

UPL PROSPECTOUS

early break even | best experience



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/-

CERTIFICATION COURSE(PG DIPLOMA COURSES):

ABOUT THIS MODEL

- **Target Audience:** Minimum Degree in any discipline (BA, BCOM, MSc, BE CIVIL . ..) /ALL NON-IT members want to restart in the IT
- **Details:** About the Program
- **Course Coverage:** Online virtual 2 Semester course having Foundation of IT in 1st Semester and specialization web development, mobile, cyber, and Data scientist in 2ND Semester with 2 capstone project.

PG DIPLOMA COURSE:

PGD_WEB : WEB DEVELOPMENT

SEMESTER I: CORE MODULE FUNDAMENTALS

CODE	SUBJECT	MARKS	HRS
WEB_001	FUNDAMENTAL PROGRAMMING -C /PYTHON	100	60
WEB_002	ALGORITHMS AND DATA STRUCTURE IN C	100	60
WEB_003	DATABASE AND SQL	100	60
WEB_004	OOAD AND C++	100	60
WEB_005	CRACKING WRITTEN TEST	100	60
	TOTAL	500	360

SEMESTER II:SPECIALIZATION

CODE	SUBJECT	MARKS	HRS
WEB_006	PROGRAMMING IN FRONTEND TECHNOLOGIES	100	60
WEB_007	SOFTWARE ENGINEERING	100	60
WEB_008	JAVA WEB PROGRAMMING	100	60
WEB_009	SOFTWARE ARCHITECTURE	100	60
WEB_010	CRACKING CODING INTERVIEW AND PERSONALITY DEVELOPMENT	100	60
	TOTAL	500	360

WEB_008: JAVA WEB PROGRAMMING

CORE JAVA

UNIT_001 : INTRODUCTION TO JAVA

01 HRS

- Overview of Java and its features
- Installing Java Development Kit (JDK) and setting up the development environment
- Writing and executing Java programs using a text editor and command-line tools

UNIT_002 : VARIABLES, DATA TYPES, AND OPERATORS

01 HRS

- Declaring and using variables
- Primitive data types: int, double, boolean, etc.
- Type conversion and casting
- Arithmetic, comparison, and logical operators

UNIT_003 : CONTROL FLOW AND LOOPS

01 HRS

- Conditional statements: if, else if, else
- Switch statements
- Looping constructs: for loop, while loop, do-while loop
- Control flow keywords: break, continue

UNIT_004 : ARRAYS AND STRINGS

01 HRS

- Declaring and initializing arrays
- Accessing and manipulating array elements
- Working with multi-dimensional arrays
- String manipulation: concatenation, substring, length, etc.

01 HRS

UNIT_005 : OBJECT-ORIENTED PROGRAMMING (OOP) CONCEPTS

- Introduction to OOP
- Classes and objects
- Encapsulation, inheritance, and polymorphism
- Method overloading and overriding
- Constructors and static members

UNIT_006 : EXCEPTION HANDLING

01 HRS

- Handling and throwing exceptions
- try-catch blocks
- Multiple catch blocks and finally block
- Built-in exceptions and creating custom exceptions

UNIT_007 : INPUT AND OUTPUT (I/O) OPERATIONS

01 HRS

- Reading from and writing to files
- Working with file streams and readers/writers
- Handling file I/O exceptions

UNIT_008 : COLLECTIONS FRAMEWORK

01 HRS

- Overview of collections framework
- Working with ArrayList, LinkedList, HashSet, HashMap, etc.
- Iterating over collections using loops and iterators
- Sorting and searching collections

UNIT_009 : JAVA GENERICS

01 HRS

- Introduction to generics
- Using generic classes, interfaces, and methods
- Type erasure and wildcards

UNIT_010 : MULTITHREADING

- Introduction to threads and multithreading
- Creating and running threads
- Thread synchronization and inter-thread communication
- Handling thread exceptions and termination

UNIT_011 : JDBC (JAVA DATABASE CONNECTIVITY)

