Here’s a 7-day, 12-hours-a-day study plan designed for quick revision of the software engineer syllabus you shared:

Day 1: Programming Fundamentals (12 hours)

Morning (4 hours)

- **Data Structures**
  - Arrays, Linked Lists, Stacks, Queues
  - Practice coding problems on each topic (e.g., LeetCode, HackerRank)

Afternoon (4 hours)

- **Trees, Graphs**
  - Binary Trees, Binary Search Trees, Graph Traversal (BFS, DFS)
  - Solve problems related to these data structures

Evening (4 hours)

- **Algorithms**
  - Sorting (Merge, Quick, Heap), Searching Algorithms (Binary, Linear)
  - Practice dynamic programming problems (Knapsack, LCS, etc.)

Day 2: Object-Oriented Programming & Software Development Methodologies (12 hours)

Morning (4 hours)

- **OOP Concepts**

    - Inheritance, Polymorphism, Encapsulation
    - Practice implementing these in your preferred language (e.g., Python, Java)

Afternoon (4 hours)

- **Agile Development & Scrum**
    - Revise Agile principles, Scrum roles, and ceremonies
    - Review Test-Driven Development (TDD) and Continuous Integration (CI) concepts

Evening (4 hours)

- **Waterfall Model & Continuous Deployment (CD)**
    - Compare Agile with Waterfall
    - Study the CI/CD pipeline and deployment automation

Day 3: Data Storage & Management (12 hours)

Morning (4 hours)

- **Relational Databases**
    - SQL, MySQL, Data Modeling
    - Practice SQL queries, focus on joins, subqueries, and optimization

Afternoon (4 hours)

- **NoSQL Databases**
    - MongoDB, Cassandra, Data Schema Design
    - Study use cases and differences between SQL and NoSQL databases

Evening (4 hours)

- **Database Performance Optimization**

    - Indexing, Sharding, Data Partitioning
    - Focus on real-world database performance tuning techniques

Day 4: Software Design Patterns (12 hours)

Morning (4 hours)

- **Creational Patterns**
    - Singleton, Factory, Builder
    - Review examples in code and identify their use cases

Afternoon (4 hours)

- **Structural Patterns**
    - Adapter, Bridge, Composite
    - Focus on solving design pattern-related problems

Evening (4 hours)

- **Behavioral Patterns**
    - Observer, Strategy, Template Method
    - Study practical implementations and benefits in code architecture

Day 5: Web Development (12 hours)

Morning (4 hours)

- **Front-End Development**
    - HTML, CSS, JavaScript, React
    - Build a basic front-end project using React and CSS

Afternoon (4 hours)

- **Back-End Development**
  - Node.js, Django
  - Set up a basic RESTful API using Node.js or Django

Evening (4 hours)

- **RESTful API Design & Web Services**
  - Study RESTful API best practices, focus on statelessness and scalability
  - Understand SOAP vs REST

Day 6: Testing, Debugging, and Cloud Computing (12 hours)

Morning (4 hours)

- **Unit & Integration Testing**
  - JUnit, PyUnit
  - Write and run unit tests for your back-end API from Day 5

Afternoon (4 hours)

- **Debugging Techniques**
  - Practice using print statements, breakpoints, and debuggers in your IDE
  - Focus on debugging complex algorithms and code

Evening (4 hours)

- **Cloud Computing**
  - AWS, Google Cloud basics
  - Study cloud deployment models (IaaS, PaaS, SaaS) and cloud security

Day 7: DevOps & Soft Skills (12 hours)

Morning (4 hours)

- **Containerization**
  - Docker basics, container orchestration using Kubernetes
  - Set up a basic Docker container for your API

Afternoon (4 hours)

- **Infrastructure as Code & Monitoring**
  - Study Terraform basics and review monitoring tools like Prometheus, Grafana
  - Go through Infrastructure as Code (IaC) principles

Evening (4 hours)

- **Soft Skills & Interview Prep**
  - Review common behavioral interview questions
  - Practice problem-solving scenarios and time management techniques

This plan ensures you cover each topic thoroughly in 7 days with hands-on practice, coding exercises, and revision of theoretical concepts.