Java Syllabus

For Summer Training Program


```
@Override
public Function simplify(boolean comput
    List<Expression> sim
    for ((int i, Express
        Expression simpl
        if (simplifiedOp
            if (simplifi
                simplifi
            if (i >
                simp
            }
        }
        simplifiedOp
    } else {
        if (simplifi
            simplifi
        }
    }
```

Java

AI in Artificial Intelligence

Java

AI   ML

Java is a high-level, object-oriented, class-based programming language that is widely used for building applications across a variety of platforms.

Key Features of Java:

1. Platform Independent: Java programs run on the Java Virtual Machine (JVM), making them platform-independent (Write Once, Run Anywhere - WORA).
2. Object-Oriented: Everything in Java is treated as an object, promoting modular and reusable code.
3. Simple and Familiar: Its syntax is similar to C++, but it removes many complex features like pointers and multiple inheritance.
4. Secure: It provides a secure environment through features like bytecode verification, exception handling, and security APIs.
5. Robust: Java emphasizes error handling and runtime checking, making it reliable.
6. Multithreaded: Supports concurrent execution of two or more threads for better resource utilization.
7. Rich

Where Java is Used:

- Android App Development
- Web Applications (via Spring, JSP, etc.)
- Enterprise Applications
- Desktop GUI Applications
- Scientific and Research Applications

Core Components of Java Platform:
- **Java Language –** The syntax and rules (keywords, variables, classes, etc.)
- **Java Development Kit (JDK) –** Contains tools to develop Java programs (compiler, debugger, etc.)
- **Java Runtime Environment (JRE) –** Provides libraries and JVM to run Java apps
- **Java Virtual Machine (JVM) –** Executes Java bytecode and makes Java platform-independent

Java's syntax is largely influenced by C and C++, but it removes complex and error-prone features like pointers, operator overloading, and multiple inheritance, thus simplifying the learning curve while maintaining powerful capabilities.

Security is another major strength of Java. The runtime environment enforces strict access controls, sandboxes untrusted code, and checks bytecode for illegal code before execution. This makes Java suitable for networked and distributed environments where code often comes from unknown or untrusted sources.

MODULE 1: JAVA PROGRAMMING BASICS

Objective: Build strong foundations in Java — OOPs, syntax, data types, methods, etc.

Java Setup & IDE
- Install JDK & IntelliJ / Eclipse
- Write first program: System.out.println("Hello, World!")

Java Syntax & Data Types
- Variables, Data Types, Typecasting
- Operators and Expressions

Control Statements
- if-else, switch-case
- for, while, do-while loops

Functions & Methods
- Define & call methods
- Method parameters, return types, recursion

Object-Oriented Programming (OOP)
- Classes, Objects, Constructors
- Inheritance, Polymorphism, Abstraction, Encapsulation
- this keyword, static keyword, access modifiers

Collections & Arrays
- Arrays, ArrayList, HashMap, Set, Looping over collections

Exception Handling
- try-catch-finally
- Custom exceptions, throws and throw

Practice
- 15–20 basic coding problems (loops, arrays, functions, OOP)

MODULE 2: JAVA WITH SPRING BOOT

Objective: Learn how to build REST APIs using Spring Boot

Spring Boot Basics
- What is Spring Boot?
- Create project using Spring Initializr

Project Structure
- Understand @RestController, @RequestMapping, and @PostMapping
- Create endpoints like /hello, /add, etc.

Build REST APIs
- Accept user input (JSON)
- Return a response (JSON)
- Use DTOs (Data Transfer Objects)

WebClient or RestTemplate
- Learn to make external HTTP requests using WebClient
- Use Postman to test APIs

Error Handling
- Global Exception Handling using @ControllerAdvice

Logging
- Use Logger for system logs
- Log API request and response data

MODULE 3: JAVA WITH SPRING BOOT

Objective: Integrate Google Gemini AI to respond to user prompts

Understand Gemini API
- Read Gemini documentation from ai.google.dev
- Create API key from Google AI Studio
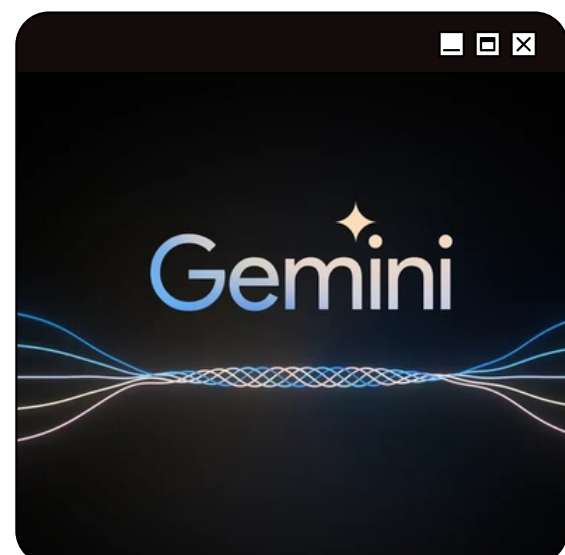
Build Service to Use Gemini
- Use WebClient to send POST request to Gemini
- Handle input prompt and parse response text

