**Module 1: Software Engineering Fundamentals**

* **Software Engineering**: Application of scientific principles to develop software systems.
* **Responsibilities**: Design, build, test, and maintain software; collaborate with stakeholders.
* **SDLC Benefits**: Clear structure, better communication, reduced risks, increased efficiency.
* **Phases of SDLC**:
  1. Requirement Gathering
  2. Design
  3. Implementation
  4. Testing
  5. Deployment
  6. Maintenance
* **Requirement Process**: Identify stakeholders → Elicit → Document → Analyze → Prioritize → Confirm.
* **Requirement Documents**:
  1. **URS**: User needs.
  2. **SRS**: Functional + performance requirements.
  3. **SysRS**: System-wide specs (security, interfaces, policies).
* **Development Models**:
  1. **Waterfall**: Sequential
  2. **V-Model**: Emphasizes testing
  3. **Agile**: Iterative, flexible
* **Testing Types**:
  1. **Functional**: Correct output for input.
  2. **Non-functional**: Performance, scalability, security.
  3. **Regression**: Confirms no breakage from updates.
* **Documentation Types**: Requirements, technical, QA, user guides.
* **Project Roles**: Project Manager, Devs, QA, UX Designers, Product Owners, Tech Writers.

**Module 2: Web & Cloud Technologies**

* **Website Basics**: Front-end (HTML/CSS/JS) interacts with back-end via HTTP.
* **Responsive Design**: CSS (media queries), JS frameworks (React, Bootstrap).
* **Back-End Technologies**: Handle business logic, database, authentication (Node.js, Python, PHP).
* **Team Collaboration**: Better code quality, fewer bugs, knowledge sharing.
* **Pair Programming**: Enhances skills, reduces bugs.
* **Tech Classifications**:
  + **Front-end**: HTML, CSS, JavaScript, React
  + **Back-end**: Node.js, Express, SQL, MongoDB
* **Development Tools**:
  + **Version Control**: Git, GitHub
  + **Frameworks**: Django, Angular
  + **CI/CD Tools**: Jenkins, GitHub Actions
  + **Package Managers**: npm, pip
* **Popular Stacks**:
  + **LAMP**: Linux, Apache, MySQL, PHP
  + **MEAN**: MongoDB, Express, Angular, Node.js
  + **MERN**: MongoDB, Express, React, Node.js
* **IDE Components**: Code editor, terminal, debugger, output console.

**Module 3: Programming Fundamentals**

* **Language Types**:
  + **Interpreted**: Python, JavaScript
  + **Compiled**: C++, Java
* **High-Level Languages**: Easy to understand (Python, Java)
* **Low-Level Languages**: Close to machine code (Assembly)
* **Code Planning**: Use **flowcharts** (visual logic) and **pseudocode** (plain language).
* **Identifiers**: Names for constants and variables.
* **Functions**: Reusable code blocks for specific tasks.
* **OOP**: Focuses on **objects** (data + behavior) based on **classes** (blueprints).

**Module 4: Software Architecture & Environments**

* **Software Architecture**: Acts as a **blueprint** for development, ensuring consistency and quality.
* **Structured Design**: Breaks down complex problems into manageable elements.
* **Behavioral Models**: Describe what the system does without detailing how.
* **UML Diagrams**: Save time/money. Types: Class, State Transition, Interaction.
* **Objects vs Classes**:
  + **Object**: Instance with data and behavior.
  + **Class**: Blueprint to create objects.
* **SOA**: Loosely coupled services communicating via network protocols.
* **Distributed Systems**: Appear as one system but run across multiple machines.
* **Architectural Patterns**:
  + **2-tier, 3-tier**
  + **Event-driven**
  + **Microservices**
  + **Peer-to-peer**
* **Application Environments**:
  + **Development**, **Testing/QA**, **Staging**, **Production**
  + Production includes non-functional needs like load and security.
* **Deployment Options**:
  + On-premises, Public/Private/Hybrid Cloud
* **Production Setup Includes**: Firewalls, Load balancers, Web/app servers, Proxy servers, DB servers.

**Module 5: Career & Professional Practice**

* **Role of a Software Engineer**: Build and maintain software; continuous learning is key.
* **Skills Needed**:
  + **Hard Skills**: Coding, debugging, architecture
  + **Soft Skills**: Communication, teamwork, problem-solving
* **Job Outlook**: High demand, flexible, rewarding roles.
* **Career Paths**:
  + **Technical**: Architect, Lead Developer
  + **Managerial**: Project/Product Manager
  + **Other Fields**: QA, DevOps, Tech Writing
* **Job Titles May Include**: Software Developer, Front-End/Back-End Engineer, QA Engineer, DevOps Engineer.
* **Code of Ethics** (8 Principles):
  + Public
  + Client/Employer
  + Product
  + Judgment
  + Management
  + Profession
  + Colleagues
  + Self