01 HRS

- Introduction to JDBC
- Establishing database connections
- Executing SQL statements and retrieving data
- Handling transactions and working with result sets

UNIT_012 : INTRODUCTION TO GUI PROGRAMMING

01 HRS

- Basics of GUI programming using Swing or JavaFX
- Creating windows, buttons, labels, and other GUI components
- Handling user events and interactions

LAB SET CORE JAVA

LAB 1 :

- learn to compile and run a very simple Java program
- To know how to use environment
- Learn to use scanner class and take user input
 - (a) WAP print “Welcome to Java Programming” in console
 - (b) WAP print “addition of two numbers”
 - (c) WAP to generate Fibonacci Series
 - (d) WAP to generate Prime Number generation
 - (e) WAP to convert given Celsius to Fahrenheit

LAB 2 :

- learn to compile and run a very simple Java program
- To know how to use environment
- To learn basic principles of Object , Class
- To learn default constructor, parametrized constructor, POJO (Encapsulation)
 - (a)WAP display employee information using class and object(default constructor)
 - (b)WAP display employee information using class and object(parameterized constructor)
 - (c)WAP display employee information using class and object(POJO-setter/getter)
 - (d) WAP to generate Prime Number using object and class (constructor)
 - (e)WAP to build simple calculator(class , object and constructor)

LAB 3 :

- learn to compile and run a very simple Java program
- learn to Arrays
- (a) WAP to find the sum of 'n' Numbers
- (b) WAP to Sort 'n' Numbers
- (c) WAP to Addition, Subtraction and Multiplication of two matrices

LAB 4 :

- learn to compile and run a very simple Java program
- To learn Inheritance and Polymorphism concepts
 - (a) WAP display Dog information reusing Animal (Single Inheritance)
 - (b) WAP to build Parttime employee and full-time employees from the derived class Employee and do necessary changes and implement the same (Hierarchical Inheritance)
 - (c) WAP to build FullTime employee from derived Employee class which is inherited from Person to display Employee information (Multilevel Inheritance)
 - (d) WAP to demonstrate access modifier default, private, protected and public modifier
 - (e) WAP to demonstrate display method of person class as abstract extending Employee class and display the same (abstraction)
 - (f) WAP to build a calculator to display the Addition, Subtraction, Multiplication and Division calculator using interface Calculator and CalculatorImpl (Interface and class implementations)
 - (g) WAP to demonstrate to display compute operation using method overloading (Polymorphism)
 - (h) WAP to demonstrate to override addition, subtraction, division and multiplication method in subclass of calculator (Runtime Polymorphism, interface)
 - (f) WAP to demonstrate abstract class using person class extending the Employee concrete class to display information (abstract class)

LAB 5 :

- learn to use of MATH library
- learn passing command line arguments
 - Demonstrate an example for Constants, Truncating, comparison, power, Trigonometric, square roots and generate random number using Math library

LAB 6 :

- learn to compile and run a very simple Java program
- learn to use String, StringBuffer, and StringBuilder
 - (a) Demonstrate an example extract substring from a string
 - (b) WAP to parse String using String Tokenizer
 - (c) WAP to reverse a String (By character)
 - (d) WAP to reverse a String (By words)
 - (e) Demonstrate an example StringBuffer and String and StringBuilder
 - (f) Demonstrate difference between == and equals operator on String
 - (g) WAP to convert cases of given String
 - (h) WAP to sort given 'N' Strings
 - (i) WAP to the concatenation of String
 - (j) WAP to string comparison

LAB 7 :

- learn the basic jargon of object-oriented programming and how it appears in code
- learn how a Java program is organized into multiple source files
- learn to compile and run a very simple Java program
 - (a) Create an application program that consists of 2 classes, a "startup class" and a second class that prints out the values of at least 3 instance variables that are initialized in one method and printed from another. The initialization method should have two forms. One of them will have no arguments and the other 3 arguments corresponding to the data types of the 3 instance variables. The second class should also contain a "class variable" of one of the 8 primitive data types (you choose). Create 2 instances of the second class. Using one instance, set the class variable to some value and, using the second instance, print that value out. Also, using either or both of the 2 instances, call the methods that set and print the 3 instance variables.
 - (b) Demonstrate a Bank Application
 - Demonstrate 1.Create an Account 2. Deposit 3. Withdraw and 3. Display the account details

