About ProGrad:

About ProGrad:

We are solving the great tech talent crunch, one client at a time! We disrupt conventional hiring and training processes, and ensure our client partners onboard the top 1% of best-fit candidates from the length and breadth of India. We break conventional barriers and enable thousands of students of all backgrounds to launch their dream careers. We are betting on technology and our team of passionate coders, creators, trainers, marketers, thinkers and doers to leapfrog our growth. Our founding team includes alumni of IIML, IIMA and BITS Pilani with a wealth of industry experience.

https://prograd.org

We are looking for a Technical Trainer to provide training on Java FSD. Following is the curriculum for the same:

JAVA Curriculum

Foundations (10 Weeks)
(Foundational Curriculum for Java FSD, SDET and Data Streams)

Module	# Weeks	Topics / Areas Covered
Module 1: Fundamentals & Core Language: Java		
Foundation: Java and Programming Fundamentals	3	● ●Basic Java Fundamentals (Syntax, Data types, Variables, Operators, Control Structures, Arrays, Methods etc.,) ●Introduction to IDEs, Plug-ins, Setup, Tools, Enablers to standardize/speed up development ●Documentation and Communication - UML, Mental Maps, PowerPoint etc., ●Basics of working with Source Control systems like Git ●**Exercises**

Core Language: OO Programming in Java, Code Quality, QE Concepts, DB concepts		● ●**OO Programming** (Classes, Objects, Interfaces, Packages, Access modifiers, Encapsulation, Abstraction, Inheritance, Polymorphism etc.,), Generics, Collections, Strings, Multithreading etc., ●Entity classes, Serialization/De-Serialization ●Error/ Exceptions Handling, Custom Exceptions ●Reading and writing data from files, JSON, and XML ●**Unit testing** with Junit, Mockito & Basic Debugging skills. Introduction to BDD approach ●Importance of **Code Quality**, Coding Standards, Standard Solution/Project Structures. Ensuring Code Quality, Code Coverage using tools like SonarQube ●Introduction to **Quality Engineering**, various phases of Testing along with gist of tools (Selenium, TestNG etc.,) in the Market. Introduction to Headless Testing. Aspects around Shift Left approach ●Introduction to **Database, SQL concepts** (Working with MySQL/PostgreSQL) – Tables, Columns, Querying – simple joins, inner/outer joins, “where” clauses, aggregation functions, subquery, stored procedures, index; query tuning basics ●**Exercises**
Advanced Language: Programming Paradigms and Coding in Modern Java	1	● ●Java Versions, LTS, Release Cycles ●**Java (8 – n) Features** Understanding (Interface Enhancements, Functional / Reactive / Modular / Fluent / Stream Based Programming. Functional Interfaces, Lambda Expression / Functions. ForEach – Iterable, Method References, Streams API etc.,) ●Introduction to **SOLID Principles, Design Patterns** ●**Exercises/Hackathon**

Module	# Weeks	Topics / Areas Covered
Module 1: Fundamentals & Core Language: Java		
Foundation: Java and Programming Fundamentals	3	● ●Basic Java Fundamentals (Syntax, Data types, Variables, Operators, Control Structures, Arrays, Methods etc.,) ●Introduction to IDEs, Plug-ins, Setup, Tools, Enablers to standardize/speed up development ●Documentation and Communication - UML, Mental Maps, PowerPoint etc., ●Basics of working with Source Control systems like Git

		●**Exercises**
Core Language: OO Programming in Java, Code Quality, QE Concepts, DB concepts		● ●**OO Programming** (Classes, Objects, Interfaces, Packages, Access modifiers, Encapsulation, Abstraction, Inheritance, Polymorphism etc.,), Generics, Collections, Strings, Multithreading etc., ●Entity classes, Serialization/De-Serialization ●Error/ Exceptions Handling, Custom Exceptions ●Reading and writing data from files, JSON, and XML ●**Unit testing** with Junit, Mockito & Basic Debugging skills. Introduction to BDD approach ●Importance of **Code Quality**, Coding Standards, Standard Solution/Project Structures. Ensuring Code Quality, Code Coverage using tools like SonarQube ●Introduction to **Quality Engineering**, various phases of Testing along with gist of tools (Selenium, TestNG etc.,) in the Market. Introduction to Headless Testing. Aspects around Shift Left approach ●Introduction to **Database, SQL concepts** (Working with MySQL/PostgreSQL) – Tables, Columns, Querying – simple joins, inner/outer joins, “where” clauses, aggregation functions, subquery, stored procedures, index; query tuning basics ●**Exercises**
Advanced Language: Programming Paradigms and Coding in Modern Java	1	● ●Java Versions, LTS, Release Cycles ●**Java (8 – n) Features** Understanding (Interface Enhancements, Functional / Reactive / Modular / Fluent / Stream Based Programming. Functional Interfaces, Lambda Expression / Functions. ForEach – Iterable, Method References, Streams API etc.,) ●Introduction to **SOLID Principles, Design Patterns** ●**Exercises/Hackathon**

