.P to count the character repeat in string.
- Function
- Types of Function
- Function Argument
- anonymous function
- Practical Example: 1) W.A.P to print the String using the function. 2) W.A.P to create the parameterized function. 3) W.A.P to print multiple string using function. 4) W.A.P to create calculator using function. 5) W.A.P to create lamba function using one expression. 6) W.A.P to create lamba function using two expression. 7) W.A.P to create lamba function using three expression. 8) W.A.P to create a return type function using lamda function.
- Modules
- Importing Module
- Math Module
- Random module
- Packages
- Practical Example: 1) W.A.P to import another module into one module. 2) W.A.P to use all the Math module function.

Level 2- Module 3) Python - Advance python programming	**15**

- Printing on screen
- Reading data from keyboard
- opeaning and closing file
- reading and writing file

- Practical Example : 1) W.A.P to create the file using the python. 2) W.A.P to create a file and print the string into the file. 3) W.A.P to read a file and print the data on console. 4) W.A.P to write the multiple String into file 5) W.A.P to read multiple String from the file. 6) W.A.P to check where is the cursor in the file.
- Exception Handling
- Handling Exception
- Finally Clause
- PRactical Example: a) W.A.P to handle exception in calculator. b) W.A.P to handle multiple exception at time in one program. c) W.A.P to handle File Exception and use finally block for closing the file. d) W.A.P to print multiple exception using if else. e) W.A.P to print user define exception.
- class and object
- Attribute
- Inheritance
- Overloading
- Overriding
- Practical Example: 1) W.A.P to create a class and access the property of class using object. 2) W.A.P to create local variable and global variable. 3) W.A.P to show single inheritance. 4) W.A.P to show Multilevel inheritance. 5) W.A.P to show Multiple inheritance. 6) W.A.P to show Hierarchical inheritance. 7) W.A.P to show Hybrid inheritance. 8) W.A.P to using super() in inheritance. 9) W.A.P to show method overloading. 10) W.A.P to show Method overriding.
- Search Function
- Match Function
- Matching Vs Searching
- Modifiers
- Practical Examples: 1) W.A.P to search a word from the string using Search() 2) W.A.P to match the word in string using Match().
- Socket
- Socket Modules
- Methods
- Clients and Sever
- Internet modules
- Practical Example: 1) W.A.P to create the server with all its method. 2) W.A.P to create the client with all its method. 3) W.A.P to show the communication between client and server.
- Thread
- Started a thread
- Threading module
- Synchronizing threads
- Multithreaded Priority Queue
- Practical Example: 1) W.A.P to create a thread of function 2) W.A.P to create multiple thread of multiple function 3) W.A.P to join the thread using join(). 4) W.A.P to synchronize the multiple thread.
- GUI Programming Introduction Tkinter programming
- Tkinter widgets
- Practical Example: 1) W.A.P to create GUI Frame. 2) W.A.P to create all the widgets using Tkinter.

Level 3 - Module 5) - Working with Numpy	**3**

- What is AI , Ml And Dl and Difference Between them,Project life Cycle of ML Project,Math Refresher, basic statistics
- creating 1D array and ND array
- Working with numpy functions
- numpy array attributes
- reshaping and ravel function
- Arithmetic operations on array
- Boradcasting and upcasting
- Conditional Operators in array
- array indexing and slicing, boolean indexing

Level 3 - Module 6) - Working with Pandas	**6**

- Handle the Missing And Categorical data,Outliers,Feature Engineering,Model Selection
- working on Series objects
- indexing on Series
- Creating DataFrame
- Multiindexing in dataframe
- Droping level , transposing
- Accessing rows, Adding and removing columns
- Querying and Sorting DataFrame
- Operations on DataFrame
- Automatic alignment

Level 3 - Module 7) -Ploting data with Matplotlib and Seaborn	**2**

- Creating and customizing line charts using Matplotlib
- Visualizing relationships between two or more variables using scatter plots
- Studying distributions of variables using histograms &amp; bar charts
- Visualizing two-dimensional data using heatmaps

Level 3 - Module 8) - Machine Learning	**10**

- Logistic Regression,K-Nearest Neighbors,SVM And Kernel SVM,Naive Bays
- Decision Tree Classifier,Evaluation Of Classification Model,Small Dataset Assignment
- Introduction to Machine Learning and Problem Domain
- Decision Tree
- Machine Learning Fundamental
- kNNs-SVM-RBF
- Preprocessing Pipeline
- column-transformer