LAB 8 :

- learn to use inheritance, Exception Handling, Packaging
- learn to design your own hierarchy
- learn how to write and use a constructor method
- learn how to use access specifiers (public, protected, private, and default or package)
 - (a) Write an application with a hierarchy composed of at least 2 classes. One of these classes will be a subclass derived from the other. The subclass is to have at least 3 constructor methods. Each of these will be called during the creation of 3 subclass objects. In the example shown, the subclass student has a default constructor, a constructor with one string argument, and a constructor with a string argument giving the student's name and an integer argument giving an ID number.
 - (b) Create an hierarchy of Person
 - (c) Assume necessary hierarchy and demonstrate computation of Student Result generation/Marks sheet generation.
 - (d) WAP to create custom exception handling.
 - (e) WAP Develop a Calculator using layered architecture with appropriate exception handling

LAB 9 :

- learn to use I/O Streams
 - (a) WAP to demonstrate Reading/Writing to file using FileInputStream/OutputStream and Reader/Writer classes
 - (b) WAP to reverse file content
 - (c) WAP to concatenate two files
 - (d) WAP to copy file to another
 - (e) WAP to copy all files from src directory to destination directory
 - (f) WAP to demonstrate create, Delete and modify a file.
 - (g). WAP Reading and writing the data using DataInput and DataOutput Streams.
 - (h). WAP Reading and writing using File Input and File output Streams
 - (I). WAP Reading and writing a files using FileReader and FileWriter classes

LAB 10 :

- learn to use Multithreading
 - (a) WAP to demonstrate an example for Thread extending class and Implementing Runnable Interface
 - (b) WAP to demonstrate thread life cycle
 - (c) WAP to demonstrate thread priority
 - (d) WAP to demonstrate different types of Thread Pool
 - (e) WAP to demonstrate Thread call/future

LAB 11 :

- learn to use Collection class
 - (a) WAP to demonstrate an ArrayList
 - (b) WAP to demonstrate Hashtable and Hash Map
 - (c) WAP to implementation Stack and Queue
 - (d) WAP to demonstrate Tree Set and Sorted Set
 - (e) Demonstrate examples with ArrayList
 - (f). Demonstrate the Linked List
 - (g). Demonstrate the Vector
 - (h). Demonstrate the stack implementations
 - (i). Demonstrate the collections utility class
 - (j). Demonstrate the Sort by comparator
 - (k). Demonstrate Deque operations
 - (l). Demonstration of Map interface using HashMap
 - (m). Demonstarate Map using HashTable
 - (n). Demonstrate the ENUMERATIONS

LAB 12 :

- learn to use Java Applet/Swing/AWT
 - (a) WAP to creation of GUI and by assuming necessary steps for student marks sheet generation
 - (b) WAP to Copy file from one directory to another
 - (c) WAP to develop a simple calculator.

LAB 13 :

- learn to use Java Networking
 - **Write a chat application:**
 - **One-One:** By opening socket connection and displaying what is written by one party to the other.
 - **Many-Many (Broad cast) :** Each client opens a socket connection to that chat server and writes to the socket. What ever is written by one party can be seen by all other parties.

LAB 14 :

- learn to use lambda expressions and Streams API
 - (a).WAP to demonstrate simple with and without lambda expression
 - (b).Display lambda expression with one argument
 - (c).Perform addition of two input numbers showcase two arguments
 - (d).Perform showcase default and static keywords usage in functional interface
 - (e).Display the square of number forms of return statement
 - (f).Demonstrate the multithreading example using lambda expression

LAB 15:

- learn to use Annotations
 - (a). Demonstrate @Override annotations
 - (b). Demonstrate the suppress warning annotations
 - (c). Demonstrate the deprecated annotations
 - (d). write a example to demonstrate functional interface
 - (e). Write a custom annotations to show Course details

LAB 16:

- learn to use JDBC Programming
 - (a) WAP to build calculator operations such create, add, mod and delete operations
 - (b) WAP to build simple Employee database management system

LAB 17 :

- CAPSTONE PROJECT – Use Core Java and JDBC

(a) Develop and Design the HR Department which maintains the employee details with basic payroll information. Also Generate necessary report information.

