

FUNDAMENTALS OF AI-AUGMENTED ENGINEERING

CHAPTER 2 / HOMEWORK 2

Engineering Homework

The Cooling Gear — Predicting a Safe-Handling Time

Think → Pseudocode → AI → Code · Runs ≠ Correct

Suggested Assessment · 20 points

target effort ≈ 75–90 minutes · type all answers and paste all screenshots directly into this document



AI First Engineering

Figure: AI First
Engineering
(AIFE) course
logo

Taner Onsay, Ph.D. & Sebnem Onsay · © 2026 AI First Engineering ·

AI use is allowed and encouraged — as a coding companion. But your algorithm and pseudocode must exist BEFORE AI sees them, and every screenshot must show YOUR OWN Python session, running on your computer — not an AI transcript. Assessment rewards evidence + reasoning + verification, not polished prose.

THE ENGINEERING CONTEXT — an automotive steel gear

The steel gear pictured below — about 2.5 inches (64 mm) in diameter — comes out of a thermal-processing operation hot, at about 180 °C. Before it can be safely handled or move to the next manufacturing operation, its temperature must fall below an allowable handling temperature of 45 °C. The shop air around the cooling rack is at 22 °C.

The production plan allots a 30-minute cooling slot. An engineer must predict: **how long must the gear cool before it reaches the allowable temperature — and is the 30-minute plan enough?**

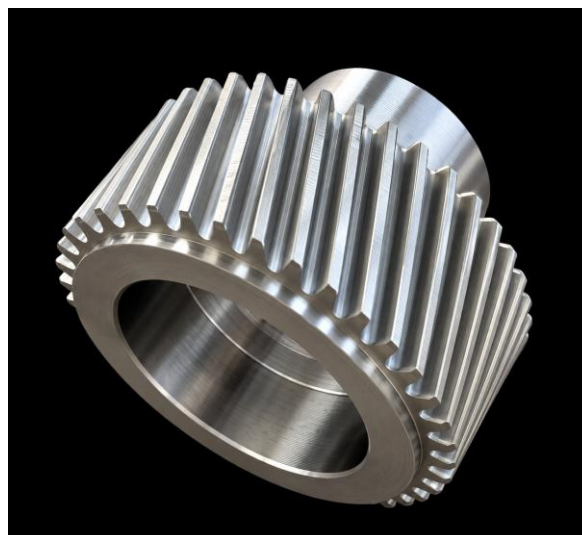


Figure: Automotive steel gear used as the physical inspiration for this computational cooling problem.

THE COOLING MODEL — provided; no heat-transfer theory required

A hot object in still air loses temperature fastest when it is much hotter than the air, and more and more slowly as it approaches air temperature. For this homework, use this rule:

During each short time interval dt , the gear loses the fraction $k \cdot dt$ of its current temperature difference above the surrounding air.

$$T_{\text{new}} = T - k \cdot (T - T_{\text{amb}}) \cdot dt$$

T = gear temperature ($^{\circ}\text{C}$) · T_{amb} = air temperature ($^{\circ}\text{C}$) · k = cooling parameter (1/min) · dt = time step (min)

The numbers below are instructional model parameters chosen for this assignment. They were NOT measured from the pictured gear. Everything you compute here is a MODEL PREDICTION — not yet experimental evidence. That distinction will matter later in the course.

Quantity	Symbol	Value	Units
Starting gear temperature	T_{start}	180	$^{\circ}\text{C}$
Surrounding air temperature	T_{amb}	22	$^{\circ}\text{C}$
Allowable handling temperature	T_{allow}	45	$^{\circ}\text{C}$
Cooling parameter	k	0.06	1/min
Time step	dt	1.0	min

Time guide: Part A $\approx$ 10 min · Part B $\approx$ 15 min · Part C $\approx$ 10 min · Part D $\approx$ 10 min · Part E $\approx$ 15 min · Part F $\approx$ 15 min · Part G $\approx$ 15 min. If you are far past 90 minutes, submit what you have with honest notes — do not expand the assignment.

PART A · Understand & Predict — No Computing Yet ($\approx$ 10 min)

Read the context and model boxes above. Then commit — in writing, before any computation. A prediction made after seeing the answer is not a prediction. Wrong is fine; uncommitted is not. Do not edit these later.

A1 — The Problem, in Your Own Words. In one or two sentences: what exactly are you asking the computer to calculate, and what engineering decision depends on the result?

ANSWER A1:

A2 — Prediction P1 · Cooling time. Your committed estimate of the time for the gear to cool from 180°C to 45°C :

P1: predicted cooling time $\approx$ _____ minutes

A3 — Prediction P2 · Shape. Sketch the temperature-versus-time behavior in your mind. Which do you expect?

P2: straight line ☐ **curve, cooling fastest at the START** ☐ **curve, cooling fastest at the END** ☐

A4 — Prediction P3 · Halves. Which takes LONGER: cooling from 180 °C down to 110 °C, or from 110 °C down to 45 °C? Circle one and give one sentence of physical reasoning.

