



Java

Java from Basic to
Professional [128 Hrs]

Instructor: Pondit Team

E-Mail:

Phone:

Overview

আইটি ইন্ডাস্ট্রির জন্য দক্ষ জনশক্তি তৈরিতে পন্ডিট স্পন্সরশীপ প্রদান করছে। যার পরিপ্রেক্ষিতে ইউনিভার্সিটির ছাত্রছাত্রীরা একটি পরীক্ষায় ভালো রেজাল্টের মাধ্যমে ১০০% পর্যন্ত ফ্রি কোর্স করার সুযোগ পাচ্ছেন।

পরীক্ষার মাধ্যমে আপনি ১০০% পর্যন্ত স্পন্সরশীপ পাবেন তবে আপনাকে অবশ্যই পরীক্ষায় ৮০% মার্ক্স পেতে হবে। ১০০% স্পন্সরশীপ না পেলে পরীক্ষার রেজাল্টের উপর ভিত্তি করে ৭৫% অথবা ৫০% ডিসকাউন্ট পাবেন।

কাদের জন্যে কোর্স:

- এই কোর্স মূলত বিশ্ববিদ্যালয় এর তৃতীয় বর্ষের ছাত্র-ছাত্রীদের জন্যে যারা এই কোর্সটি করার পরে এবং তাদের ফাইনাল সেমিস্টার শেষ হওয়ার সাথে সাথেই যারা জবে জয়েন করে চায়।
- প্রথম এবং দ্বিতীয় বর্ষের ছাত্র-ছাত্রীরা এখন থেকেই অবজেক্ট ওরিয়েন্টেড কনসেপ্ট শুরু করতে চায় এবং ভবিষ্যতে জবের জন্য নিজেদেরকে তৈরি রাখতে চায়।

কেন করবেন এই কোর্সঃ

- আইটি কোম্পানীতে ভালো বেতনে জব করার সুযোগ আছে।
- অনলাইন মার্কেটপ্লেসে প্রচুর জাভা প্রোগ্রামারদের চাহিদা রয়েছে।
- দক্ষ ট্রেনার দিয়ে কোর্স পরিচালনা করানো হয়।

Audience

- ছাত্র/ছাত্রী
- চাকুরীজীবী

Evaluation

Curriculum

1. Learning Object Oriented Programming Concepts with Java

- 1.1. Installing and configuring Java 11
- 1.2. Practicing code without an IDE
- 1.3. Installing and IDE and using it
- 1.4. Data types, variables and assignments

Prerequisites

প্রোগ্রামিং এর ধারণা থাকতে হবে

Course Information

- Duration : 128 Hours
- Day : রবিবার, মঙ্গলবার এবং বৃহস্পতিবার
- Time : রাত ৮.৩০ টা থেকে ১০.৩০ টা পর্যন্ত
- Reg. Start : May 1, 2022
- Reg. End : Aug 13, 2022
- Class Start : Sep 11, 2022

- 1.5. Unchecked Exception / Runtime Exception
- 1.6. Creating own exception classes
- 1.7. Knowing the classes & interfaces of collection framework
- 1.8. Using the most common collection frameworks
- 1.9. Knowing the best practices and use cases of several collection framework classes
- 1.10. Conceptual understanding and knowing the necessity of JDBC API
- 1.11. Checked Exception / Compile time exception
- 1.12. Installing a database server
- 1.13. Downloading and using driver class for the database server
- 1.14. Writing and executing database SQL queries with the JDBC API
- 1.15. Working with result set
- 1.16. Developing small project
- 1.17. Method Overloading / Static binding / Compile time polymorphism
- 1.18. Inputs and outputs
- 1.19. Conditions
- 1.20. Iterations
- 1.21. Arrays
- 1.22. Methods
- 1.23. Theory of thinking in Object Oriented approach
- 1.24. Basic discussion of the OOP conceptual terminologies
- 1.25. Inheritance (IS-A relationship) & Aggregation (HAS-A relationship)
- 1.26. Encapsulation
- 1.27. Operators and keywords
- 1.28. Method Overriding / Dynamic Binding / Runtime polymorphism
- 1.29. Overriding cases while implementing interfaces
- 1.30. Understanding Abstraction
- 1.31. Use cases of Abstraction
- 1.32. String Literal vs using new keyword
- 1.33. Immutable Strings
- 1.34. Using the methods of String class
- 1.35. String Buffer & String Builder
- 1.36. Exception vs Error
- 1.37. Introduction to Servlet and it's APIs
- 1.38. Simple Servlet HelloWorld program
- 1.39. Load on Startup
- 1.40. ServletRequest interface
- 1.41. Understanding different request methods and implementations in Servlet
- 1.42. RequestDispatcher and SendRedirect
- 1.43. Cookies and Sessions in Servlet
- 1.44. Servlet with JDBC
- 1.45. Simple Login and Logout

