Course Information

Java is a popular programming language, created in 1995. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.) It is open-source, free, secure, fast and powerful and it is one of the most popular programming language in the world. It has a huge community support.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection

Following professionals can go for it:

- Fresher's
- IT Experts

This course will cover 12 months of training in which 80% of the training will be practical based with regular assignments and after completion of the training, a project will be given to the student and their evaluation will be based on their projects. Also regular tests and mock sessions on technical as well as on HR rounds will be a part of the curriculum. This course also includes soft skill development which will help students to perform better in interview.

Eligibility: Technical graduate having fundamental knowledge of any programming languages like C, C++.

Lecture Duration: 12 months

Placement: 100% Placement Assistance

Job Profile: Java developer

Contents

Java

- Object: The Cosmic Superclass
- The Class Class
- Reflection
- Design Hints for Inheritance

Interfaces and Inner Classes

- Interfaces
- Object Cloning
- Inner Classes
- Proxies

Graphics Programming

- Introduction to Swing
- Creating a Frame
- Frame Positioning
- Displaying Information in a Panel
- 2D Shapes
- Colors
- Text and Fonts
- Images

Event Handling

- Basics of Event Handling
- The AWT Event Hierarchy
- Semantic and Low-Level Events in the AWT
- Low-Level Event Types
- Actions
- Multicasting
- The Event Queue

User Interface Components with Swing

- The Model-View-Controller Design Pattern
- An Introduction to Layout Management
- Text Input
- Making Choices
- Menus
- Sophisticated Layout Management
- Dialog Boxes

Applets

- Applet Basics
- The Applet HTML Tags and Attributes
- Multimedia

The Applet Context
JAR Files

Exceptions and Debugging

Dealing with Errors
Catching Exceptions
Some Tips on Using Exceptions
Debugging Techniques
Using a Debugger

Streams and Files

Streams
The Complete Stream Zoo
ZIP File Streams
Putting Streams to Use
Object Streams
File Management

Advanced Java

Multithreading

What Are Threads?
Interrupting Threads
Thread Properties
Thread Priorities
Selfish Threads
Synchronization
Deadlocks
User Interface Programming with Threads
Using Pipes for Communication between Threads

Collections

Collection Interfaces
Concrete Collections
The Collections Framework
Algorithms
Legacy Collections

Database Connectivity: JDBC

The Design of JDBC
The Structured Query Language
Installing JDBC
Basic JDBC Programming Concepts
Executing Queries

Scrollable and Updatable Result Sets
Metadata
Transactions
Advanced Connection Management
Java IDL and CORBA

Advanced Swings

Lists
Trees
Tables
Styled Text Components
Component Organizers

Advanced Swings

Lists
Trees
Tables
Styled Text Components
Component Organizers

Advanced AWT

The Rendering Pipeline
Shapes
Areas
Strokes
Paint
Coordinate transformations
Clipping
Transparency and Composition
Rendering Hints
Reading and Writing Images
Image Manipulation
Printing
The Clipboard
Drag and Drop

J2EE

Overview
Distributed Multitiered Applications
J2EE Containers
Web Services Support
Packaging Applications
Development Roles
J2EE APIs

The Example JSP Document
Creating a JSP Document
Identifying the JSP Document to the Container
JavaServer Pages Standard Tag Library
The Example JSP Pages
Using JSTL
Core Tag Library
XML Tag Library

Internationalization Tag Library
SQL Tag Library
Functions
Further Information
Custom Tags in JSP Pages
What Is a Custom Tag?
The Example JSP Pages
Types of Tags

Encapsulating Reusable Content Using Tag Files
Tag Library Descriptors
Programming Simple Tag Handlers

Scripting in JSP Pages
The Example JSP Pages
Using Scripting
Disabling Scripting
Declarations
Scriptlets

Expressions
Programming Tags That Accept Scripting Elements
JavaServer Faces Technology
JavaServer Faces Technology Benefits
What Is a JavaServer Faces Application?
Framework Roles

