

MAHONING COUNTY CAREER & TECHNICAL CENTER

Programming

Syllabus & Curriculum Map · ODE Subject Code 145060

AI Automation & Software Development | Junior Year, Semester 1

2026–2027 · September 8, 2026 – January 29, 2027

Version 2.0 · Instructor: Michael Sekol · michael.sekol@mahoningctc.com

CONCEPT What this course is

The basics of building simple interactive applications. Sequence, selection, and loop, applied to problems you actually have.

You will finish having shipped working software to a public URL, built a text adventure with an AI that writes part of it, reviewed code you did not write, and documented every decision you made.

CHECKPOINT College credit

Programming 145060 is CTAG-eligible for college credit at Ohio public colleges and universities. Credit is triggered by the acknowledgement survey at the end of your WebXam post-test. Say yes.

Course Overview

Official course description

In this course, students will learn the basics of building simple interactive applications. Students will learn the basic units of logic: sequence, selection, and loop. Students will apply algorithmic solutions to problem-domain scenarios. Students will gain experience in using commercial and open-source languages, programs, and applications.

— Ohio Department of Education & Workforce, Information Technology Career Field

Where this sits in your two years

YEAR	SEMESTER 1	SEMESTER 2	LANGUAGE
Junior	145060 Programming	145065 Object-Oriented Programming	Python → Python + C#
Senior	145130 Applications of AI	145010 Web Design + Capstone	C# + Python → Web

This is the entry point. Every convention installed here — the Git ritual, the three gates, the AI usage log, DMAIC — carries through all four courses and into your senior capstone.

Structure

Length	19 calendar weeks, approximately 18 instructional. Thanksgiving week is two days.
Grading Period 1	September 8 – November 6 · 9 weeks · Units 0–4
Grading Period 2	November 9 – January 29 · 9 weeks plus a partial · Units 5–8
Daily block	Periods 1–3, 8:00–10:22. About 118 working minutes after announcements.
Language	Python 3.14, managed with uv
Editor	VS Code
Deployment	Render — free tier, no credit card required, includes PostgreSQL
Version control	Git and GitHub, from Week 1

Assessments and hard dates

ASSESSMENT	WINDOW
WebXam 145060 Pretest	Week 3 — September 21–25. Fall window closes October 2.
Congressional App Challenge	October 26, 12:00 pm ET
BPA Regional pre-submission	November 24

ASSESSMENT	WINDOW
BPA Regional coding events (MCCTC)	December 7–9
BPA Regional presentation events	December 11
WebXam 145060 Post-test	Week 19 — January 25–29

BPA 355 Python Programming is the primary event alignment for this course. BPA 391 Computer Programming Concepts is the secondary. Note that 390 and 391 swapped numbers for 2026–27.

How Class Runs

The daily block

MIN	SEGMENT	WHAT HAPPENS
15	Bell ringer & Gate 1 rep	A question on the board when you walk in, then a timed rep. No AI, no autocomplete.
15	Instruction	One concept. Live coding, including a deliberate mistake debugged in front of you.
35	Build 1	Hands-on.
5	Reset	Stand up, walk, reset.
40	Build 2	Hands-on or pair work.
8	Commit & close	Push your work. Someone demos or names what broke.

Friday introduces no new content. Fridays are BPA preparation, credential work, side quests, and catch-up.

The three gates

GATE	AI USE	WHAT IT IS
Gate 1 — Closed	None	Short timed reps in a plain editor. Trace, predict, fix. This is where fundamentals get built.
Gate 2 — Adversarial	AI is the opponent	You receive generated code and find what is wrong with it. Scored on caught versus missed.
Gate 3 — Open	Full tooling	Ambitious builds with everything available. Decision log required.

WARNING Where the AI comes from

Commercial AI developer APIs require users to be 18 or older. This course uses locally hosted models running on lab hardware, which carry no such terms and send no data outside the building.

Local models are slower and less capable than cloud models. For a text adventure that generates room descriptions, that is a feature.

Grading

CATEGORY	WEIGHT	WHAT COUNTS
Projects	35%	Major builds, the text adventure, deployed applications, team sprints.
Written & Documentation	20%	Gate 2 reviews, AI usage logs, decision logs, READMEs, technical writing.
Lab & Practice	20%	Gate 1 reps, bell ringers, daily labs, problem drops.
Quiz & Exam	15%	WebXam-aligned quizzes and unit exams.
BPA / Credential / Capstone	10%	Competition prep, credentials, side quests, independent projects.