Module	# Weeks	Topics / Areas Covered
Module 2: Front End: UI/UX (Angular/React)		

Foundation: UI/Web (Angular/React can be chosen based on the UI stream)	2	· ·Introduction to HTML 5/CSS 3 ·Bootstrap / Similar framework ·Working with Modern Javascript ·Introduction to Angular/React ·Thinking in Angular/React; Lifecycle ·Developer, Debugging Tools ·**Exercises**

Module 3: Enterprise Application Development		
Enterprise Application Development	2	● ● ●Introduction to Spring Framework (Dependency Injection, IOC, MVC etc.,) ●Spring Boot Basics, Creating a Spring Boot application ●Introduction to REST and Microservices. Building REST and Microservices using Spring Boot ●Connecting to Databases, Frameworks for Persistence (Hibernate / ORM, SQL (MySQL/PostgreSQL) & No-SQL (MongoDB), Transactions) ●Introduction to **Application Security** with OAuth / Auth 2.0, Spring Security / JWT ●Swagger Specification and Documentation ●Introduction to Docker, Microservices Containerization using Docker ●API Testing with Postman ●**Exercises**

Module 4: Cloud Basics and App Deployment Pipeline		
Option 1: Build AWS Pipeline		(Preference for Java batch)
Cloud Basics & App Deployment: Build AWS Pipeline	1	● ● ●Introduction to Cloud computing – Cloud concepts, SaaS / PaaS / IaaS, Private / Public / Hybrid Clouds, Cloud Native Development ●Compute, Storage, Network concepts & Services (S3, EC2, VPC, EBS, Files etc.,) ●DB Concepts (RDS, DynamoDB overview etc.,) ●Security Concepts & Services (IAM, Security Group, Policies etc.,)

		●Build and deploy the full stack application to AWS, Elastic Beanstalk ●**Exercises/Hackathon**
Option 2: Cloud Basics & App Deployment: Build GCP Pipeline	1	● ●Similar as above with GCP Services
Option 3: Cloud Basics & App Deployment: Build Azure Pipeline	1	● ●Similar as above with Azure Services (Preference for .NET Batch) ●
Module 5: Capstone Project		
Capstone Project: Skill Evaluation (*Project will start from earlier stages)	1	● ●Complete predefined project in small teams ●Web application that includes UI, RESTful API, Database (Postgres) ●Build, Test and Deploy on Cloud (Based on which cloud they learn) ●Project Documentation ●**Project Presentation/Review**
Common Aspects		● ●Presentation on what they learned each week (as teams) ●Every member of the team presents a small chunk ●Demonstrate individually the successful completion of exercises

Advanced Topics (6 Weeks)

(Advanced Track separately for Java FSD, SDET and Data)

Module	# Weeks	Topics / Areas Covered
Java, Spring Boot, Cloud Native Approach: Advanced Concepts	2	●Introduction to **Cloud Native** concepts and principles ●Introduction to Spring Cloud and Microservices Operations (Registry, Inter-Service communication, Load balancing, Circuit breaker, Monitoring etc.,), Spring Batch ●Introduction to API Gateway. Patterns like Messaging, Event Driven Architecture, CQRS ●Application Security with OAuth / Auth 2.0, Spring Security / JWT ●Spring Boot App for Production (Spring Actuator, KPI to measure in Production) ●Containerization Best practices, Introduction to Kubernetes

		●Implement CI/CD with Jenkins. Toll gate strategy, overview of Telemetry, ELK, JMeter ●**Cloud Services** - Organization, Cost, SNS, SQS, Lambda, Cloud Formation, CLI etc., ●**Exercises**
Option 1: UI/UX: Build and Test Single Page Applications with **Angular**	2	●Working with Typescript ●Building client-side applications with Angular ●Using routing, Directives, Build single page applications. ●Working with server-side APIs, invoke APIs from SPA and packaging apps. ●TDD: Basic use of testing tools (Mocha, Chai, Jasmine). Use test automation tools and Introduction to component programming ●**Exercises/ Hackathon (JavaScript & Angular)**
Option 2: UI/UX: Build and Test Single Page Applications with **React**	2	●Building ReactJS App using JSX and Introduction to Typescript and Webpack. ●React Router & React Context. ●ReactJS, Performance & Optimization ●Test Driven Development – Testing with JEST & Enzyme; ●**Exercises/ Hackathon (JavaScript & React)**
Review/Re-Architect Capstone Project (Based on learnings from Advanced concepts)	2	·Design based on Cloud native best practices ·Introduce API Gateway. Apply Spring Cloud features, Operations ·Develop one of the service as a Serverless function ·Develop with compliance to standards, integrated code quality checks ·Define/Implement NFR (E.g.: Performance: <2 seconds, ADA Compliance – Level A) ·**Project Presentation/Review**