It may be having the following tables

Employee {employee identification number, first and last name, designation, home address, contact number, hire date, work location, and such details.}

Address { like building ID, company's physical locations, zip code, address, name of the manager, etc., for each physical location etc}

Payroll {employee Id, payroll info-Basic, HRA, DA,TA}

Department{ DeptId, dname, Address}

Whenever a new hire takes place, data is added to relevant records on the relevant tables like payroll, employees table, department table etc. The HR department creates a new record and updates it to reflect the changes. When the enterprise needs to send a letter, it simply reads the employee's table to select the relevant personal details, or when the employee leaves their service.

Expected features

- Persisting employee, department, and Payroll information
- Modify the employee, department, and payroll information
- Delete employee, department
- Display the all employee works for a certain department
- List of all departments

Note:

Maintain the layered architecture

User interface will be mocked Mock Controller, with services and dao

Dao layer should be using JDBC to connect mysql database

ADVANCED JAVA

UNIT_013 : JAVA GENERICS

08 HRS

- Introduction to generics and type parameters
- Generic classes, methods, and interfaces
- Wildcards and bounded type parameters

UNIT_014 : JAVA COLLECTIONS FRAMEWORK

07 HRS

- Overview of the Collections Framework
- Working with lists, sets, and maps
- Customizing collections with comparators and comparables

UNIT_015 : MULTITHREADING AND CONCURRENCY

05 HRS

- Introduction to threads and thread synchronization
- Concurrency utilities (locks, semaphores, barriers)
- Executors and thread pools

UNIT_016 : JAVA I/O AND NIO

08 HRS

- Working with file I/O and streams
- Serialization and deserialization
- Introduction to NIO (New I/O) and non-blocking I/O

UNIT_017 : JAVA DATABASE CONNECTIVITY (JDBC)

09 HRS

- Introduction to JDBC and database connectivity
- Executing SQL queries and retrieving results
- Working with prepared statements and transactions

UNIT_018 : JAVA NETWORKING

08 HRS

- Socket programming with Java
- Building client-server applications
- Working with HTTP and network protocols

UNIT_019 : JAVA SERVLETS AND JAVASERVER PAGES (JSP)

07 HRS

- Overview of web development with Java
- Introduction to servlets and JSPs
- Building dynamic web applications with Java EE

UNIT_020 : JAVA MESSAGING SERVICES (JMS)

07 HRS

- Introduction to messaging systems
- Working with JMS and message queues
- Asynchronous communication and message-driven beans

LAB SET ADVANCED JAVA

LAB 1 : MULTITHREADING AND CONCURRENCY

- Create multiple threads in Java and explore thread synchronization.
- Implement synchronization mechanisms like locks, semaphores, or condition variables.
- Solve concurrency-related problems using techniques like thread pooling or message passing.

LAB 2 : JAVA GENERICS

- Define generic classes, interfaces, and methods in Java.
- Use bounded types and wildcards to enhance type safety and flexibility.
- Implement generic algorithms and data structures.

LAB 3 : JAVA I/O AND SERIALIZATION

- Perform file I/O operations using streams in Java.
- Serialize and deserialize objects using Java's serialization mechanism.
- Handle different types of I/O exceptions and error conditions.

LAB 4 : NETWORKING AND SOCKET PROGRAMMING

- Implement client-server communication using Java's networking APIs.
- Create TCP/IP and UDP-based client-server applications.
- Exchange data over the network using sockets and streams.