P3: 180 $\rightarrow$ 110 takes longer ☐ **110 $\rightarrow$ 45 takes longer** ☐ **about the same** ☐

REASONING P3:

PART B · Model & Algorithm — Before Any Code ($\approx$ 15 min)

The intellectual work of this homework happens here: turning a physical rule into a computational procedure. Answer on paper — still no code.

B1 — Quantities and Roles. Complete the Role column: is each quantity an input parameter (set once, never changes), a state variable (changes every pass of the computation), or an output (reported at the end)?

Quantity	Symbol	Units	Role — your answer
Starting gear temperature	T_start	°C	
Surrounding air temperature	T_amb	°C	
Allowable handling temperature	T_allow	°C	
Cooling parameter	k	1/min	
Time step	dt	min	
Gear temperature during cooling	T	°C	
Elapsed cooling time	t	min	

B2 — One Update, by Hand. Using the model rule and the parameter table, hand-calculate the gear temperature after the FIRST minute of cooling ($T = 180$ °C at $t = 0$). Show the arithmetic. Keep this number — it is your known case for testing every program that follows.

ANSWER B2: T after 1 minute = _____ °C. Arithmetic:

B3 — Which Loop, and Why. Does this computation want a FOR loop or a WHILE loop? One or two sentences, using the lecture distinction: do you know the cases in advance, or do you know the condition?

ANSWER B3:

B4 — Stopping Criterion & the Boundary Call. State precisely the condition under which the repetition stops. Then make the boundary call: if the computed temperature lands EXACTLY on 45.0 °C, is the gear safe to handle? The choice ($\leq$ or $<$) is yours — write it down and defend it in one sentence.

ANSWER B4: Stop when: _____. Boundary rule and reason:

B5 — Why Must the Loop End? One sentence: what inside the loop body guarantees the stopping condition is eventually reached? (Recall the heater that never stopped.)

ANSWER B5:

PART C · Pseudocode — Yours, Before AI (≈ 10 min)

Write complete pseudocode for the cool-down calculation. AI has seen nothing yet — you are at step 2 of THINK → PSEUDOCODE → AI → CODE. Any wording works; precision is the only rule.

Your pseudocode must contain: named inputs with units · the temperature update as a formula · time accumulation · a WHILE with your explicit stopping criterion from B4 · the reported output, with units.

```
# SKELETON — shape only; you supply the substance
GET  T_start [°C] · T_amb [°C] · T_allow [°C] · k [1/min] · dt [min]
SET  T = T_start,  t = 0
WHILE ( ...the gear is still too hot to handle... ):
    ...
REPORT ...the predicted cooling time, with units...
```

WRITE (OR PASTE) YOUR PSEUDOCODE HERE

(type your pseudocode directly, or paste a photo of your handwritten version)

PART D · AI Coding Companion — Critique, Not Replacement (≈ 10 min)

Now — and only now — hand your pseudocode to any capable AI assistant. Use a prompt close to this:

“Here is my pseudocode for a gear cool-down calculation. Do not write Python yet. Critique the logic only — unclear steps, unit risks, my stopping criterion, anything ambiguous. Ask me questions if my intent is unclear.”

If AI writes code anyway: “I asked for critique only.” For each point AI raises, ask: is it right, for MY problem?

D1 — One Suggestion, Recorded. Copy ONE meaningful AI suggestion about your pseudocode, verbatim (one or two lines — not the whole conversation).

ANSWER D1:

D2 — Your Verdict. Judge that suggestion as the engineer. Do not manufacture disagreement — ACCEPT is a fully legitimate verdict when independently justified.

ACCEPT ☐ **MODIFY** ☐ **REJECT** ☐

WHY (1–2 sentences):

PART E · Python — Implement Your Algorithm (≈ 15 min)

Translate YOUR pseudocode into Python (AI may help translate — but read every line until you can explain it to a classmate; unexplainable code does not enter your program). Keep it short: a dozen lines is right-sized.

Required structure: put the calculation in one function with this signature, then call it and print the result — a named, testable engineering task, exactly as in lecture:

```
def cooling_time(T_start, T_amb, T_allow, k, dt):  
    # your WHILE search from Part C lives here  
    # return the predicted cooling time in minutes  
  
print(cooling_time(180.0, 22.0, 45.0, 0.06, 1.0))
```

Before you run it, write down: your expected neighborhood for the answer (your P1), and your hand value from B2 for the first minute. Then run, and compare.

E1 — Evidence. Paste ONE screenshot showing your Python code together with its printed result, with no unresolved error.

PASTE PYTHON CODE / OUTPUT SCREENSHOT HERE

(click here and paste your screenshot — Ctrl+V / Cmd+V)

E2 — Result. Report your computed answer and check it against your own evidence.

