

coursera

Adaptive Assignment Coach

Transforming Learning Friction into Mastery

Coursera: Vision & Global Scale



Vision

To provide universal access to world-class learning, partnering with 300+ leading universities and companies.



Revenue

Diverse streams across Consumer (Subscriptions), Enterprise (Training), and Degrees (Online Programs).



Ecosystem

Serving over 100M+ learners globally who love the flexibility and prestige of the platform.

Coursera: User & Business Value

User Value

- **Credibility:** Recognized certificates from world-class institutions.
- **Accessibility:** High-quality education at a fraction of on-campus costs.
- **Career Growth:** Job-ready skills mapped to employer requirements.

Business Value

- **Partnership Network:** Unmatched distribution for university and corporate content.
- **Ecosystem Lock-In:** Multi-product portfolio from free MOOCs to full degrees.
- **B2B Expansion:** Strategic training platform for Fortune 500 companies.

The Problem: High Attrition in STEM

⚠️ The "Cognitive Wall"

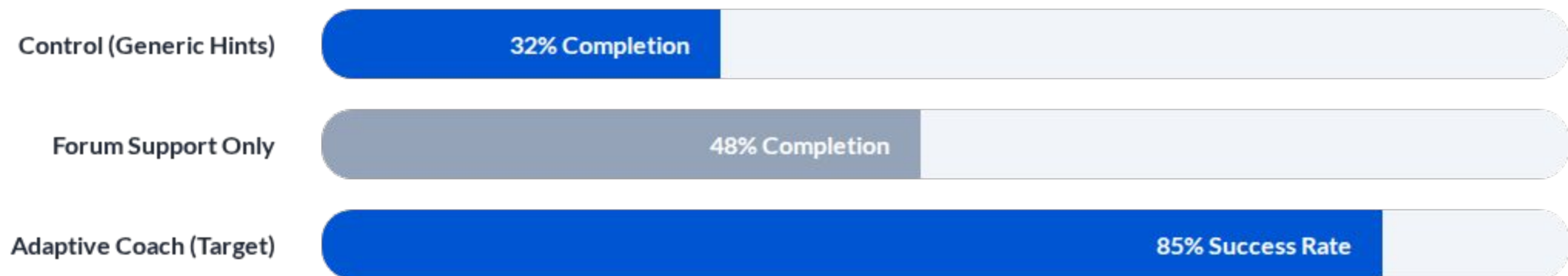
Learners face immense friction during graded coding assignments. When generic hints fail, two outcomes occur:

- **Abandonment:** Learners churn from the course and platform.
- **Dishonesty:** Learners resort to external solution copying (GitHub/Chegg).

We are losing learners at the most critical moment of their mastery journey.



Market Opportunity & Strategic Impact



Target: Drive a 2.5x increase in assignment pass rates for technical specializations.

These percentages are modeled target benchmarks, not historical results.

User Persona: Priya



The "Career Switcher"

Bio: 29, QA Engineer upskilling to Data Automation. Studies around a busy full-time job.

Pain Point: "I understand the videos, but I freeze when I see the code editor. I need a nudge, not the answer."

Goal: True mastery of algorithmic logic for technical interviews.

In-Context AI Tutoring

The "Need Help?" button activates a Socratic side-panel directly within the Coursera Lab interface.

- **Safe:** No final code reveal.
- **Contextual:** Reads user attempts in real-time.
- **Menu-Driven:** Four targeted help paths.

The screenshot shows the Coursera Lab interface for a Python programming assignment. The page title is "Week 3 · Assignment 2 Introduction to Python Programming". The problem is titled "Find Maximum Number". The instructions ask for a Python function `find_max(numbers)` that returns the largest number in a list. Requirements include handling lists of any length, returning `None` for empty lists, and not using Python's built-in `max()` function. Examples show `find_max([3, 1, 4, 1, 5, 9])` returning 9, `find_max([-1, -5, -2])` returning -1, `find_max([42])` returning 42, and `find_max([])` returning `None`. Constraints state that all elements are integers or floats and the list length is at most 10,000. A terminal window shows the command `python solution.py` and the output `9`. A "Need Help?" button is visible in the bottom right corner.