- Linear Models
- Classification metrics
- regression metrics
- K-means Algorithms

Level 4 - Android -----&gt; Module 1) Android - Fundamental	**10**

- Software Engineering with SDLC
- Use Case
- Flowchart
- DFD
- Design project flow
- Design SQL Database
- Practical Example: Diagrams practical in draw.io
- Introduction of Core Java
- Bytecode and JVM
- Java Development Kit
- Eclipse IDE
- JVM, JDK, JRE
- Application and Applet
- Class, Object, Method
- Java Statements
- Java Data type, Variable, Constant, Operators
- Constructor
- Garbage Collections
- Conditional Statement
- Finalize
- Looping Statement
- Arrays in Java
- Source File Layout
- Package Management
- Java Object Oriented Programming
- Import Statement
- Encapsulation
- Data types
- Primitive Types
- Reference Types
- Modifiers- Public, Private, Protected, Default Conditional Statements and Looping Statements
- Array Introduction
- Abstraction
- Java Inheritance
- Why Array?
- Advantages
- Types of Array

- Java Package and Exception Handling
- Resizing Array
- Copying Array
- Primitive types and Reference type Arrays
- Classes
- Object Class (only Important Methods)
- String Class (Only Important Methods)
- String Buffer &amp; String Builder
- Wrapper Classes
- Encapsulations
- Inheritance
- Advantages of Inheritance
- Types of Inheritance
- Practical of Inheritance
- Practical of Inheritance with Constructor
- Polymorphism
- Types of Polymorphism
- Method Overloading and Method Overriding
- Abstract and Interface
- Introduction and Difference Keywords This, Static, Final, Super
- Exceptions
- Why Exceptions
- Types of Exceptions
- Try catch and Finally Block
- Method Overriding with Exceptions Custom Exceptions
- Throw &amp; Throws
- Multi catch Exceptions
- FILE I/O
- What is Stream and Types of Stream
- File Input Output Streams and Its Methods
- File class Command Line Arguments
- Thread
- Thread Life Cycle
- Creating Threads
- Thread Class Methods (Only Important Methods)
- Runnable Interface
- Synchronized block and Synchronized Methods
- Collection Framework
- Collection API
- Hierarchy of Collections
- List and Set and Map Collections
- Array list
- vector and Other Classes
- Generics

- Comparator and Comparable

Level 4 - Module 2) Android - Android Overview &amp;amp; Development	**18**

- View Binding
- Material Design
- AndroidX
- Android Menu
- Menu implementation with bottom
- PRactical Example: 1) Registration form using Material Design 2) Validation 3) Alert dialog practical 4) Custom dialog 5) Custom toast
- navigation and navigation drawer
- Navigation Drawer menus for each entity of database
- Tab Layout
- Android dialog and pickers
- Android Annotations
- View Model
- Practical Example 1) Datepicker dialog &amp; get current date 2) Time picker dialog 3) Context menu 4) Overflow menu 5) Popup menu 6) Working with custom toolbar 7) Working with viewpager - create slider 8) Bottom Navigation drawer 9) Tab layouts 10) Navigation drawer with fragments
- List view
- Spinner
- Practical Example : 1) Listview with static data 2) Dynamic data with listview 3) Listview with adapter 4) Working with spinner as dialog
- Grid view
- Recycler view &amp; Card view
- Practical Example: 1) Grid view - Gallery view 2) Recycler view and Card view 3) Swipe refresh layout with list view 4) Search view with Recycler view
- Introduction to Android
- Development with Android Platforms, Tools, Versions
- Latest updates in Android
- Android Architecture
- Android Installation
- Setup Android Environment
- Building Blocks of Android Application
- Work with Activity
- Intents &amp; intent filters
- Activity Lifecycle
- Working with resources
- Practical Example: 1) Hello world program using textview 2) Activity Lifecycle with Log.d and Toast 3) Working on button click event 4) Phone call via android app 5) Send sms and email via android app 6) Interaction between two activities using intent 7) Pass message between two activities using intent 8) Working with colors.xml and strings.xml file - R.java file practical