LAB 5 : JAVA DATABASE CONNECTIVITY (JDBC)

- Connect Java applications to relational databases using JDBC.
- Execute SQL queries and retrieve results in Java.
- Implement database transactions and handle exceptions.

LAB 6 : JAVA REFLECTION

- Use Java Reflection API to examine and modify the structure and behavior of classes at runtime.
- Access and invoke methods dynamically using reflection.
- Analyze and manipulate class metadata using reflection.

LAB 7 : JAVA SERVLETS AND JAVASERVER PAGES (JSP)

- Develop web applications using Java Servlets to handle HTTP requests and responses.
- Use JSP to generate dynamic web content and combine it with Java code.
- Implement session management and handle form submissions.

LAB 8 : JAVA PERSISTENCE API (JPA) AND HIBERNATE

- Use JPA and Hibernate for object-relational mapping (ORM).
- Perform CRUD operations on database entities using JPA annotations.
- Configure Hibernate and establish a connection to a relational database.

LAB 9 : JAVA MESSAGING SERVICE (JMS)

- Implement message-driven applications using the Java Messaging Service.
- Send and receive messages asynchronously using JMS.
- Handle message queues, topics, and message selectors.

LAB 10 : JAVA ENTERPRISE EDITION (JAVA EE) COMPONENTS

- Develop Java EE applications using components like EJB (Enterprise JavaBeans) and JMS.
- Implement business logic and expose it as web services using SOAP or REST.
- Deploy and test Java EE applications on application servers like Apache Tomcat or WildFly.

JAVA (WEB) PROGRAMMING (JSP/ SERVLET)

UNIT_001 : JAVA SERVLETS

07 HRS

- Introduction to Java Servlets
- Servlet lifecycle and request handling
- Handling form data and URL parameters
- Session management and cookies

UNIT_002 : JAVASERVER PAGES (JSP)

08 HRS

- Introduction to JavaServer Pages
- JSP lifecycle and scripting elements
- Expression Language (EL) and JSP standard actions
- JSP custom tags and tag libraries

UNIT_003 : JAVA DATABASE CONNECTIVITY (JDBC)

10 HRS

- Introduction to JDBC
- Establishing database connections
- Executing SQL statements and retrieving data
- Handling transactions and connection pooling

9 HRS

UNIT_004 : MODEL-VIEW-CONTROLLER (MVC) ARCHITECTURE

- Overview of MVC architecture
- Separation of concerns: models, views, and controllers
- Implementing MVC pattern using Servlets and JSP

UNIT_005 : DEPLOYMENT AND APPLICATION SERVERS

07 HRS

- Packaging and deploying web applications
- Overview of application servers (Tomcat, JBoss, etc.)
- Configuring and managing web applications in an application server environment

LAB SET SERVLET

LAB 1 :

Write a simple Servlet program to display welcome message.

LAB 2 :

- Write a program to change the background color selected by the user.

LAB 3 :

- Write a Servlet program to display “Hello World” in HTML/Text format.

LAB 4 :

Write a servlet program to display current date in server.

LAB 5 :

- Write a servlet program to display information about server.

LAB 6 :

Write servlet program to read init parameter values.

LAB 7 :

- Write a servlet program to submit information from html to servlet and display the message.

LAB 8 :

Write a servlet program to pass multiple values.

LAB 9 :

Write a Servlet program to demonstrate sendRedirect
loginForSendRedirect.jsp

LAB 10 :

- (a) Write a Servlet program to display session.
- (b) To develop a simple program to get session elements.

LAB 11 :

- (a) Write a program how to send cookie in servlets.
- (b) Write a program to read cookies from Servlets.

LAB 12 :

Write a program to demonstrate URL-Rewriting.

LAB 13 :

- (a) Write a program to demonstrate Hidden Form Fields.
- (b) To demonstrate a servlet program to show Multiple Hidden formFields.

LAB 14 :

Write a servlet program using RequestDispatchcher to forward the request to an Error page.

LAB 15 :

Write a servlet program to submit information from html to servlet and connection to the database.

LAB 16 :

Write a servlet program for handling database connections using SingleThreadModel.