This screenshot shows the same Coursera Lab interface as above, but with the "Adaptive Assignment Coach" side-panel open. The coach is powered by AI in Socratic Mode and offers four help options: "Explain Task" (simplifies the prompt), "Show Similar Example" (shows related problems), "Check My Approach" (analyzes code for logic gaps), and "Practice More" (provides drills on similar concepts). A footer message states: "I'm a Socratic coach. I can't give you the final code, but I can help you find it."

Multimodal Scaffolding

The screenshot shows the Coursera interface for 'Week 3 - Assignment 2 Introduction to Python Programming'. The main content area displays the 'Find Maximum Number' problem description, requirements, examples, and constraints. The 'Adaptive Assignment Coach' sidebar is open, showing a 'Task in Plain Language' section. The task is described in simple terms: 'Imagine you have a bag of numbered balls. Your job is to reach in, compare them one by one, and find the biggest one.' It includes input, output, and key constraint information. A note asks the user to think about handling an empty bag. At the bottom, a message from the Socratic coach says, 'I'm a Socratic coach. I can't give you the final code, but I can help you find it.'

Plain Language

Simplifies complex tasks in simple plain language.

The screenshot shows the same Coursera interface as the previous one, but the 'Adaptive Assignment Coach' sidebar is open to a 'Similar Pattern: Find Minimum' section. It provides a working solution for a related problem, 'Find Minimum', with a code snippet:

```
def find_min(numbers):  
    """Find the smallest number."""  
    if not numbers:  
        return None  
    min_num = numbers[0]  
    for num in numbers:  
        if num < min_num:  
            min_num = num  
    return min_num
```

 A note highlights the empty list check: 'Notice the highlighted lines — this function checks for an empty list before accessing any elements. Does your code do this?'. At the bottom, the same Socratic coach message is present: 'I'm a Socratic coach. I can't give you the final code, but I can help you find it.'

Similar Pattern

Uses "Brother Problems" to spark recall.

Approach Analysis: Precision Logic Auditing

The AI maps the student's attempt against the solution path to identify exactly where reasoning diverged.

Divergence Map: Highlights line-by-line gaps, such as missing "Guard Rails" for empty lists.

Socratic Hint:

"Look closely at Line 3. If someone passes an empty list, your function tries to access the first element of... nothing. How might you guard against that?"

The screenshot shows a Coursera assignment page for "Week 3 · Assignment 2 Introduction to Python Programming". The main content area displays a problem description for "Find Maximum Number" with requirements, examples, and constraints. A terminal window at the bottom shows the execution of a Python script. On the right, an "Adaptive Assignment Coach" panel is open, showing an "Approach Analysis" section with a "DIVERGENCE MAP". The map highlights a "Logic Gap" on line 3, where the code attempts to access the first element of an empty list. A Socratic Hint is provided, asking the student to consider what happens before the loop starts. A message at the bottom of the coach panel states: "I'm a Socratic coach. I can't give you the final code, but I can help you find it."

Problem Description: Find Maximum Number

Write a Python function `find_max(numbers)` that takes a list of numbers as input and returns the largest number in the list.

Requirements:

- The function should handle lists of any length
- Return `None` if the list is empty
- Handle both positive and negative numbers
- Do not use Python's built-in `max()` function

Examples:

```
find_max([3, 1, 4, 1, 5, 9]) # Returns 9
find_max([-1, -5, -2]) # Returns -1
find_max([42]) # Returns 42
find_max([]) # Returns None
```

Constraints:

- All elements are integers or floats
- The list will contain at most 10,000 elements

Terminal:

```
>_ Terminal
$ python solution.py
9
>>> |
```

Adaptive Assignment Coach: Powered by AI - Socratic Mode

Approach Analysis:

DIVERGENCE MAP:

- L1 `def find_max(numbers):` ✓ OK
- L2 `"""Find the largest..."""` ✓ OK
- L3 `max_num = numbers[0]` ⚠ Logic Gap
- L4-7 `for num in numbers: ...` ✓ Logic sound

Socratic Hint:

You're iterating through the list correctly, but **have you considered what happens before the loop starts?**

Look closely at Line 3. If someone passes an empty list, your function tries to access the first element of... nothing. How might you guard against that?