- Working with viewgroup and views Layouts
- Practical Example: 1) All Layouts practicals 2) LinearLayout 3) FrameLayout 4) GridLayout 5) Relative Layout 6) Table Layout 7) Coordinator Layout 8) Constraint Layout 9) Custom Layout
- Android widget Tools kit
- Fragments
- Fragment Lifecycle
- 1) Registration and login form all widgets (edit text, checkbox, radio button, radio group) 2) Working with drawables file gradient effect 3) Working with web view - load static html page and dynamic html page 4) Static fragments and dynamic fragments 5) Splash screen using fragment 6) Fragments lifecycle 7) Fragment back events

Level 4 - Module 3) Android - Android : Storage	**5**

- Shared Preference
- Practical Example: 1) Skip login screen if already user login using shared preference 2) Setting screen 3) Login - logout session
- Android Run time permission
- File storage
- Realm - no Sql database
- Practical Example : 1) File read write operations 2) Create folder when application start 3) CRUD operations using Realm-no sql database
- SQLite Database
- Room Persistence Library
- Practical Example: 1) SQlite database CRUD operations 2) Room Persistence Library CRUE operations

Level 4 - Module 4) Android - Android : Online Database	**5**

- XML Parsing : 1) Pull parsing 2) DOM parsing 3) SAX parsing
- JSON Parsing
- Practical Example : 1) XML parsing using 3 types 2) JSON parsing practical 3) JSON Array and JSON object 4) JSON parsing using Gson Library
- Mysql connectivity in Android using web services
- Asynchronous Data loading
- Third party library : 1) Retrofit 2) Glide 3) Picasso 4) Gson
- Practical Example: 1) CRUD operations using MYSQL web services - Retrofit library 2) ASYNC task practical 3) Registration and login via Retrofit 4) Load image using Glide or picasso

Level 4 - Module 5) Android - Advance Android	**13**

- Background Process
- Android Service
- Broadcast - Receivers

- Alarm manager - push notification
- Practical Example: a) Service start - stop practical b) Broadcast receiver practical c) Battery status alert - Broadcast receiver practical d) Alarm manager push notification practical
- Google Map
- Location Services and GPS
- Geo-coding API
- Practical Example : !) Google map API integrations 2) Find current location 3) Polyline draw on map 4) Working with markers and zoom -location button
- Firebase : FCM (firebase cloud messaging)
- PRactical Example : 1) Working with firebase 2) Post and get data from firebase 3) Working with storage specifier using firebase
- Sensors
- Wifi
- Bluetooth
- 1) Sensor practical - screen color change 2) Wifi enable - disable practical 4) Bluetooth Connectivity - Paired list 4) Check internet is connected or not
- Media
- Camera
- Wake lock
- Practical Example: 1) Add new contact in phone book 2) Get all contacts from mobile phone book 3) Play music and video - use raw and assets folder 4) Youtubeapi integrations 5) Browse gallery image on imageview 6) Camera API integrations 7) Transition animation , alpha animation , rotate animation
- Android Animation
- Practical Example : Working with lottie animation library
- Payment integration
- Social Media Integration - Login via Google+, Facebook Google Admob
- Practical Example : 1) Payment integration via Google Pay or paypal 2) Social media integration (Gmail Login or Facebook Login) 3) TextToSpeech , SpeechToText practical 4) Google Banner and Interstitial ad

Level 4 - Module 6) Android - Android Deployment	**2**

- Generate Signed APK
- Deployment
- Public application on play store
- Deploy code on github
- Practical Example : 1) google play developer console account 2) Deploy Application on Play store 3) Deploy code on github

Level 4 - Module 6) Flutter - Introduction	**1**

- Introduction to student Career in Android Understanding Student Login of TOPS ERP Exam Process Working on Project and Assignment Using Lab Assign Project

Level 4 - Module 7) Fundamental - Dart Programming	**7**

- Dart SDK
- Flutter Installation - Android Studio Configuration - Flutter doctor
- Dart Introduction
- Data types in Dart
- String interpolations
- Operators
- Working on Control Statements
- conditional statements : if statement , if..else statement , nested if ,switch statement
- looping statements : for loop , while loop , for .. in loop
- Jumping statements
- Working with collection
- Working with list , set , map and methods
- working with function
- Advance Dart Programing
- Class, Object, Inheritance, Polymorphism
- Keywords : this , super , static , async