A Simple JavaServer Faces Application
User Interface Component Model
Navigation Model
Backing Bean Management
How the Pieces Fit Together
The Life Cycle of a JavaServer Faces Page
Further Information

Understanding Login Authentication

Installing and Configuring SSL Support

XML and Web Services Security
EJB-Tier Security
Application Client-Tier Security
EIS-Tier Security
Propagating Security Identity
What Is Java Authorization Contract for Containers?
Further Information
The Java Message Service API
Overview
Basic JMS API Concepts
The JMS API Programming Model

Writing Simple JMS Client Applications

Creating Robust JMS Applications
Using the JMS API in a J2EE Application
Further Information
J2EE Examples Using the JMS API
A J2EE Application That Uses the JMS API with a Session Bean
A J2EE Application That Uses the JMS API with an Entity Bean
An Application Example That Consumes Messages from a Remote J2EE Server

An Application Example That Deploys aMessage-Driven Bean on Two J2EE Servers
Enterprise Beans
Application Client
Web Client
Internationalization
Building, Packaging, Deploying, and Running the Application
Running the Clients

Spring Framework

Introduction to Spring
Steps to use Spring Framework in applications
Understanding IOC and Dependency Injection
Understanding the bean life-cycle – Auto wiring and bean scopes
Annotation-based dependency injection
Adding behaviour to an application using aspects – AOP
Creating and applying aspects

Introducing data access with Spring – JDBC through spring
Transactions in a Spring environment
Getting started with Hibernate in a Spring environment
Working with Spring MVC
Spring MVC Form Handling
Creating Views in Spring MVC

Hibernate

Introduction
Architecture
Hibernate with annotation
Web Application using hibernate
Generator Classes in Hibernate
Inheritance mapping
Table per Hierarchy
Table per Concrete
Collection in hibernate
Hibernate Query language
Hibernate Criteria Query Language
Caching in Hibernate
First level Cache
Second Level Cache

Struts

Introduction
Architecture
Struts Configuration File
Struts.xml
Multi configuration
Multi namespace
Struts Validation
Custom
Bundled
Ajax
Struts Interface
Servlet Action Context
Session Ae are
Servlet Context Aware
Hibernate With Struts
Struts Date Time Picker
Registration Example
Login Example

Fetch all Record of a table

Web Development

HTML & CSS Overview

Introduction
HTML Basics
HTML Elements
HTML Attributes
HTML Styles
HTML Forms
HTML Form Elements
HTML Input Element Types
HTML Input Attributes
HTML File Paths
Script tag and its uses
HTML & XHTML
CSS Introduction
CSS Syntax
CSS Selectors
CSS Styling

Javascript Primer

Introduction to Javascript
Javascript Statements
Javascript Keywords
Javascript Functions
Javascript Programs
Javascript Operators
Function Parameters
Function Return Values
Javascript Data Types
Primitive Types

Working with Objects

Object Overview
Object Oriented Programming
Object creation
Adding Properties to Objects
Adding Methods to Objects
Javascript Conditional Statements
Javascript Loops & Iteration
Enumerating properties
Callbacks
JSON

Environmental setup

MVC Architecture

Model-View-Controller explained

Why MVC matters

MVC - the AngularJS way

First Application

Directives

Introduction to Directives

Directive lifecycle

Using AngularJS built-in directives

Binding controls to data

Matching directives

Creating a custom directive

Expressions

Controllers

Role of a Controller

Attaching properties and functions to scope

Nested Controllers

Using filters in Controllers

Controllers in External Files

Controllers & Modules

Filters

Built-in filters

Using AngularJS filters

Creating custom filters

Tables

HTML DOM

Modules

Introduction to AngularJS Modules

Bootstrapping

Forms

Working with Angular Forms

Model binding

Form controller

Validating Angular Forms

Form events

Updating models with a twist

$error object

Scope

What is scope

Scope lifecycle

Two way data binding
Scope inheritance
Scope & controllers
Scope & directives
$apply and $watch
Rootscope
Scope broadcasting

