AIM

CODE SCHOOL

AN ACCREDITED AIM INSTITUTE PROGRAM

Full Stack with Java Syllabus

Class Description: Love that ultimate control? In this 144-hour course you'll learn the programming language that's relied upon by corporations of all shapes and sizes, Java.

Class Goals: By the end of the course you'll design and implement a fully functional web application using Eclipse as your Integrated Development Environment (IDE) and implement web pages using HTML5 and CSS3. You'll code interactive, dynamic web pages using JavaScript and jQuery. Finally, you'll learn how to connect the client to the server by using AJAX to retrieve data.

Class Objectives:

- Participants will come out of this module with foundational knowledge equivalent to that needed of a full stack developer.
- Participants will build a foundation of knowledge on which to further grow programming language knowledge

Needed Materials:

- Laptop with the following minimum specs:
    - 3 years old or newer and have 2 GB of RAM (8GB or more recommended) as well as 2.5 GB available hard-disk space for installation; additional free space required during installation (cannot install on a volume that uses a case-sensitive file system or on removable flash storage devices).
- Access to portal.aimcodeschool.org
- Pens
- Paper

Grading:

- Grading will be on a pass-fail basis and determined on participation, attendance and completion of assigned course work.
- All assignment deadlines are final, no extensions will be granted.
- If a participant misses one day, it is up to him/her to reach out to the instructors for makeup assignments.
- Pass: A student must successfully complete at least 70% of the class objectives, tasks, and knowledge requirements
- Fail: a student who does not successfully complete at least 70% of the class objectives, tasks, and knowledge requirements

Assignments:

- This class is designed to give participants ample opportunity to complete all assignments in class. However, should a participant be unable to complete the assigned course work in class, it will be up to them to complete at home.

Ideal Student: This course is open to individuals with 6 months of front-end web development experience and/or graduates of a Code School Foundations course who want to build the skills to begin a career as a web application developer.

Topics Covered in This Course:

1. Introduction to backend programming
2. Initial setup
    a. Git
    b. Github
    c. Push/pull repositories
    d. Discord
    e. Visual Studio Code
3. JDK/JRE
    a. Download / install
    b. Importance - why do we need these?
4. OOP

    a. Structure
    b. Difference between OOP and “procedural” programming

5. Taste of Java
    a. How to create a java project in VS Code
    b. Hello World Program
6. Primitives
    a. Data Types
7. Variables
    a. Naming rules
    b. Instance
    c. Class
    d. Local
    e. Parameters/Arguments
    f. Assignment operators
    g. Constance
    h. Variables vs Constants
    i. Default values
    j. Literals
8. Arrays
9. Operators
    a. Precedence (BODMAs)
    b. Bitwise and Bitshift Operators
10. Expressions
11. Statements
    a. Control Flow Statements
12. Conditional statements
    a. If-then
    b. If-then-else
    c. Switch

13. Loops
    a. While
    b. Do-while
14. Input and Output
    a. Scanner Class
    b. Scanner Object
    c. Scanner Methods
    d. in.nextInt() method
    e. im.nextDouble() method
    f. System.out. println(), system.out. print(), & system.out. printf() differences
    g. Format
    h. Format specifiers
15. Classes (blueprints)
    a. Class Name
    b. Data Members
    c. Methods
        i. Defining methods
            1. Modifiers
            2. Return type
            3. Method name
            4. Parameter list
            5. Method body
16. Constructors
    a. What is it
    b. How does it work
    c. Types of constructors
    d. Parameterized constructor
17. Constructors for classes
18. Methods

    a. Passing information to a method or constructor
    b. Passing by Value
    c. . Returning a Value from a method

19. Packages
    a. What makes up Java packages
    b. Java API
    c. Importing packages
    d. Syntax
20. Abstraction
21. Encapsulation
22. Inheritance
23. Polymorphism
24. Interfaces
25. Collections & Generics
26. JUnit
27. Debugging
28. NoSQL
29. MySQL
30. SQL
31. HTML/ CSS Review
32. JavaScript Review
33. Closure & Scope
34. AJAX
35. Algorithms in JavaScript
36. WireFrames
37. Data Access Objects
38. JSP/ Servlets
39. POJOs
40. Deployment

41. Spring Framework
42. RESTful Web Services
43. Server Management
44. Heroku
45. CI/CD Pipelines
46. Final Project
47. 1. Build a fully functioning web-based application