Level 4 - Module 8) Flutter - UI Designing and Development	**13**

- Text, elevated button
- Project creation in Flutter - run project in read device and emulator - hot reloading app
- Flutter Architecture
- Working with Row and Column Widgets
- Flutter widgets, Flutter layout
- State ful Widgets
- Stateless Widgets
- Material App
- Scaffold , contain
- Properties of container widgets
- Stateless widgets - row and column widgets
- Working with text and button ontap and onpress event
- icons ,alert dialogs ,radio button , checkbox , switch ,TextFormField - working with all style properties
- Working with forms
- Design registration form
- Design Login Form
- Working with Form key
- Customize widgets
- Form validation , apply email validation – password hide and unhide
- Change dynamic background color on button click
- working with setState, init
- images , network images - working with assets

- Floating Action button with types
- Working with pageviewbuilder and dots controller

Level 4 - Module 9) Flutter - Advance UI Designing and Development	7

- working with init method
- working with them
- Working Routing
- Named Routes
- Arguments in routes
- Return data from screen
- Send data to screen
- Navigation , navigator , push and pop - navigation between different screens
- Listview and list item
- Working with grid view
- working with bottom navigation bar
- working with tab bar with icons
- working with navigation drawer with navigation screens
- Splash screen
- working with listview - dividers
- Flutter gestures

Level 4 - Module 10) Flutter - Offline Database- Sqlite	6

- Database Introduction
- working with Sqflite Database
- Dependencies
- working with model class
- CRUD operations using sqflite database
- working with async , await , future

Level 4 - Module 11) Flutter - Advance App Development	15

- Gallery Access , Camera access in Flutter
- Fetch data from internet
- Working on json parsing
- Animation
- retrieve data and display in listview format
- working with listview indexing
- working with gridview
- working with firebase database
- firebase authentication
- firebase real time database

- notification with firebase
- working with background services
- Google map integration and social media integration with flutter application
- Project implementation - Splash Screen • Login - registration Screen • Database integration or api integration • working with listview - navigation • navigation drawer with user header layout and list tile items
- Making phone calls , sms and url launcher
- Flutter state management
- Bloc concept

Module 5) Java - Rest Framework - Industry - NOT IN USE	**10**

- Design Pattern
- MVC Design Pattern with Example
- AJAX Programming With Example
- Practical Example: 1. Perform dynamic search operation in project using AJAX. 2. Register user with unique email using AJAX
- Introduction to Distributed Technologies RMI, EJB and WEB Services Introduction Types of Web Services What is Restful Web Services? Restful Web Services Annotations Restful Web Services with Exampl
- ractical Example: 1. Restful web service CRUD operation

Level 5 - Module 6) java - Frameworks - Industry	**16**

- All Core Interface Query and Criteria Named Query
- Relationships Many to Many
- Relationships One to Many
- Relationships Many to One
- Hibernate Introduction
- Relationships One to One
- Hibernate Architecture
- All Database Operations with hibernate
- Practical Example: 1. CRUD Operation with hibernate using xml files. 2. CRUD operation with hibernate using annotation. 3. Perform onetoone relationship(Employee class with eid, uname and password &amp; EmployeePersonalINfo class with epid, fname, lname,email). 4. Perform onetomany &amp; manytoone relationship(Employee class with eid, fname,lname,email &amp; Department class with deptno, dname,location). 5. Perform manytomany relationship(Student class with sid, sname &amp; Course class with cid, cname)
- Introduction of Spring Framework Architecture
- Overview Of Spring Framework
- Core Container AOP
- Spring DAO (Data Integration)
- Spring Using IDE, Using Library Spring Hello World Example
- Practical Example: 1. Hello world spring app to introduce spring framework

- 1) Spring IOC Container 2) Bean Factory 3) Application Context Spring Bean Definition 4) Configuration 5) Life Cycle 6) Inheritance 7) Scopes
- Practical Example: 2. Perform spring inheritance, life cycle &amp; abstraction. 3. Perform singleton &amp; prototype scope to use spring beans variable
- 1) Spring Dependency Injection 2) Constructor based 3) Setter Getter based 4) Inner Beans , Aliases and ID-ref Collections and References 5) Auto Wiring
- Practical Example : 4. Demonstrate spring dependency injection by setter method.5. Demonstrate spring dependency injection by constructor. 6. Demonstrate spring dependency injection by object. 7. Perform inner bean concept in xml file. 8. Use all type of collection refences in spring xml file. 9. Minimize spring xml file using spring auto wire concept
- 1) Spring AOP 2) AOP Term 3) Write the Aspects 4) Configure Where the Aspects
- Practical Example : 10. Perfomr AOP(aspect oriented programming concept(login, perform, logout sequence)
- Spring ORM
- Practical Example : 11. Perform CRUD operation in spring web using hibernate integration
- 1) Spring MVC Web Forms 2) Spring Form Handling 3) Spring Form Tags 4) Spring Controller XML and Annotation Based
- Practical Example : 12. Create spring MVC pattern using dispatcher servlet. 13. CRUD operation using Spring MVC+ORM
- Spring MVC with Session Management
- Practical Example : 14. Spring MVC+ORM+Session