- 1.46. ServletFilters
- 1.47. Servlet Events

2. Spring By Practical Examples

- 2.1. Installing and configuring Java 11
- 2.2. Installing Git (Version Control tool)
- 2.3. Installing IntelliJ Idea (IDE)
- 2.4. Introducing common git commands (if needed)
- 2.5. Introduction to Maven
- 2.6. Introduction to Servlet and Servlet Container
- 2.7. Configuring Servlet Application
- 2.8. Implementing HttpServlet and working with different HTTP Methods
- 2.9. View rendering with Servlet and JSP
- 2.10. Simple form data handling with Servlet and JSP
- 2.11. Downloading application server (Tomcat)
- 2.12. Configuring tomcat server with IntelliJ Idea
- 2.13. MVC architecture in Spring
- 2.14. Introduction to Servlet & Dispatcher Servlet Configuration
- 2.15. What is a Bean, how to work with it
- 2.16. View Resolver Bean definition, working with different view resolvers.
- 2.17. @Component and @Configuration annotations
- 2.18. Introduction to Taglibs
- 2.19. Form Taglib, core Taglib, and other taglib examples
- 2.20. Accessing the form data and rendering those data to page
- 2.21. Storing the form data in a list based storage, rendering them in the page
- 2.22. Understanding Java validation framework and it's needs.
- 2.23. Implementing validation for the form data processing.
- 2.24. Understanding Client side business logic rendering and manipulation with JavaScript.
- 2.25. Practicing JavaScript with real world examples.
- 2.26. Converting javascript examples with JQuery.
- 2.27. Handling different types of events and event listeners.
- 2.28. Learning Simple Ajax requests.
- 2.29. Installing Postgresql locally
- 2.30. Integrating JDBC Template to make a CRUD (Connecting with a PSql DB)
- 2.31. Opening a Heroku account and creating a Postgresql DB in Heroku
- 2.32. Setting up different profiles (Dev, QA, Prod etc)
- 2.33. Integrating and configuring Hibernate Session Factory, Transaction Managers etc.
- 2.34. Performing the CRUD with Hibernate

- 2.35. Using Criteria API
- 2.36. Implementing Global Exception Handler to prettify and well format the exception messages.
- 2.37. Configuration to render Static contents
- 2.38. Adding CSS, JS, images and other static contents
- 2.39. Collecting data from Form and Rendering them into a page
- 2.40. Integrating and Configuring Spring security
- 2.41. Discussing Role Based Access Control feature (RBAC) in Spring framework
- 2.42. Configuring Login, Log Out, default error pages
- 2.43. Integrating custom filter and custom interceptor with spring security
- 2.44. Defining Custom Authentication Provider
- 2.45. Defining Custom Authentication Success handler, Custom Logout handler
- 2.46. Using Password encryption algorithms
- 2.47. Role Based Menu access handling
- 2.48. Configuring heroku deployment environment
- 2.49. Grooming, development and evaluation
- 2.50. Understanding GET, POST, PUT, DELETE, FETCH methods
- 2.51. Understanding HATEOS
- 2.52. Understanding the benefits and drawbacks of making API call over browser loading
- 2.53. Developing APIs and testing with Postman
- 2.54. Making HTTP requests with Ajax and/or fetch API
- 2.55. Understanding the differences between Web and API security
- 2.56. Different approaches to secure an API
- 2.57. Discussing the JWT and OAuth 2 Security principles
- 2.58. Redesigning and redeveloping old project concept to api driven single page application