Education & Learning / Curriculum Design

Map content across time with prerequisite dependencies — curriculum architecture for coherent learning progression.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Course Planning, Curriculum Mapping
Updated: June 2026
Why This Prompt Exists
Disorganized curriculum jumps between topics without logic. Students get confused when concepts appear before prerequisites. Most courses are sequenced by textbook chapters, not learning dependencies.

You get:

- topics taught before prerequisites (students lost)
- no logical progression (random topic order)
- uneven pacing (too fast, then too slow)
- missed connections between related concepts
- spiral curriculum that doesn’t actually spiral

But effective scope and sequence has structure:

- scope: what topics are covered (breadth)
- sequence: order of topics (progression)
- prerequisites: what must come before what
- pacing: time allocation per topic
- spiraling: revisiting topics with increasing depth

Without scope and sequence, curriculum is chaotic.

This prompt maps content across time with prerequisite dependencies.

The Prompt

```
Assume the role of a curriculum architect who designs scope and sequence.

Your task is to map content across time with prerequisite dependencies.

Generate:

1. COURSE PARAMETERS
   - Course title: [name]
   - Duration: [weeks / months / semesters]
   - Meeting frequency: [X hours per week]
   - Total instructional hours: [X]

2. TOPIC INVENTORY (Scope)
   - Major units: [list of 3-7 units]
   - Subtopics per unit: [list]
   - Total topics: [X]

3. PREREQUISITE DEPENDENCY MAP

| Topic | Requires (prerequisite) | Required for (subsequent) |
|-------|-------------------------|---------------------------|
| [topic 1] | None | [topic 2, topic 3] |
| [topic 2] | [topic 1] | [topic 4] |
```

4. SEQUENCE BY WEEK/UNIT

Unit 1: [Name] (Weeks X-Y)
- Prerequisites: none
- Topics:
  - Week 1: [topic 1.1]
  - Week 2: [topic 1.2]
  - Week 3: [topic 1.3]

Unit 2: [Name] (Weeks X-Y)
- Prerequisites: Unit 1
- Topics:
  - Week 4: [topic 2.1]
  - Week 5: [topic 2.2]

5. PACING CALCULATION

Topic	Estimated Hours	Activities	Assessment
[topic 1]	X	Lecture, practice, discussion	Quiz
[topic 2]	X	Lab, group work	Project

6. SPIRAL CURRICULUM MAP (recurring concepts)

Concept	Unit 1	Unit 2	Unit 3	Unit 4
[concept A]	Introduce	Practice	Apply	Master
[concept B]	—	Introduce	Practice	Apply

7. SCOPE AND SEQUENCE VERIFICATION

- [ ] No topic appears before its prerequisites
- [ ] Pacing is realistic (not too fast, not too slow)
- [ ] Units are coherent (topics within unit belong together)
- [ ] Spiral concepts reappear with increasing depth
- [ ] Assessment matches timing (not all at end)
- [ ] Total hours match available time

8. COMMON SCOPE AND SEQUENCE MISTAKES

Mistake	Why It Fails	Correct Approach
No prerequisites map	Topics out of order	Map dependencies first
Uneven pacing	Rushed or bored students	Allocate time by complexity
No spiraling	Concepts forgotten	Revisit with increasing depth
Too many topics	Surface coverage only	Fewer topics, deeper learning
Assessment only at end	No feedback loop	Formative assessments throughout

INPUTS:

Course title:
[PASTE TITLE]

Course duration:

```
[E.G., "15 weeks", "1 semester", "6 months"]

Topics to cover (scope):
[PASTE LIST OF TOPICS]

Prerequisite knowledge (what students already know):
[PASTE LIST OR "NONE"]

Available hours:
[E.G., "45 instructional hours"]

RULES:
- Map prerequisites before sequencing (what must come before what)
- No topic appears before its prerequisites
- Allocate time by topic complexity (hard topics need more time)
- Build spiral curriculum (revisit concepts with increasing depth)
- Include formative assessments throughout (not just final exam)
- Verify total hours match available time (reduce scope if needed)
- Leave buffer time for review and catch-up (10-15% of total time)
```

How To Use It

- Map prerequisites before sequencing — know what must come before what.
- No topic appears before its prerequisites — that’s the cardinal rule of sequencing.
- Allocate time by topic complexity — hard topics need more time, easy topics need less.
- Build spiral curriculum — revisit concepts with increasing depth, not just once.
- Include formative assessments throughout — not just a final exam at the end.
- Verify total hours match available time — if they don’t, reduce the scope.
- Leave buffer time for review and catch-up — 10-15% of total time for flexibility.

Example Input

Course title: “Introduction to Web Development”

Course duration: “12 weeks”

Topics to cover: “HTML basics, CSS styling, Flexbox, Grid, JavaScript variables, JavaScript functions, DOM manipulation, APIs, Git basics, Deployment”

Prerequisite knowledge: “Basic computer literacy (file management, web browsing)”

Available hours: “36 instructional hours (3 hours/week)”

Why It Works
Disorganized curriculum jumps between topics without logic. Students get confused when concepts appear before prerequisites.

This framework improves outcomes by forcing:

- topic inventory (scope: what’s covered)
- prerequisite dependency mapping (what must come before what)
- weekly sequencing (when each topic is taught)
- pacing calculation (time allocation per topic)
- spiral curriculum mapping (recurring concepts with increasing depth)

Failure modes this prevents:

- topics taught before prerequisites (students lost)
- no logical progression (random topic order)
- uneven pacing (too fast, then too slow)
- missed connections between related concepts

This improves on: Textbook-chapter sequencing. Prerequisite-driven sequencing ensures

coherent progression.

Related to: CD-01 (Learning Objectives) for outcomes; CD-03 (Unit Plan) for detailed design; CD-04 (Assessment Blueprint) for testing alignment.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Assessment Blueprint