Level 5 - Module 4) Python - DB and Python Framework	**20**

- creating App
- ORM
- Query set
- ajax
- form validation
- deployment
- web app development
- Virtual Environment setup
- admin panel
- Django models
- setting web server
- django forms
- designing
- HTML
- CSS
- javascript
- bootstrap
- Authentication
- URL Pattern
- dynamic data in templates

- Practical Example: 1) Create Django Admin Panel 2) Creating the Doctor Finder Project.
- Simple web application
- Client Server Architecture.
- Intro of Flask and Bottle.
- Advance Web Framework Django
- Connectivity with MySql Connection Steps.
- CRUD Operation Using Tkinter with MVC Pattern
- Practical Examples: 1) W.A.P to preform the CRUD operation using Tkinter. 2) Crud operation using Django.

Level 6 - Web Development -----&gt; Module 1) WD - HTML and HTML5	**10**

- 1) HTML Doctypes 2) HTML Layout 3) HTML Head 4) HTML Meta 5) HTML Scripts
- 1) HTML Introduction 2) HTML Getting Started 3) HTML Elements 4) HTML Attributes
- Practical Examples: 1) Create any simple web page to display your name. 2) Importance of meta tag and Doctypes
- HTML Headings HTML Paragraphs HTML Links HTML Text Formatting HTML Styles HTML Images
- PRactical Examples: 1) Create simple Doc and display your name using different heading tag 2) Create link for open google. 3) Create document using all text formatting tags
- HTML Tables HTML Lists HTML Forms HTML Iframes
- Practical Examples: 1) Create simple table 2) Create time table for your school 3) Create table with colspanrowspan example 4) Create invoice using table 5) Create hotel menu. 6) Create index page for your book. 7) Create list with different categories.
- HTML Entities HTML Validation HTML-HTML5 Tags
- PRactical Examples: Create registration form with all fields and validation

Level 6 - Module 2) WD - CSS and CSS 3	**16**

- 1) CSS 2) In-line CSS Internal Style External Style Sheet @import Style Sheet 3) CSS Class CSS ID
- Practical example : Create page with difference color text
- 1) CSS Text 2)CSS Font 3) CSS Background 4) CSS Links 5) CSS Lists 6) CSS Display 7) CSS Visibility
- PRactical Example : Create layout for your project
- 1) CSS Layout Model 2) CSS Border 3) CSS Margin 4) CSS Padding 5) CSS Outline
- 1) CSS Float 2) CSS Align 3) CSS Position 4) CSS Element Size 5) CSS Layer
- Practical Example : Create image gallery
- 1) CSS Pseudo Class Selector 2) CSS Pseudo Element Selector
- CSS Properties 1) Background, 2) border 3) bottom 4) caption-side 5) clear 6) clip 7) color 8) content
- Practical Example: Create Menu with logo at left side and contact info at right side using clear effect
- 1) counter-increment 2) counter-reset 3) cursor 4) direction 5) display 6) empty-cells
- Practical Example: 1) Create submenu list using counter

- 1) float 2) font 3) height 4) left 5) letter-spacing 6) line [height, style, style-7) image, style-position, 8) style-type] 9) margin 10) outline 11) overflow 12) padding
- 1) page-break 2) position 3) quotes 4) right 5) table-layout 6) text 7) top 8) vertical-align 9) visibility 10) white-space 11) width 12) word-spacing 13) z-index
- Practical Example: wireframe layout for your template using div