Build /chat Endpoint
- Accept JSON: { "prompt": "Tell me a joke" }
- Return JSON: { "response": "Why did..." }

Test with Postman
- Try different inputs: creative writing, FAQs, small talk

MODULE 4: AI IMAGE GENERATOR

Objective: Generate images using imagepig based on user prompt

Choose Image API
- imagepig (Easy to use)

Generate API Key
- Create account → Get key

Create Service for API Call
- Send prompt → Receive image URL
- Parse response and handle errors

Build /image Endpoint
- Input: { "prompt": "A cat flying in space" }
- Output: { "imageUrl": "https://..." }

Test with Postman

MODULE 5: FRONTEND

Objective: Simple HTML + JS to interact with your chatbot and image generator

1. Create a single HTML file with two sections:
   - Chat with Gemini (form + response display)
   - Generate Image (form + image preview)
2. Use JavaScript fetch() to call /chat and /image APIs
3. Show loading spinners, handle errors, add CSS for styling
4. Make the page responsive for mobile view

MODULE 6: FINAL POLISH

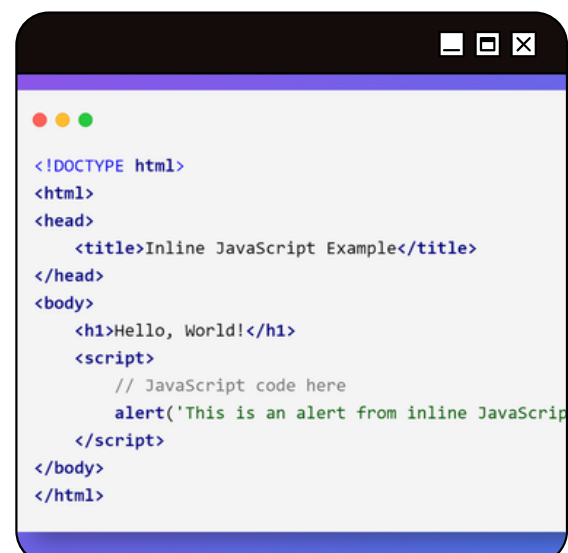**Objective:** Add final features and deploy

Logging
- Log each prompt and AI response to console

Exception Handling
- Handle invalid inputs, API limits, and internal server errors

Code Refactoring
- Organize code into clean packages: controller, service, model, etc.

```html
<!DOCTYPE html>
<html>
<head>
    <title>Inline JavaScript Example</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <script>
        // JavaScript code here
        alert('This is an alert from inline JavaScrip
    </script>
</body>
</html>
```

README + GitHub
- Push code to GitHub
- Add instructions to run project locally
- Optional: Add demo video

MODULE 7: DEPLOYMENT

Objective: Make your project publicly accessible

Backend Deployment
- Use platforms like Render, Railway, or Fly.io

Frontend Deployment (If built)
- Deploy HTML page using Netlify or Vercel

Final Testing
- Test all scenarios end-to-end: UI → Backend → Gemini/Image API

PROJECT DEVELOPMENT

1. **AI Chat & Gen (JAVA)**

DURATION : 50 DAYS

OUTCOME:

1. **Prompt-to-Image Workflow Functional:** Understand and implement a complete workflow where a user's text idea is enhanced using Gemini to generate a creative prompt, which is then used to generate an AI image using services like DALL·E or Replicate.

2. **Build and Integrate Image Generation API:** Gain hands-on experience in building RESTful endpoints in Spring Boot that interface with external image generation APIs, handle HTTP requests/responses, and manage JSON parsing and error handling.

3. **Understand Prompt Engineering for Visual Content:** Learn to design and optimize text prompts using Gemini to produce visually rich and accurate descriptions suitable for realistic or artistic image generation.

PROJECT VIEW

JAVA SUMMER TRAINING SYLLABUS
PROGRESS TRACKER

PROJECTS

◯ AI Chat & Gen (JAVA)

NOTES

HINDTECH
IT SOLUTIONS

Training	Duration	Amount
Java Summer Training Program	**45 to 50 Days**	**₹6,000**

🌐 www.hindtechitsolutions.com

🌐 www.hindtechlearningpoint.com

✉ hr@hindtechitsolutions.com

✉ hindtechlearningpointlko@gmail.com

📞 +91 7905320279

📞 +91 6307738600

📍 Building No 10/703, Ground Floor, near Arvindo Park Road, Sector 10, Indira Nagar, Lucknow, Uttar Pradesh 226016

Branch 1

📍 PARMAR PLAZA, 1st Floor, Bakshi Ka Talab, Bargadi Magath, Uttar Pradesh 226201

Branch 2

Scan Me:

Scan this QR code for details

HINDTECH
LEARNING POINT

Skill India
कौशल भारत - कुशल भारत

N·S·D·C
National Skill Development Corporation
Transforming the skill landscape