Dependency Injection & Services

What is Dependency Injection
Using Dependency Injection
What are services
Creating services
Factory, Service & Provider
Using AngularJS built in services

Single Page Application (SPA)

What is SPA
Pros & Cons of SPA
Installing the ngRoute module
Configure routes
Passing parameters
Changing location
Resolving promises
Create a Single Page Application

Angular-X

Getting Started

1. Course Introduction.
2. What is Angular?
3. Angular vs Angular 2 vs Angular 4+
4. Project Setup and First App.
5. Editing the First App.
6. The Course Structure.
7. What is TypeScript.
8. A Basic Project Setup using Bootstrap for Styling

The Basics

How an Angular App gets Loaded and Started

Components

Using Custom Components
Creating Components with the CLI & Nesting Components
Working with Component Templates
Working with Component Style

Assignment 1: Practicing Components

What is Databinding

String Interpolation

- Property Binding
- Property Binding vs String Interpolation
- Event Binding
- Bindable Properties and Events
- Passing and Using Data with Event Binding
- Two-Way-Databinding

Important: FormsModule is required for Two-Way-Binding

- Combining all Forms of Databinding

Assignment 2: Practicing Databinding

- Understanding Directives
- Using ngIf to Output Data Conditionally
- Enhancing ngIf with an Else Condition.

Components & Databinding Deep Dive

- Splitting Apps into Components
- Property & Event Binding Overview
- Binding to Custom Properties
- Assigning an Alias to Custom Properties
- Binding to Custom Events
- Assigning an Alias to Custom Events
- Custom Property and Event Binding Summary
- Understanding View Encapsulation
- More on View Encapsulation
- Using Local References in Templates.
- Getting Access to the Template & DOM with @ViewChild.
- Understanding the Component Lifecycle

Lifecycle Hooks.

Assignment 3: Practicing Property & Event Binding and View Encapsulation

Directives Deep Dive

- ngFor and ngIf
- ngClass and ngStyle
- How to create a Basic Directive
- Using Services & Dependency Injection
- Why would you Need Services?
- Creating a Logging Service
- Injecting the Logging Service into Components
- Creating a Data Service
- Understanding the Hierarchical Injector
- How many Instances of Service Should It Be?
- Injecting Services into Services
- Using Services for Cross-Component Communication

Assignment 4: Practicing Services

Changing Pages with Routing

- Why do we need a Router?
- Understanding the Example Project.
- Setting up and Loading Routes.
- Navigating with Router Links.
- Understanding Navigation Paths.

  - Styling Active Router Links.

- Navigating Programmatically.
  - Using Relative Paths in Programmatic Navigation
  - Passing Parameters to Routes
  - Fetching Route Parameters
  - Fetching Route Parameters Reactively
  - An Important Note about Route Observables
  - Passing Query Parameters and Fragments
  - Retrieving Query Parameters and Fragments
  - Setting up Child (Nested) Routes
  - Using Query Parameters - Practice
  - Configuring the Handling of Query Parameters
  - Redirecting and Wildcard Routes
  - Important: Redirection Path Matching
  - An Introduction to Guards
  - Protecting Routes with canActivate
  - Protecting Child (Nested) Routes with canActivateChild
  - Using a Fake Auth Service

- Controlling Navigation with canDeactivate.
  - Handling Forms in Angular Apps
  - Template-Driven (TD) vs Reactive Approach
  - TD Forms
- Assignment 5: Practicing Template-Driven Forms.
  - Introduction to the Reactive Approach
  - Reactive Forms
  - Assignment 6: Practicing Reactive Forms.
  - Using Pipes to Transform Output
  - Introduction & Why Pipes are Useful
  - Using Pipes
  - Making Http Requests
  - Introduction & How Http Requests Work in SPAs
  - Sending Requests (Example: POST Request)
  - Adjusting Request Headers
  - Sending GET Requests
  - Sending a PUT Request
  - Transform Responses Easily with Observable Operators (map())
  - Using the Returned Data
  - Catching Http Errors
  - Using the "async" Pipe with Http Requests.

