



AI CREATOR • AGENTIC AI & RAG ENGINEERING

Engineer AI Systems. Think Beyond the Prompt.

6 Months • Weekly 60-Minute LIVE Class • Small Skill-Based Cohorts

Designed for advanced students ready to architect agentic AI systems, advanced RAG pipelines, tool-using workflows and evaluated AI applications.

100% LIVE

AI SYSTEMS LAB

PROJECT-BASED

AI SYSTEMS MINDSET

Strong Python skills are the foundation for building reliable AI systems — not just using AI tools.

Agents → advanced RAG → guardrails → evaluation → AI systems.



PROGRAM OVERVIEW

Six months. From AI Builder to confident AI Creator.

24 LIVE CLASSES

Approximately 24 weekly sessions
60 minutes each
Hands-on coding every week
Real-time instructor guidance

AI SYSTEMS ENGINEERING

agent design → advanced RAG → tools →
orchestration → guardrails → evaluation →
production-minded AI systems

Students learn to connect models, data,
tools and evaluation into systems.

LEARN BY BUILDING

AI Systems Lab • Agent Debugging •
Advanced RAG • tools • evaluation •
automation • architecture • testing •
capstone

Every project emphasizes design decisions
and maintainability.

OUTCOME

**Students finish able to architect, test, debug, document and present substantial AI systems —
agentic, grounded and evaluated.**



MONTHS 1-2

AGENTIC AI + SYSTEM ARCHITECTURE

01 • AGENT ARCHITECTURE

agents vs workflows • state • tools • planning concepts • memory patterns • orchestration • separation of concerns

AI Systems Lab: design deterministic workflows before adding agent autonomy

02 • ADVANCED RAG + DATA DESIGN

chunking strategies • metadata • hybrid retrieval concepts • reranking • context assembly • retrieval evaluation

Build: RAG systems that retrieve and reason over richer knowledge sets

GOAL Students move from building AI apps to designing AI systems with clear responsibilities, state and structure.

MONTHS 3–4

TOOLS + GUARDRAILS + AGENT DEBUGGING

03 • TOOLS + ORCHESTRATION

function/tool calling • multi-tool routing • API integrations • state transitions • retries • failure handling

Build: AI workflows that coordinate multiple controlled tools

04 • GUARDRAILS + EVALUATION

eval datasets • rubrics • LLM-as-a-judge concepts • deterministic checks • safety boundaries • human review

Build: AI systems designed to fail visibly, safely and measurably

AGENT DEBUGGING LAB

Reproduce → isolate → inspect state → trace across modules → use logs/tracebacks → fix root cause → prevent regression

AI challenge: audit generated code for logic, security and reliability issues.

GOAL Coordinate tools and build AI systems that can be evaluated and debugged systematically across components.



MONTHS 5–6

AGENTIC WORKFLOWS + AI CREATOR CAPSTONE

05 • AGENTIC WORKFLOWS

multi-step reasoning workflows • tool routing • state • memory concepts
• approvals • observability • recovery

Build an AI workflow that completes a meaningful multi-step task with controls.

06 • AI SYSTEM DESIGN + AI CREATOR CAPSTONE

requirements • architecture diagram • model/data/tools • RAG •
guardrails • evaluation • GitHub • AI Creator capstone

Define → architect → build → test → debug → document → demo.

GOAL Deliver a substantial system whose architecture, tradeoffs and behavior the student can explain, defend and extend.



LEARNING BY BUILDING

Not just demos — AI Creators engineer evaluated AI systems.

Advanced RAG Pipelines

retrieve • rerank • ground

Tool-Using Agents

multiple tools • state • resilience

Agentic Workflows

orchestration • approvals • recovery

AI System Architecture

models • data • tools • guardrails

Agent Debugging Lab

tracing + evaluation + root cause

AI Creator Capstone

architect • build • evaluate • demo

GitHub becomes an engineering portfolio: meaningful commits, README documentation, architecture notes and a polished final project.



AI CREATOR CAPSTONE

Real AI systems. Creator-level ownership.

AGENTIC PRODUCTIVITY SYSTEM

Goals • planning • tool calls • persistent state • approvals • recovery • reporting

Designed as a controlled multi-step agentic system.

ADVANCED RAG INTELLIGENCE HUB

Ingest multiple sources • metadata • retrieval • reranking • grounded synthesis • citations • evaluation

Designed to demonstrate advanced retrieval and system evaluation.

MY OWN IDEA

Student proposes a real system, defines scope and architecture with Ankit, builds a Creator-level version, tests it and documents it on GitHub.

The capstone proves the student can own a system end-to-end: requirements, architecture, implementation, testing, debugging, documentation and demo.



WHAT COMES NEXT

AI Creator is where AI apps become engineered AI systems.

1 • AI EXPLORER

AI literacy checkpoint
prompt + context foundations
verification + AI Lab
AI Explorer projects

2 • AI BUILDER

LLM APIs + RAG
retrieval + tools
evaluation + workflows
AI Builder capstone

3 • AI CREATOR

agentic AI systems
advanced RAG + guardrails
orchestration + evaluation
AI Creator portfolio

Recommended placement: students should demonstrate Builder-level Python and software fundamentals before Creator. Experienced students can take a readiness assessment for direct placement.

Know where your child stands in the age of AI → thinkpythonai.com/will-ai-take-my-kids-job



THINKPYTHONAI YOUNG CODERS

From “I can build AI apps” to “I can engineer AI systems.”

AI CREATOR • 6 MONTHS • LIVE • SMALL SKILL-BASED COHORTS

Architecture • AI Systems Lab • Agent Debugging • Advanced RAG • tools • evaluation • testing • automation • GitHub • AI Creator capstone

Young Coders Coordinator: Crystal
+1 (650) 444-3020

[AI Readiness Quiz: thinkpythonai.com/will-ai-take-my-kids-job](https://thinkpythonai.com/will-ai-take-my-kids-job)

THE AI CREATOR PROMISE

We move beyond isolated AI applications.

Students learn to architect connected AI systems.

They integrate APIs, automate workflows, test deliberately and debug across components.

They leave with portfolio AI systems they can explain, defend and extend.

Strong Python skills are the foundation for building reliable AI systems — not just using AI tools.