Policies

- No work is graded that is not in a repository. Every period ends with a commit.
- AI is permitted on Gate 2 and Gate 3 work and prohibited on Gate 1 reps.
- Every project includes an AI usage log: prompt, response, what you changed, why.
- No personal information, real names, or school data enters any AI tool.
- Any project may be revised once within two weeks of return if the original was on time.
- When stuck: struggle 10 minutes, ask a peer, ask an AI, then ask me.
- The only way to fail outright is to submit work you cannot explain.

Unit Sequence

Nine units. PRIMARY tags are 145060 competencies. LATENT tags are 145130 competencies you will be tested on senior year, covered here in advance.

UNIT 0 Onboarding, Version Control & Ground Rules

WEEKS 1–2

DATES Sept 8 – Sept 18

Focus: Set up the environment, install the Git ritual, and establish how AI works in this room.

Students will:

- Configure Python 3.14, uv, VS Code, and Git from scratch
- Create a repository, write a useful README, and commit with meaningful messages
- Write a Problem Inventory: ten real annoyances in your own life
- Evaluate an AI artifact on validity, relevance, authenticity, bias, and hallucinations
- Explain who owns AI-generated output and what licensing applies

Deliverables:

- SQ-01 Hello, Version Control — required before any other work is graded
- Problem Inventory — your project backlog for two years, including your senior capstone
- Infographic Autopsy: find the errors in a polished, widely shared, wrong AI diagram
- First AI usage log

PRIMARY 145060 5.1.8, 5.4.1, 5.4.2, 1.2.1, 1.2.12, 1.4.7, 1.7.13, 1.3.7

LATENT 145130 2.14.3, 2.14.4, 9.6.1, 1.3.3, 7.1.7

UNIT 1 Sequence, Data & Output

WEEKS 2–4

DATES Sept 16 – Oct 2

Focus: The straight line. Variables, types, input, output, and how a computer stores what you give it.

Students will:

- Compare primitive numeric and non-numeric types and identify variable scope
- Write code performing arithmetic and string operations including concatenation, pattern matching, and substring
- Convert between binary, hexadecimal, and decimal; explain ASCII and Unicode
- Read input from the user and from a data file

Deliverables:

- CLI Toolsmith — a command-line tool solving one item from your Problem Inventory
- SQ-02 The Regex Wrangler
- WEBXAM 145060 PRETEST (Week 3)

PRIMARY 145060 5.2.1, 5.2.2, 5.2.3, 5.2.4, 5.5.5, 5.5.6, 5.5.7, 5.4.3, 5.1.1, 5.1.7, 2.3.1, 2.3.2

LATENT 145130 2.14.4

UNIT 2 Selection & Boolean Logic

WEEKS 4–6

DATES Sept 28 – Oct 16

Focus: The fork in the road. Conditions, truth tables, and how machine learning differs from a decision tree.

Students will:

- Explain Boolean logic and solve a truth table
- Write code using logical, relational, and compound conditional operators
- Write conditional and selection control structures, and know when to flatten nesting
- Describe how machine learning and neural networks operate differently than standard decision trees

Deliverables:

- Decision engine project — classify input using nested conditions
- Gate 2 review: a generated validation function with a boundary defect
- Problem drop #1

PRIMARY 145060 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.5, 5.3.7, 5.1.2

LATENT 145130 2.14.1, 2.14.4

UNIT 3 Repetition & the Text Adventure

WEEKS 6–8

DATES Oct 12 – Oct 30

Focus: Loops, iteration, and modeling before coding — taught by building a text adventure. Scope grows as your skills do.

Students will:

- Write code using repetition control structures and nested structures
- Use recursion and explain when it is and is not the right tool
- Model a solution using flowcharts, IPO charts, decision trees, and pseudocode
- Explain how algorithms and data structures are used in information processing

Deliverables:

- TEXT ADVENTURE v1: a game loop, multiple rooms, and win and lose states
- Flowchart and pseudocode produced before any code is written
- CONGRESSIONAL APP CHALLENGE submission — October 26
- SQ-05 Bug Hunt

PRIMARY 145060 5.3.6, 5.3.8, 5.1.2, 5.1.3, 5.4.7

LATENT 145130 2.14.3, 2.11.5

UNIT 4 Functions, Errors & Debugging

WEEKS 8–9

DATES Oct 26 – Nov 6

Focus: Breaking work into pieces and handling the fact that everything eventually fails. The text adventure grows up.