- Authentication & Route Protection in Angular Apps
  - How Authentication Works in Single-Page-Applications.
  - Creating a Signup Page and Route
  - Setting up the Firebase SDK
  - Signing Users Up
  - Signin Users In

Requiring a Token (on the Backend)
Sending the Token
Checking and Using Authentication Status
Adding a Logout Button
Route Protection and Redirection.
Wrap Up
The HttpClient (ANGULAR 5 Addition Bonus SECTION)
Request Configuration and Response.
Requesting Events.
Setting Headers.
Interceptors.

SQL - Structure Query Language

RDBMS - An Introduction
Database
Relational Database Systems
Working with the Book's Sample Database
SQL: A Relational Database Language
Normal Forms
Entity-Relationship Model
Syntax Conventions

Foundations of T-SQL

A Short History of T-SQL
Imperative vs. Declarative Languages
SQL Basics
Statements
Databases
Transaction Logs
Schemas
Tables
Views
Indexes
Stored Procedures
User-Defined Functions
SQL CLR Assemblies
Elements of Style
Whitespace
Naming Conventions
One Entry, One Exit
Defensive Coding
SQL-92 Syntax Outer Joins
The SELECT * Statement
Variable Initialization

Procedural Code and CASE Expressions

Three-Valued Logic

Control-of-Flow Statements
The BEGIN and END Keywords
The IF...ELSE Statement
The WHILE, BREAK, and CONTINUE Statements
The GOTO Statement
The WAITFOR Statement
The RETURN Statement
The TRY...CATCH Statement
The CASE Expression
The Simple CASE Expression
The Searched CASE Expression
CASE and Pivot Tables
COALESCE and NULLIF
Cursors

User-Defined Functions

Scalar Functions
Recursion in Scalar User-Defined Functions
Procedural Code in User-Defined Functions
Multistatement Table-Valued Functions
Inline Table-Valued Functions
Restrictions on User-Defined Functions
Nondeterministic Functions
State of the Database

Stored Procedures

Introducing Stored Procedures
Calling Stored Procedures
Managing Stored Procedures
Stored Procedures in Action
Recursion in Stored Procedures
Table-Valued Parameters
Temporary Stored Procedures
Recompilation and Caching
Stored Procedure Statistics
Parameter Sniffing
Recompilation

Triggers

DML Triggers
When to Use DML Triggers
Auditing with DML Triggers
Nested and Recursive Triggers
The UPDATE and COLUMNS_UPDATED Functions
Triggers on Views
DDL Triggers
Logon Triggers

Common Table Expressions and Windowing Functions

Common Table Expressions
Multiple Common Table Expressions
Recursive Common Table Expressions
Windowing Functions
The ROW_NUMBER Function
The RANK and DENSE_RANK Functions
The NTILE Function
Aggregate Functions and OVER

XML

Legacy XML
OPENXML
OPENXML Result Formats
FOR XML Clause
FOR XML RAW
FOR XML AUTO
FOR XML EXPLICIT
FOR XML PATH
The xml Data Type
Untyped xml
Typed xml
The xml Data Type Methods
The query Method
The value Method
The exist Method
The nodes Method
The modify Method
XML Indexes
XSL Transformations

XQuery and XPath

XPath and FOR XML PATH
XPath Attributes
Columns Without Names and Wildcards
Element Grouping
The data Function
XPath and NULL
The WITH XMLNAMESPACES Clause
Node Tests
XQuery and the xml Data Type
Expressions and Sequences
The query Method
Location Paths
Node Tests
Namespaces
Axis Specifiers

