UNLIMITED POWERFUL LEARNING

UPL PROSPECTOUS

early break even | best experience

2023

@unlimited powerfull learning

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

UPL
UNLIMITED POWER FULL LEARNING

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

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

UPL
UNLIMITED POWER FULL LEARNING

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

AUTOMATION TESTING WITH JAVA

Automation testing with Java involves using Java programming language and various frameworks and tools to automate the testing process. Java provides a robust platform for automation testing due to its wide adoption, extensive libraries, and strong community support. Here's an overview of the key components and steps involved in automation testing with Java:

CORE JAVA

UNIT_001 : 05 HRS

The Project, Domain, Platform, Product, Technology, Java Tech Stack, Process, Development Environment, Java Introduct

UNIT_002 : 04 HRS

Java Fundamentals, Tokens, Conditional Statements, Looping Constructs

UNIT_003 : 05 HRS

Java Fundamentals, Tokens, Conditional Statements, Looping Constructs

UNIT_004: 05 HRS

Arrays, Strings, Exception Handling-Try, Catch, Finally, Custom Exception, Checked And Unchecked Exception

UNIT_005 : 08 HRS

Multithreading, Life Cycle, Synchronized Block, And Method, Wait, Notify, notify all, Concurrency -ThreadPool, ThreadPool Types, FixedThreadPool, CachedThreadPool, ScheduledThreadPool, SingleThreadExecutor, Threadpool size,ThreadLocal, Lock

UNIT_006 : 08 HRS

IOStreams, Inputstream, Outputstreams, Character Vs Byte Streams, Node And Filter Streams, Collection, Collection Framework, Collection Interface, Bulk Operations , Internals Of List, Set And Map, Arraylist, Hashset, Hashmap AndImportant Concepts Of Implementations Of Collection Frameworks

UNIT_007 : 016 HRS

Annotations -Build-In Annotations, Custom Annotations, Market, Singlevalue And Multivalue Annotations, Regular Expressions

UNIT_008 : 03 HRS

JAVA 1.8 and 11 features, LAMBDA Expressions, Functional Interface, Streams API, JDBC Programming.

UNIT_009 : POPULAR FRAMEWORKS FOR JAVA INCLUDE: 10 HRS

1. **Selenium WebDriver:** A widely used framework for automating web applications.

- Introduction to Selenium
- WebDriver Basics
- Locators and Element Identification
- Interacting with Web Elements
- Navigation and Browser Manipulation
- Synchronization and Waits
- Handling Frames, Alerts, and Pop-ups
- Test Design Techniques
- Handling Test Failures and Reporting
- Advanced WebDriver Techniques
- Cross-Browser Testing

2. **Appium:** An open-source framework for automating mobile applications.

- Introduction to Appium:
- Appium Basics:
- Mobile Application Platforms:
- Appium Desired Capabilities:
- Locating and Interacting with Elements:
- Mobile Gestures and Touch Actions:
- Mobile Application Contexts:
- Test Design Techniques:
- Mobile App Testing Considerations:
- Appium Advanced Topics:

3. JUnit: A unit testing framework for writing and executing test cases.

- Introduction to JUnit:
- Writing JUnit Tests:
- Assertions:
- Test Fixtures and Lifecycle:
- Parameterized Tests:
- Test Suites and Categories:
- Test Runners and Configuration:
- Test Annotations and Metadata:
- Test Execution and Reporting:
- Advanced Testing Techniques:
- Integration with Build Tools and CI/CD:
- Best Practices and Test Design Patterns:

2. TestNG: An advanced testing framework that offers more features than JUnit.

- Introduction to TestNG:
- TestNG Assertions:
- TestNG Basics:
- Test Configuration and Dependencies:
- TestNG XML Configuration:
- TestNG Reporters and Listeners:
- TestNG Data Providers:
- TestNG Annotations and Test Lifecycle:
- Parallel Test Execution:
- TestNG TestNG Extensions and Plugins:
- TestNG Integration with Build Tools and CI/CD:
- Best Practices and Test Design Patterns:

UNIT_010: TEST ENVIRONMENT SETUP: 06 HRS

Install Java Development Kit (JDK) and set up the Java environment.
Configure Integrated Development Environment (IDE) such as Eclipse or IntelliJ IDEA.
Set up the necessary dependencies and libraries for the chosen automation framework.

UNIT_011: TEST SCRIPT DEVELOPMENT: 07 HRS

Write test scripts using Java programming language and the chosen automation framework.
Use the framework's APIs and methods to interact with the application under test.
Utilize element locators to identify and interact with web elements (e.g., XPath, CSS selectors).

UNIT_012: TEST DATA MANAGEMENT: 06 HRS

Prepare test data required for test cases.
Implement techniques to manage test data such as data-driven testing using external files (Excel, CSV) or databases.

UNIT_013 : TEST CASE EXECUTION: 06 HRS

Execute test scripts either locally or on a remote testing environment.
Monitor and log test execution results.
Capture screenshots or videos for failed test cases for further analysis.

UNIT_014 : TEST REPORTING AND ANALYSIS: 04 HRS

Generate test reports to provide insights into test execution status and results.
Use reporting frameworks like TestNG or JUnit to generate comprehensive test reports.
Analyze test results and identify failed test cases or potential issues.

UNIT_015 : CONTINUOUS INTEGRATION AND BUILD TOOLS: 07 HRS

Integrate automation tests with Continuous Integration (CI) tools like Jenkins or Bamboo.
Set up a build script (e.g., using Maven or Gradle) to automate the execution of test scripts as part of the build process
Configure the CI tool to generate reports and notifications based on test execution results

UNIT_016 : TEST MAINTENANCE AND VERSION CONTROL: 06 HRS

Regularly update and maintain test scripts to accommodate application changes or new features.
Use version control systems like Git or Subversion to manage and track changes in test scripts.

UNIT_017 : PARALLEL TEST EXECUTION: 06 HRS

Utilize parallel test execution to save time and increase test coverage.
Run multiple test scripts simultaneously to achieve faster test execution

UNIT_018 : INTEGRATION WITH TEST MANAGEMENT TOOLS: 06 HRS

Integrate automation tests with test management tools like JIRA, TestRail, or Zephyr for test case management, tracking, and reporting.

UNIT_019 :

CAPSTONE PROJECT

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 Inhertance 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 inherithe ted 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 concreate 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
- Listof 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

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