Students will:

- Write functions with parameters, return values, and correct scope
- Code error handling for anticipated and unanticipated failures
- Define test cases and test a program against them
- Select and apply a troubleshooting methodology, then document the problem and verified solution

Deliverables:

- TEXT ADVENTURE v2: refactored into functions, with inventory and full error handling
- A written troubleshooting log using a named methodology
- GP1 ENDS November 6

PRIMARY 145060 5.3.9, 5.3.10, 5.4.4, 5.4.5, 5.4.6, 5.4.7, 5.5.1, 5.5.2

LATENT 145130 2.11.1, 2.11.2, 2.11.3, 2.11.4, 2.11.6, 2.11.7, 2.11.8

UNIT 5 Collections, Files & Real Data

WEEKS 10–12

DATES Nov 9 – Nov 24

Focus: Working with many things at once, choosing the right structure, and getting data out of the real world.

Students will:

- Write code that accesses data repositories and reads and writes CSV and JSON
- Choose between list, dictionary, and set based on the operation you need, and explain the cost of getting it wrong
- Preserve, convert, or migrate existing data files to a new format
- Extract structured data from a website responsibly, respecting robots.txt and rate limits
- Generate reports including calculated fields

Deliverables:

- TEXT ADVENTURE v3: save and load using JSON, world data moved out of code
- Data pipeline: scrape or ingest messy data, clean it, convert format, output a report
- SQ-13 The Scraper
- BPA PRE-SUBMISSION November 24

PRIMARY 145060 5.3.11, 5.5.7, 5.2.2, 5.1.5, 5.5.3

LATENT 145130 2.6.4, 8.3.1, 8.3.2, 8.5.4, 2.8.2, 2.8.9

UNIT 6 SDLC, DMAIC & the Competition Sprint

WEEKS 13–15

DATES Nov 30 – Dec 18

Focus: No new syntax. A full development lifecycle run as a team, timed against real external deadlines.

Students will:

- Determine requirements, identify constraints, and adhere to a timeline
- Compare and contrast agile and waterfall methodologies
- Perform code reviews via peer walkthrough and static analysis
- Develop test cases and acceptance criteria agreed with stakeholders
- Provide group leadership and work in a team

Deliverables:

- Team sprint run through Define, Measure, Analyze, Improve, Control
- Two peer code reviews scored on the five-dimension rubric
- BPA REGIONAL CODING EVENTS December 7–9 · PRESENTATIONS December 11

PRIMARY 145060 5.6.1, 5.6.2, 5.6.3, 5.6.9, 5.6.10, 5.6.11, 5.6.12, 5.6.13, 5.6.14, 5.1.4, 5.1.6, 1.2.3, 1.2.4, 1.2.7, 1.2.10, 1.8.2, 1.5.5, 1.5.8

LATENT 145130 2.11.1–2.11.8, 2.12.1, 2.12.3, 2.12.5, 2.12.6, 2.15.2, 1.8.4, 1.2.14

UNIT 7 Objects, APIs & Local AI

WEEKS 16–18

DATES Jan 4 – Jan 22

Focus: Classes, external data sources, and putting a language model inside your own program. This unit sets up 145065.

Students will:

- Write code that creates classes, objects, and methods
- Consume an external API and handle 404s, timeouts, and rate limits distinctly
- Call a locally hosted language model from your own code and parse its response
- Identify application vulnerabilities and implement secure coding including input validation
- Explain confidentiality, integrity, and availability

Deliverables:

- TEXT ADVENTURE v4: a local model generates room descriptions on demand
- API-consuming application with full failure handling and no hardcoded credentials
- Security review of a deliberately vulnerable codebase
- SQ-10 API First Contact

PRIMARY 145060 5.3.12, 5.5.4, 5.5.7, 9.3.1, 9.3.3, 9.3.4, 9.1.1, 9.1.4, 2.1.1, 3.2.1

LATENT 145130 2.1.12, 2.14.5, 2.14.6, 2.14.2

UNIT 8 Deploy, Document & Demonstrate

WEEKS 19

DATES Jan 25 – Jan 29

Focus: Ship it to a public URL, explain it, and prove you understand it.

