

SAMIR GENAI ACADEMY

PROFESSIONAL STUDY GUIDE SERIES

Phase 03 · Agentic AI Engineering

14

CrewAI Multi-Agent Systems

Role-playing agent teams, crews, flows & task orchestration

Module 14 of 25 · 1 hour

Instructor: **Samir Kumar** · Edition 2026

STUDY GUIDE

01 Overview

Role-playing agent teams, crews, flows & task orchestration. This guide gives you the core theory, the tools that matter, a hands-on build, working code, and the pitfalls to avoid — everything you need to learn the topic to a professional standard and apply it in real projects.

1 hour

LESSON LENGTH

4 + 4

CONCEPTS + BUILD
STEPS

3

FRAMEWORKS & TOOLS

14/25

PHASE 03

02 Learning Objectives

- ▶ Design role-based agent teams (crews).
- ▶ Define tasks and assign them to agents.
- ▶ Choose sequential vs hierarchical process orchestration.
- ▶ Build a research crew that produces a deliverable.

03 Core Concepts

Crews & roles

CrewAI frames multi-agent work as a crew of role-playing agents (researcher, analyst, writer), each with a goal and backstory that shape behaviour. Role specialisation produces higher-quality output than one generalist agent.

Tasks

Work is expressed as tasks with descriptions and expected outputs, assigned to agents. Tasks can pass context to one another, forming a pipeline where one agent's output feeds the next.

Process types

Sequential runs tasks in order; hierarchical adds a manager agent that delegates and reviews. Choose hierarchical when work needs coordination and quality control, sequential when steps are well defined.

Tools & flows

Agents use tools (search, code, APIs) to act. CrewAI Flows add event-driven control for more deterministic, branching orchestration when crews alone are too loose.

04 Visual Diagram



Figure 14 — Role-specialised agents pass work down a pipeline to produce a final deliverable.

05 Key Frameworks & Tools



06 Hands-On Build

✂ Stock Picker Financial Research Crew

1. Define researcher, analyst, and writer agents with goals/backstories.
2. Create tasks: gather data -> analyse -> write recommendation.
3. Give the researcher a web-search tool.
4. Run sequentially, then try a hierarchical manager and compare.

07 Code Walkthrough

A CrewAI crew with roles and tasks

PYTHON

```
from crewai import Agent, Task, Crew, Process

researcher = Agent(role="Analyst", goal="Find key financials",
                   backstory="Veteran equity researcher.")
writer = Agent(role="Writer", goal="Write a clear recommendation",
               backstory="Investment newsletter author.")

research = Task(description="Research {ticker} fundamentals", agent=researcher,
                 expected_output="Bulleted financial summary")
report = Task(description="Write a buy/hold/sell note", agent=writer,
               expected_output="One-page recommendation")

crew = Crew(agents=[researcher, writer], tasks=[research, report],
            process=Process.sequential)
print(crew.kickoff(inputs={"ticker": "AAPL"}))
```

08 Lab Assistant

Lab 14 • Build a research crew

Prerequisites: pip install crewai crewai-tools • A search-tool API key

1. Define researcher/analyst/writer with goals + backstories

Expected: Distinct, role-appropriate behaviour

2. Chain tasks with expected_output and run sequentially

Expected: A one-page recommendation

3. Switch to a hierarchical manager process

Expected: A manager delegates and reviews

Troubleshooting

- Generic output → make goals/backstories specific.
- Tasks drift → always set expected_output.

09 Pitfalls & Key Takeaways

⚠ COMMON PITFALLS

- Vague roles/goals that produce generic, overlapping output.
- Too many agents for a task that one or two could handle.
- No expected_output, so tasks drift off target.

✓ KEY TAKEAWAYS

- Role specialisation raises multi-agent output quality.
- Tasks with clear expected outputs keep crews on target.
- Hierarchical processes add coordination and review.

 **Companion video:** Watch the module 14 lesson in your Student Dashboard, then use this guide to revise, copy the code, and complete the build. Download both for offline study.

10 Further Resources

CrewAI docs • CrewAI examples repo • CrewAI Flows guide