Level 6 - Module 3) WD - JAVASCRIPT BASIC &amp; DOM	**10**

- 1) JS Introduction 2) JS Getting Started 3) JS Syntax 4) JS Variables 5) JS Generating Output 6) JS Data Types 7) JS Operators 8) JS Events 9) JS Strings 10) JS Numbers 11) JS If, Else 12) JS Switch Case
- Practical Example: 1) Create program for input color and output that code 2) Create program for pattern using loop
- a) JS Arrays b) JS Sorting Arrays c) JS Loops d) JS Functions e) JS Objects
- Practical Example: 1) Create function Finding the Maximum and Minimum Value in an Array 2) Create pyramid pattern program
- JAVASCRIPT &amp; DOM 1) JS DOM Nodes 2) JS DOM Selectors 3) JS DOM Styling 4) JS DOM Get Set Attributes 5) JS DOM Manipulation 6) JS DOM Navigation
- Practical Examples: 1) Get input data and perform different operations 2) Make dynamic CSS by click
- JAVASCRIPT ADVANCED 1) JS Date and Time 2) JS Math Operations 3) JS Type Conversions 4) JS Event Listener 5) JS Regular Expressions 6) JS Error Handling
- Practical Example: Create custom Validation

Level 6 - Module 4) WD - JQuery Basic, Effects &amp;amp; Advanced	**8**

- jQuery Basic a) jQuery Introduction b) jQuery Getting Started c) jQuery Syntax d) jQuery Selectors e) jQuery Events
- Practical Example: Change CSS
- JQuery Effects 1) jQuery Show/Hide 2) jQuery Fade 3) jQuery Slide 4) jQuery Animation 5) jQuery Stop 6) jQuery Chaining 7) jQuery Callback
- Practical Example: Create slider with animation
- JQuery Advanced 1) jQuery Traversing 2) jQuery Ancestors 3) jQuery Descendants 4) jQuery Siblings 5) jQuery Filtering 6) jQuery Load 7) jQuery No-Conflict

Level 6 - Module 6) WD - Bootstrap Basic &amp; Advanced	**9**

- Bootstrap Basic 1) Bootstrap Introduction 2) Bootstrap Getting Started 3) Bootstrap Grid System 4) Bootstrap Fixed Layout 5) Bootstrap Fluid Layout 6) Bootstrap Responsive Layout
- Bootstrap Utilities
- 1) Bootstrap Typography 2) Bootstrap Tables 3) Bootstrap Lists 4) Bootstrap List Groups 5) Bootstrap Forms 6) Bootstrap Custom Forms 7) Bootstrap Input Groups 8) Bootstrap Buttons 9)

Bootstrap Button Groups

- Bootstrap with CSS: Grid System
- 1) Bootstrap Images 2) Bootstrap Cards 3) Bootstrap Media Objects 4) Bootstrap Icons 5) Bootstrap Navs 6) Bootstrap Navbar 7) Bootstrap Breadcrumbs 8) Bootstrap Pagination 9) Bootstrap Badges 10) Bootstrap Progress Bars 11) Bootstrap Spinners 12) Bootstrap Jumbotron 13) Bootstrap Helper Classes
- Bootstrap with CSS - typography
- Bootstrap Advanced 1) Bootstrap Modals 2) Bootstrap Dropdowns 3) Bootstrap Tabs 4) Bootstrap Tooltips 5) Bootstrap Popovers 6) Bootstrap Alerts 7) Bootstrap Stateful Buttons 8) Bootstrap Accordion 9) Bootstrap Carousel 10) Bootstrap Typeahead 11) Bootstrap ScrollSpy 12) Bootstrap Toasts
- Bootstrap with CSS: Tables
- Bootstrap Striped Row Table
- Bootstrap Bordered Table
- Bootstrap Hover Row Table
- Bootstrap Condensed Table
- Bootstrap Contextual Classes
- Bootstrap Responsive Table
- Bootstrap with CSS - Forms
- bootstrap 4 Form
- Bootstrap with CSS - Buttons
- Bootstrap with CSS - Images
- Bootstrap helper Classes
- Border Classes
- Bootstrap with CSS - Responsive Utilities
- Bootstrap Layout - Glyphicon
- bootstrap Layout - Dropdowns
- Bootstrap Dropup
- Bootstrap - Button Group
- Bootstrap Layout - Navigation Elements
- Navbar
- Breadcrumb
- pagination
- Input Group
- Labels
- Badges
- Jumbotron
- Page Headers
- Alerts
- Progress Bar
- List Group
- Panels
- Wells

Level 7 - Module-3) React - Components, State, Props	8

- Installation - Add React to a HTML Website - Create New React App - Hello World
- Getting started in React
- JSX
- Components
- Component Composition
- JSX - Why JSX? - Embedding Expressions in JSX - Attributes with JSX - Children with JSX
- Props &amp; Prop Types
- Event Handlers
- State
- React Web App
- Components, State, Props - Function Component - Class Component - Props - State - Class Component Lifecycle