Dynamic XML Construction
XQuery Comments
Data Types
Predicates
Conditional Expressions (if...then...else)
Arithmetic Expressions
XQuery Functions
Constructors and Casting
FLWOR Expressions

Catalog Views and Dynamic Management Views

Catalog Views
Table and Column Metadata
Index Metadata
Querying Permissions
Dynamic Management Views and Functions
Session Information
Connection Information
Currently Executing SQL
Tempdb Space
Server Resources
Unused Indexes
INFORMATION_SCHEMA Views

SQL CLR Programming

The Old Way
The SQL CLR Way
SQL CLR Assemblies
User-Defined Functions
Stored Procedures
User-Defined Aggregates
Creating a Simple UDA
Creating an Advanced UDA
SQL CLR User-Defined Types

New T-SQL Features

Set Operators
The OUTPUT Clause
The TOP Keyword
CROSS APPLY and OUTER APPLY
The TABLESAMPLE Clause
The NEWSEQUENTIALID Function
Date and Time Functions
The max Data Types
Synonyms
FILESTREAM Support
Enabling FILESTREAM Support
Creating FILESTREAM Filegroups
FILESTREAM-Enabling Tables

Accessing FILESTREAM Data

Error Handling and Dynamic SQL

Error Handling

Legacy Error Handling

Try...Catch Exception Handling

The RAISERROR Statement

Debugging Tools

PRINT Statement Debugging

Trace Flags

SSMS Integrated Debugger

Visual Studio T-SQL Debugger

Dynamic SQL

The EXECUTE Statement

SQL Injection and Dynamic SQL

Troubleshooting Dynamic SQL

The sp_executesql Stored Procedure

Dynamic SQL and Scope

Client-Side Parameterization

Performance Tuning

SQL Server Storage

Files and Filegroups

Space Allocation

Data Compression

Indexes

Heaps

Clustered Indexes

Nonclustered Indexes

Filtered Indexes

Optimizing Queries

Reading Query Plans

Methodology

Software Testing

MANUAL TESTING

Duration: 12 hrs.

Software Development Life Cycle:

- What are the different phases of SDLC?
- How does the process of Software Development Start?
- Project Initiation

Requirement Gathering and Analysis

- What is Requirement document and what it contains?
- What is use case document and what it contains?
- What is Basic path and Alternate Path?
- Role of Business Analyst

- Example for explaining each phase
- Role of technical specification team
- What is Technical specification document?

What is System Design?

- Role of Design team
- What is design document?
- Role of architecture team

System development

- Role of development team
- Deliverable of Development phase

System testing

- Role of testers and types of testing
- User acceptance testing
- System deployment

System maintenance

- Events in the maintenance phase like bug fixes

Software Testing Life Cycle

- How are the phases of STLC carried out?
- What is testing?
- Role of testers
- Why do we need to test?
- Activities involved in the testing phase

What is test plan and test case document?

- Steps of test case execution
- What does test case document contain?
- How to write test case document?
- What is required to test any application?

TEST CASES

- What is test case?
- What does test case document contain?
- How to write test case document?
- Different test case techniques

TEST PLAN

- What is Test Plan?
- How to write test plan document?
- What does the test plan document contain?

- Who writes and approves the test plan document?
- How manage the test case documents?
- What is the pass/fail criterion?

TYPES OF TESTING

- Different Phases of testing
- What is unit testing?
- What is Minimum acceptance testing?
- What is integration, system and system integration testing?
- What is User acceptance testing?
- What is Regression Testing?