LAB 17 :

To develop a mini project using servlet (ATM BANK).

LAB SET JSP

LAB 1 : INTRODUCTION TO JSP

- Set up the development environment with a servlet container (e.g., Apache Tomcat).
- Create a simple JSP file that displays "Hello, JSP!" in the browser.
- Execute the JSP file and verify the output.

LAB 2 : JSP BASICS

- Create a JSP page that includes an HTML form.
- Retrieve form data using request parameters in JSP.
- Display the submitted form data on the JSP page.

LAB 3 : JSP SCRIPTING ELEMENTS

- Use scriptlets to write Java code within a JSP page.
- Retrieve request parameters and perform server-side processing using scriptlets.
- Display the processed data on the JSP page.

LAB 4 : JSP EXPRESSION LANGUAGE (EL)

- Use EL expressions to access and display variables and attributes in a JSP page.
- Perform arithmetic calculations and conditional statements using EL.
- Use EL to access request parameters and display their values.

LAB 5 : JSP STANDARD ACTIONS

- Use JSP standard actions like `<jsp:include>` and `<jsp:forward>` to include and forward JSP pages.
- Use `<jsp:useBean>` to create and use JavaBeans in JSP.
- Use `<jsp:setProperty>` and `<jsp:getProperty>` to set and get JavaBean properties.

LAB 6 : JSP CUSTOM TAGS

- Create a custom tag handler class and define a custom tag in a JSP.
- Implement custom tag attributes and body content in the tag handler class.
- Use the custom tag in a JSP page and observe its behavior.

LAB 7 : JSP AND JDBC

- Set up a database connection using JDBC in a JSP page.
- Retrieve data from a database using JDBC queries in a JSP page.
- Display the retrieved data on the JSP page.

LAB 8 : JSP AND SESSION MANAGEMENT

- Use session attributes to store and retrieve user-specific data in a JSP page.
- Implement login/logout functionality using session management in JSP.
- Display user-specific data based on session attributes in the JSP page.

LAB 9 : JSP AND ERROR HANDLING

1. Handle exceptions and errors in JSP using the `<%@ page errorPage="..." %>` directive.
2. Create an error handling JSP page to display custom error messages.
3. Test the error handling mechanism by deliberately throwing exceptions in a JSP page.

LAB 10 : JSP AND INTERNATIONALIZATION

- Implement internationalization in JSP using resource bundles and `<fmt:message>` tag.
- Create multiple property files for different languages and configure them in the JSP.
- Display localized messages and labels in the JSP based on the user's preferred language.

ORM HIBERNATE

01 HRS

UNIT_006 : HIBERNATE SETUP AND BASIC CRUD OPERATIONS

- Set up a Hibernate project with the required dependencies.
- Create a Hibernate configuration file and establish a connection to the database.
- Create an entity class and map it to a database table.
- Perform basic CRUD (Create, Read, Update, Delete) operations using Hibernate APIs.

UNIT_007 : MAPPING ASSOCIATIONS AND RELATIONSHIPS

01 HRS

- Create entity classes for different entities with various relationships (One-to-One, One-to-Many, Many-to-One, Many-to-Many).
- Define appropriate mapping annotations for each relationship in the entity classes.
- Implement CRUD operations involving associated entities using Hibernate.

UNIT_008 : HIBERNATE QUERY LANGUAGE (HQL)

01 HRS

- Write HQL queries to retrieve data from the database.
- Use HQL for filtering, sorting, and pagination.
- Perform aggregate functions and joins in HQL queries.

01 HRS

UNIT_009 : CRITERIA QUERIES IN HIBERNATE

- Use Hibernate's Criteria API to perform queries without writing HQL.
- Build dynamic queries using criteria expressions and conditions.
- Execute criteria queries and process the results.

01 HRS

UNIT_010 : HIBERNATE CACHING AND PERFORMANCE OPTIMIZATION

- Enable and configure Hibernate's first-level and second-level caches.
- Monitor cache usage and analyze performance improvements.
- Implement cache eviction and invalidation strategies.