Students will:

- Deploy an application to a publicly accessible URL on Render
- Create documentation including an implementation plan and user help
- Explain version management, baseline and lifecycle phases, and the impact of changes
- Deliver a formal presentation of your work

Deliverables:

- Final project deployed live with README, documentation, and decision log
- Five-minute demo: what it does, why you built it this way, what you rejected
- WEBXAM 145060 POST-TEST — say YES to the CTAG acknowledgement survey
- GP2 ENDS January 29

PRIMARY 145060 5.6.8, 5.6.16, 5.6.17, 5.7.1, 5.7.2, 5.7.3, 1.2.2, 1.2.11, 1.2.5

LATENT 145130 7.1.9, 7.3.1, 2.14.5

Semester Calendar

Grading Period 1 • September 8 – November 6 • 9 weeks

WK	DATES	UNIT	MILESTONES
1	Sept 8–11	Unit 0	First day Sept 8. Setup, SQ-01, Problem Inventory.
2	Sept 14–18	Unit 0 / 1	Infographic Autopsy. Git ritual begins.
3	Sept 21–25	Unit 1	WEBXAM PRETEST. Window closes Oct 2.
4	Sept 28 – Oct 2	Unit 1 / 2	CLI Toolsmith due. CAC teams form.
5	Oct 5–9	Unit 2	Oct 9 NEOEA Day, no classes.
6	Oct 12–16	Unit 2 / 3	Oct 16 Waiver Day, no classes. Problem drop #1. Text adventure opens.
7	Oct 19–23	Unit 3	Text adventure v1 build. CAC final week.
8	Oct 26–30	Unit 3 / 4	CAC DEADLINE Oct 26. Text adventure v1 due.
9	Nov 2–6	Unit 4	Text adventure v2 due. GP1 ENDS Nov 6.

Grading Period 2 • November 9 – January 29

WK	DATES	UNIT	MILESTONES
10	Nov 9–13	Unit 5	Collections. NASA Space Apps Nov 14–15 (weekend).
11	Nov 16–20	Unit 5	Files, CSV, JSON, scraping. Nov 19 conferences.
12	Nov 23–24	Unit 5 close	BPA PRE-SUBMISSION Nov 24. Break Nov 25–27. Two-day week.
13	Nov 30 – Dec 4	Unit 6	Team sprint opens. DMAIC Define and Measure.
14	Dec 7–11	Unit 6	BPA REGIONAL CODING Dec 7–9. PRESENTATIONS Dec 11.
15	Dec 14–18	Unit 6 close	Retrospective. Banquet Dec 17. Break begins Dec 21.
16	Jan 4–8	Unit 7	Return. Classes and objects.
17	Jan 11–15	Unit 7	APIs and local model. Problem drop #4.
18	Jan 18–22	Unit 7	Jan 18 MLK, no classes. Secure coding. WebXam review begins.

WK	DATES	UNIT	MILESTONES
19	Jan 25–29	Unit 8	Deploy to Render. WEBXAM POST-TEST. Demos. GP2 ENDS.

CONCEPT Why the text adventure runs four units

A text adventure needs loops, conditionals, functions, dictionaries, file handling, and eventually an API. It is not a decoration on the curriculum — it is the curriculum, in a form you will actually want to finish.

Version 1 in October is rooms and a game loop. Version 4 in January has a language model writing descriptions. Same project, four times the student.

Competency Coverage

Primary — 145060

STRAND	NAME	UNITS
1	Business Operations / 21st Century Skills	0, 6, 8
2	IT Fundamentals	1, 7
3	Information Security	7
5	Programming & Software Systems	0, 1, 2, 3, 4, 5, 6, 7, 8
9	Cybersecurity	7

Latent — 145130 coverage carried here

You will take 145130 Applications of Artificial Intelligence in senior year and be tested on these outcomes. Item counts are from the published WebXam 145130 blueprint.

OUTCOME	NAME	EXAM ITEMS	COVERED IN
2.14	Artificial Intelligence	21	0, 1, 2, 3, 7
9.6	Cybersecurity Law	6	0
8.5	Queries & Transactions	6	5
2.1	Security, Risks & Safeguards	5	7
8.3	Data Entry & Access	5	5
2.11	Troubleshooting	4	3, 4, 6
2.6	Installation & Configuration	4	5
2.15	UX/UI Design	4	6
7.1	Interactive Media	4	0, 8
7.3	Production	3	8
2.8	Databases	3	5
1.8	Operations Management	0	6

CHECKPOINT What you can do when you leave

Build and deploy a working Python application to a public URL. Read and critique code you did not write. Handle failure gracefully. Use Git without thinking about it. Put a language model inside your own program. And explain every decision you made, including the ones an AI helped with.

Version 2.0 · Revised August 2026 · Supersedes Version 1.0. Pairs with the Course Prompt Guide, Instructor Guide, Resource Library, and Program Style & Reference Guide.