Think about: What should the function return when there are zero items to compare?

I'm a Socratic coach. I can't give you the final code, but I can help you find it.

Reinforcement Learning

The Mastery Cycle

One-click "Practice More" generates 3 variant questions to cement concept durability.

Ensures the learner didn't just guess, but truly understands the underlying pattern.

The screenshot displays a Coursera assignment page for "Week 3 - Assignment 2 Introduction to Python Programming". The main content area is titled "Find Maximum Number" and contains the following text:

Write a Python function `find_max(numbers)` that takes a list of numbers as input and returns the largest number in the list.

Requirements:

- The function should handle lists of any length
- Return `None` if the list is empty
- Handle both positive and negative numbers
- Do not use Python's built-in `max()` function

Examples:

```
find_max([3, 1, 4, 1, 5, 9]) # Returns 9
find_max([-1, -5, -2]) # Returns -1
find_max([42]) # Returns 42
find_max([]) # Returns None
```

Constraints:

- All elements are integers or floats
- The list will contain at most 10,000 elements




At the bottom, a terminal window shows the command `python solution.py` being executed, resulting in the output `9`.

On the right side, there is an "Adaptive Assignment Coach" sidebar. It features a "Quick-Fire Practice" section with three questions:

- Q1 (Easy):** "What does `numbers[0]` return if `numbers = []`?" with options: `0`, `None`, `IndexError`, and `False`.
- Q2 (Medium):** "Which check correctly handles an empty list?" with options: `if numbers == 0:`, `if len(numbers):`, `if not numbers:`, and `if numbers is None:`.
- Q3 (Hard):** (The question text is partially obscured).

A message at the bottom of the sidebar states: "I'm a Socratic coach. I can't give you the final code, but I can help you find it."

Socratic Logic Engine

-  **Logic Auditing:** Identifies the exact moment a student's logic drifts from the instructor's solution path.
-  **Pedagogical Guardrails:** The system is hard-coded to refuse code generation, providing only scaffolding and hints.
-  **Synthetic Variation:** Automatically creates structurally similar challenges to verify true mastery



Phased Product Rollout



Alpha

Pilot with 5 technical Specializations.



Beta

Expansion to Coursera for Business partners.



Scale

Integration into all STEM degree courses.



Global

Platform-wide tutoring support GA.

Questions?

Building the future of mastery-based
learning.

coursera

Strategic Product Initiative | Q2

2026

Image Sources



https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/Coursera_logo_%282020%29.svg/1280px-Coursera_logo_%282020%29.svg.png

Source: ar.wikipedia.org



https://media.istockphoto.com/id/1284549979/photo/young-overworked-man-with-laptop-at-night.jpg?s=612x612&w=0&k=20&c=5J9ppYfM2DL8sYva1w0zRsFt_CWRTdekBdZ3UwZVc_s=

Source: www.istockphoto.com



<https://media.istockphoto.com/id/2187293569/photo/concentrated-woman-working-on-laptop-in-calm-home-office.jpg?s=612x612&w=0&k=20&c=rT3TIDe7-OqvFe4BIGFDQjkUCIm6QgBfxObzFogbyw=>

Source: www.istockphoto.com



https://static.vecteezy.com/system/resources/thumbnails/073/907/210/small_2x/abstract-digital-brain-glowing-blue-neural-connections-dark-background-represents-ai-machine-learning-technology-data-intelligence-innovation-science-ideal-for-tech-health-future-concepts-video.jpg

Source: www.vecteezy.com