UNIT_011 : TRANSACTIONS AND CONCURRENCY CONTROL 01 HRS

- Implement transaction management in Hibernate using session transactions.
- Handle concurrency issues using optimistic and pessimistic locking mechanisms.
- Test transaction isolation levels and analyze their impact on data consistency.

UNIT_012 : HIBERNATE ANNOTATIONS AND XML MAPPING 01 HRS

- Use Hibernate annotations to define mappings instead of XML.
- Compare the benefits and drawbacks of annotation-based mapping.
- Convert existing XML mappings to annotations.

UNIT_013 : HIBERNATE AND SPRING INTEGRATION 01 HRS

- Integrate Hibernate with the Spring Framework.
- Configure Hibernate session factory and transaction management in Spring.
- Create a Spring-based application that utilizes Hibernate for data persistence.

01 HRS

UNIT_014 : ADVANCED HIBERNATE CONCEPTS

- Explore advanced Hibernate features such as caching strategies, lazy loading, and fetching strategies.
- Implement custom type mapping and converters in Hibernate.
- Use Hibernate Envers for auditing and versioning of entities.

01 HRS

UNIT_015 : HIBERNATE AND WEB APPLICATION DEVELOPMENT

- Build a web application using a Java web framework (e.g., Spring MVC) and Hibernate.
- Create controllers to handle user requests and interact with the Hibernate persistence layer.
- Implement data validation, form handling, and error handling in the web application.

LAB SET ORM HIBERNATE

LAB 1 :

1. Write a program to save user details username, password in the mysql using simple hibernate object
2. Write a program to save user details username, password in the mysql using simple hibernate object without using hibernate.cfg.xml

LAB 2 :

Write a program the user defined table name and column name.

LAB 3 :

Write a program to use with annotations Id, table, column, Generated Value, Transient and Lob in user table.

LAB 4 :

Write a program to use ValueType and Embedded object using nested classes scenario such as employee works for department

LAB 5 :

Write a program to use AttributeOverride and Embedded object keys using nested classes scenario such as employee works for department located in multiple address

LAB 6 :

Write a program to showcase association company linking with multiple address

LAB 7 :

Write a program to showcase fetch proxy object, Eager and Lazy object

LAB 8 :

Write a program association from employee to department one -to-one mapping

LAB 9 :

Write a program association from one employee works for more than one department showcase one -to-many mapping

LAB 10 :

- Write a program association from many to many using employee and dept relationship

LAB 11 :

Write a program inherit from single table person_info to employee and student(default inheritance)

LAB 12 :

- Write a program polymorphic inheritance from base class person to employee and student as concrete implementations using MappedSuperClass annotations

LAB 13 :

Write a program polymorphic inheritance from base class person to employee and student as concrete implementations using SingleTableStrategy annotations

LAB 14 :

- Write a program polymorphic inheritance from base class person to employee and student as concrete implementations using Table Per class Strategy annotations

LAB 15 :

Write a program polymorphic inheritance from base class person to employee and student as concrete implementations using JOIN Strategy annotations

LAB 16 :

Write a program to develop simple crud operation

LAB 17 :

Write a program to showcase Transient, Persistent and Detached Object

LAB 18 :

Write a program to demonstrate to load user details using HQL

LAB 19 :

Write a program to showcase the pagination,update and delete

LAB 20 :

Write a program to showcase the Named Query

LAB 21 :

Write a program to demonstrate Criteria API

LAB 22 :

Write a program to demonstrate 2ND level cache in hibernate

SPRING AND SPRING BOOT

01 HRS

UNIT_016 : INTRODUCTION TO SPRING FRAMEWORK

- Overview of Spring framework and its features
- Dependency Injection (DI) and Inversion of Control (IoC) principles
- Setting up a Spring development environment

UNIT_017 : SPRING CORE

01 HRS

- Configuring Spring beans using XML and Java-based configuration
- Understanding bean scopes and lifecycle callbacks
- Injecting dependencies using constructor and setter injection
- Working with annotations for bean configuration