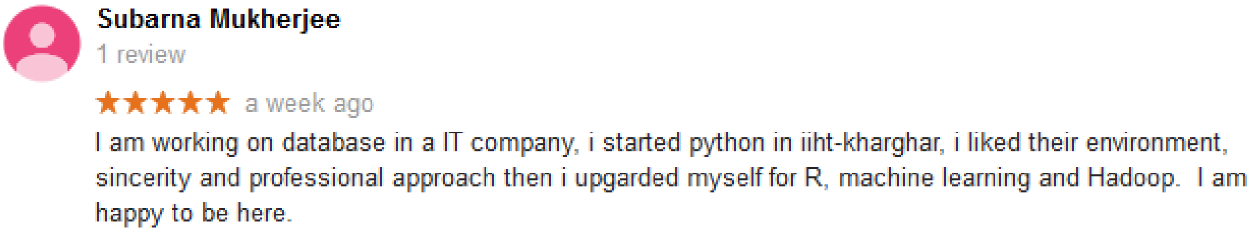
DEFECT ANALYSIS

- What is a defect?
- Various Defect tracking tools
- How to use the defect tracking tools?
- How to enter the details of defect in the defect tracking tool?
- How to identify a defect?
- What is severity and priority?

TRACEABILITY MATRIX

- What is Traceability Matrix[TM]?
- Who Prepares the TM document?
- What is the reference for writing TM?
- What is the use of TM?
- What is present in the TM document?
- Sample TM
- Tools used for developing TM

WHAT STUDENT FEEL ABOUT US:

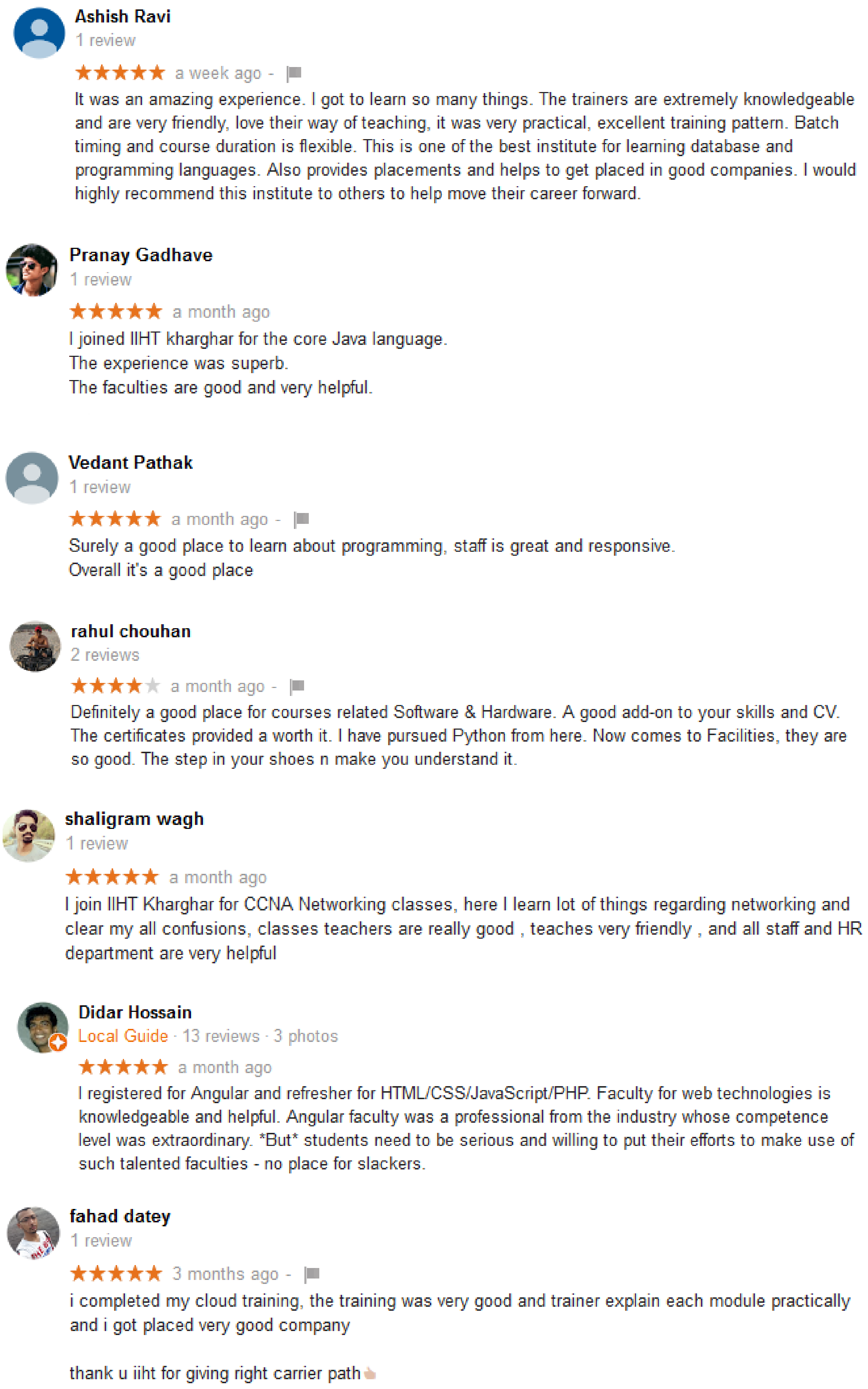


Subarna Mukherjee

1 review

★★★★★ a week ago

I am working on database in a IT company, i started python in iiht-kharghar, i liked their environment, sincerity and professional approach then i upgarded myself for R, machine learning and Hadoop. I am happy to be here.



Ashish Ravi
1 review

★★★★★ a week ago -

It was an amazing experience. I got to learn so many things. The trainers are extremely knowledgeable and are very friendly, love their way of teaching, it was very practical, excellent training pattern. Batch timing and course duration is flexible. This is one of the best institute for learning database and programming languages. Also provides placements and helps to get placed in good companies. I would highly recommend this institute to others to help move their career forward.

Pranay Gadhave
1 review

★★★★★ a month ago

I joined IIHT kharghar for the core Java language.
The experience was superb.
The faculties are good and very helpful.

Vedant Pathak
1 review

★★★★★ a month ago -

Surely a good place to learn about programming, staff is great and responsive.
Overall it's a good place

rahul chouhan
2 reviews

★★★★☆ a month ago -

Definitely a good place for courses related Software & Hardware. A good add-on to your skills and CV. The certificates provided a worth it. I have pursued Python from here. Now comes to Facilities, they are so good. The step in your shoes n make you understand it.

shaligram wagh
1 review

★★★★★ a month ago

I join IIHT Kharghar for CCNA Networking classes, here I learn lot of things regarding networking and clear my all confusions, classes teachers are really good , teaches very friendly , and all staff and HR department are very helpful

Didar Hossain
Local Guide · 13 reviews · 3 photos

★★★★★ a month ago

I registered for Angular and refresher for HTML/CSS/JavaScript/PHP. Faculty for web technologies is knowledgeable and helpful. Angular faculty was a professional from the industry whose competence level was extraordinary. *But* students need to be serious and willing to put their efforts to make use of such talented faculties - no place for slackers.

fahad datey
1 review

★★★★★ 3 months ago -

i completed my cloud training, the training was very good and trainer explain each module practically and i got placed very good company

thank u iiht for giving right carrier path👍

YOU CAN FIND OUR STUDENTS IN:

Microsoft
WIPRO
TATA
TATA CONSULTANCY SERVICES
IBM
accenture
Infosys
Google
NSE IT
JET AIRWAYS
NOKIA
Connecting People
HSBC
Deloitte.
Cognizant Technology Solutions
POLARIS
live your dream
NCR
Honeywell
DELL
MOTOROLA
KEANE
we get IT done.
INTELLIGROUP
Hexaware
TECHNOLOGIES
Reliance
Industries Limited
allied digital
IT managed Responsibly
PCS
Infotech
Creating Business Impact
Bank of America
amazon.com
Capgemini
CLOVER
INFOTECH
CONVERGYS
QUALCOMM
progressive software