Module 6) JavaScript Essentials And Advanced	25

- Basic JavaScript, Js comment, Js variables , Understanding var, let and Const, JS switch, if, else,JS loop , Js global variables, Js data types, Js operators, Js Functions
- Functions - Function Declaration in JS - Arrow Functions - Higher Order Functions - Map, Reduce and Filter
- Javascript Objects, Js object , Js Array , Js string, Js Date, Js Math, Js number, Js Boolean
- Javascript BOM ,Broswer Objects , Window object, History object, navigator object, Screen object
- Javascript DOM, Document object, getElementById, getElementByName, getElementByTagName, JS innerHTML property, JS innerTEXT property
- Javascript OOPS, JS class, JS object, JS prototype, JS constructor method, JS static method, JS encapsulation, JS inheritance, JS polymorphism, JS abstractions
- Javascript Exception Handling, JS exception handling , Javascript try-catch
- Javascript MISC, JS this keyword , JS Debugging , JS Hoisting , JS Strict Mode, JS promises, JS typeof , JS ternary operator, JS reload() method, JS setAttributes () method, JS setInterval() method, JS setTimeout() method.
- Javascript Events, Javascript Events, Javascript AddEventListener,(), jsOnclick event, jsdbclick event, JS onload event, JS onresize event.
- Array in JS, Creating Array, Array methods, The Spread &amp; Rest operators, Destructuring
- JS Async, Callbacks, Promises, Async/Await
- ES6 Basics and Babel, New features in ES 6, Arrow functions, The . Operator, For/of , Map Objects, Set Objects, Promises, Functions Rest parameter, String.includes(),String.starts.With(), String.endWith(), Array.form(), Array.keys(), Array find(), Array findIndex(), javascript Modules
- Small Project using ES6

Level 7 - Module 4) Lists and Hooks	6

- Conditional Rendering - Lists and Keys - Forms - Handling Events - Lifting State up
- Hooks - Introduction - Using the State hook - Using the Effect hook - Rules of Hook - Custom Hook

- Rendering Lists inside components
- React Keys
- Using keys wit component
- Uniqueness of keys among siblings
- React refs
- Uses of react Refs
- How to access of Refs
- Refs current properties
- Add Refs to DOM elements
- Add refs to class components
- Callback refs
- Forwarding Ref from one component to another component
- React with useRef
- React conditional rendering
- React if, logical &amp; operator, Ternary operator, switch case operator, Conditional Rendering with Enum, Preventing components from rendering

Level 7 - Module-5) React - Styling &amp; Advance React	**5**

- Creating the first App
- Understanding the App
- Styling the App
- Inspecting &amp; Debugging styles
- Built-in components
- Working with Images
- ListViews
- TextInput
- Styling React Components - CSS stylesheet - Inline Styling - CSS Modules - CSS in JS Libraries (styled components)
- Creating Views (Scenes)
- Conditional Rendering - Lists and Keys - Forms - Handling Events - Lifting State up
- Hooks - Introduction - Using the State hook - Using the Effect hook - Rules of Hook - Custom Hook
- Advance Concepts - Context, useContext() - Working with Refs and useRefs() - Fragments - Performance optimization with useMemo() - Styling React Components - CSS stylesheet - Inline Styling - CSS Modules - CSS in JS Libraries (styled components)
- Bootstrap with React
- React Router - Browser - Router - Link - Route - Template integration - Http Request in React - Get and Post data

Level 7 - Module 6) React Router	**8**

- React Router
- Browser-Router-Link-Route
- Need of react router

- Template integration - Http Request in React - Get and Post data
- React router installation
- React router, react-router-native, react-router-Dom
- Component in react router , Browser Router , HashRouter
- What is Route
- What is Link component , Adding navigation using Link component
- Link vs NavLink
- React Router Switch , React Router redirect
- Nested Routing in React
- Template integrations Using Browser Router , Routes , Route , Link and Hash Router
- Advantages of react Router

Level 7 - Module-7) React - Applying Redux	**8**

- State
- State storage problem
- Redux Basics
- Redux Principles
- Implementing Redux
- React-Redux
- Middleware
- Counter App Demo
- Redux - Complexity of Managing state - Understand the Redux Flow - Setting up Reducer and store - Dispatching Actions - Passing and Retrieving Data with Action - Combining Multiple Reducers - Adding Middleware - Redux Dev tools