UNIT_018 : SPRING MVC

01 HRS

- Introduction to Spring MVC architecture
- Handling HTTP requests and responses
- Implementing controllers, views, and request mappings
- Handling form submissions and validation

UNIT_019 : SPRING DATA ACCESS

01 HRS

- Integrating Spring with JDBC and JPA
- Implementing data access operations using Spring's JDBC template and JPA repositories
- Configuring transaction management in Spring
- Using Spring's declarative transaction support

UNIT_020 : SPRING SECURITY

01 HRS

- Overview of Spring Security framework
- Implementing authentication and authorization
- Configuring role-based access control
- Securing web applications using Spring Security

UNIT_021 : SPRING AOP

01 HRS

- Understanding Aspect-Oriented Programming (AOP) concepts
- Creating and applying aspects using Spring AOP
- Implementing cross-cutting concerns such as logging and transaction management
- Configuring AOP with XML and annotations

UNIT_022 : INTRODUCTION TO SPRING BOOT

01 HRS

- Overview of Spring Boot and its features
- Creating Spring Boot applications using the Spring Initializr
- Auto-configuration and opinionated defaults in Spring Boot
- Packaging and deploying Spring Boot applications

UNIT_023 : SPRING BOOT WEB DEVELOPMENT

01 HRS

- Building RESTful APIs using Spring Boot
- Handling JSON/XML serialization and deserialization
- Implementing CRUD operations with Spring Data REST
- Testing Spring Boot web applications

UNIT_024 : SPRING BOOT DATA ACCESS

01 HRS

- Working with relational databases using Spring Boot and Spring Data JPA
- Using Spring Data repositories for data access
- Implementing pagination and sorting
- Handling database migrations with Flyway or Liquibase

UNIT_025 : SPRING BOOT SECURITY

01 HRS

- Securing Spring Boot applications with Spring Security
- Implementing authentication and authorization mechanisms
- Configuring JWT (JSON Web Tokens) for stateless authentication
- Role-based access control in Spring Boot

UNIT_026 : SPRING BOOT ACTUATOR AND MONITORING **01 HRS**

- Monitoring and managing Spring Boot applications using Actuator
- Customizing Actuator endpoints and health checks
- Integrating with monitoring tools such as Prometheus and Grafana
- Logging and debugging Spring Boot applications

UNIT_027 : SPRING BOOT TESTING **01 HRS**

- Writing unit tests for Spring Boot applications
- Using Spring Test and Mockito for testing
- Integration testing with embedded or in-memory databases
- Testing RESTful APIs with Spring MVC Test

LAB SET SPRING AND SPRING BOOT

LAB 1 :

Write a application program to instantiate bean object using bean factory

LAB 2 :

Write a application program to instantiate bean object using application context.

LAB 3 :

Write application of bean objects from context by setting the attributes

LAB 4 :

Write application constructor injection to bean object indexing, type in spring.xml as an argument to build employee information

LAB 5 :

Write a application program to instantiate bean object using bean factory

LAB 6 :

Write a application program to instantiate bean object using application context.

LAB 7 :

Write application of bean objects from context by setting the attributes

LAB 8 :

Write application constructor injection to bean object indexing, type in spring.xml as an argument to build employee information

LAB 9 :

Write application injecting object build the employee details

LAB 10 :

Demonstrate with employee details using inner bean, alias and idref

LAB 11 :

Demonstrate with company details with initialization as collections

LAB 12 :

Demonstrate the autowiring of bean objects of employee by name, type and constructor

LAB 13 :

Demonstrate bean scopes singleton, prototype using employee object

LAB 14 :

WAP to implement ApplicationContextAware and BeanNameAware

LAB 15 :

WAP to implement ApplicationContextAware and BeanNameAware

LAB 12 :

WAP to demonstrate call back life cycle bean methods

LAB 13:

WAP reading from properties file using PropertyPlaceholder



THANK YOU

VISIT US @



www.uplsnipe.com

DOWNLOAD THE APP