Predicted cooling time (dt = 1.0 min): _____ minutes

First computed minute agrees with your hand value from B2? YES ☐
NO ☐

If NO — what you checked (one sentence):

PART F · BREAK THE AI — The Gear That Would Not Cool (≈ 15 min)

A student asked an AI assistant to “write a program that finds how long the gear takes to reach the allowable temperature.” The AI produced the program below. It is short, professional-looking, and syntactically correct. It contains one quiet flaw.

```
# gear_cooldown.py — AI-generated version (contains one quiet flaw)
T_start, T_amb, T_allow = 180.0, 22.0, 45.0    # deg C
k, dt = 0.06, 1.0                            # 1/min, min

T = T_start
t = 0.0
while T > T_allow:
    T = T - k * (T - T_allow) * dt
    t = t + dt

print("Cooling time:", t, "minutes")
```

If a program seems to run forever, that is not a disaster — it is evidence.

Interrupt the kernel (you saw how in lecture), and write down what you observed. A temporary `print(t, T)` inside the loop is a legitimate instrument.

F1 — Known Case First. Before running anything: trace the AI program’s FIRST minute by hand. What temperature does ITS update line produce after one minute? Compare with your own B2 hand value.

ANSWER F1: AI program after 1 min = _____ °C. Your model (B2) = _____ °C. Agree? YES ☐ NO ☐

F2 — Run & Observe. Run the AI program exactly as given. What happens? What does the temperature do over time? Report the evidence — not a guess.

ANSWER F2:

F3 — Diagnose from Evidence. Which line is wrong? What physical situation does the program ACTUALLY model — toward what temperature is this gear cooling? Which of the four correctness levels does it fail (runs / implements the algorithm / represents the engineering problem / validated)? And why can this loop never end? (2–3 sentences.)

ANSWER F3:

F4 — Correct & Retest. Fix the flawed line, rerun, and confirm the corrected program reproduces your Part E result. Record the corrected line and the retest result.

ANSWER F4: Corrected line: _____. Retest result: _____ minutes

PART G · Validate & Decide (≈ 15 min)

Your program runs and your BREAK-THE-AI hunt is closed. Running was only Level 1. This part earns the other three levels — with independent evidence.

INDEPENDENT VALIDATION / REFERENCE SOLUTION — used only NOW

The cooling model has an exact continuous-time solution. You are NOT expected to derive it; it exists here purely as independent evidence against your computational answer:

$$t_{\text{exact}} = \ln((T_{\text{start}} - T_{\text{amb}}) / (T_{\text{allow}} - T_{\text{amb}})) / k$$

One line of Python: `import math; print( math.log((180-22)/(45-22)) / 0.06 )`

G1 — Reference Check. Compute the reference value and compare with your Part E answer.

$t_{\text{exact}} =$ _____ min your search (dt = 1.0) = _____ min
difference = _____ min

G2 — Step Size (a seed for Chapter 3). Rerun your own program once with dt = 0.1 min. Record all three values. Both of your runs executed correctly — yet the numbers are not identical. In 1–2 sentences: why?

dt = 1.0 → _____ min dt = 0.1 → _____ min reference → _____ min

ANSWER G2:

G3 — Physical Reasonableness. Look at your program's temperature history (print T as it cools, or recall your trace). Does it behave like a real hot object — falling fastest at first, approaching but never passing below room temperature? Which of your predictions P2 and P3 survived contact with the computation? (1–2 sentences.)

ANSWER G3:

G4 — One Parameter Change: a Summer Shop Floor. In summer the shop air reaches $T_{\text{amb}} = 35^\circ\text{C}$. BEFORE rerunning, commit: the predicted cooling time will...

increase a little ☐ increase a lot ☐ decrease ☐

Now change the single parameter and rerun: summer cooling time = _____ minutes.

One sentence — why is the effect this large?

G5 — The Engineering Decision. Complete the decision statement. Every blank matters.

“Under the stated cooling model and assumptions ($T_{\text{start}} = 180\text{ }^{\circ}\text{C}$, $T_{\text{amb}} = \underline{\hspace{1cm}}$ $^{\circ}\text{C}$, $T_{\text{allow}} = 45\text{ }^{\circ}\text{C}$, $k = 0.06\text{ 1/min}$, boundary rule: $\underline{\hspace{1cm}}$), the model predicts the gear reaches the allowable handling temperature after approximately $\underline{\hspace{1cm}}$ minutes.

*The production plan’s 30-minute cooling slot IS ☐ / IS NOT ☐ sufficient.
I recommend:*

_____.
This is a model-based prediction under the stated assumptions — it is not yet experimental evidence about the physical gear.”

Evidence behind this decision (one line each): ① hand-checked known case ② reference-solution agreement ③ physical-reasonableness checks.

EVIDENCE ①:

EVIDENCE ②:

EVIDENCE ③:

WE WILL RETURN TO THIS GEAR.

Later in ENGR 200, we will return to this problem using experimental temperature-versus-time measurements from an actual automotive component and ask a harder question: **does the computational cooling model actually describe the physical system — and how do we know?**

SUBMIT · Evidence Checklist & Human Review (≈ 5 min)

Submit ONE completed Word document — this file — containing all of the following:

- ☐ A1 problem statement in your own words
- ☐ Committed predictions P1, P2, P3 — unedited
- ☐ B1 roles table · B2 hand-calculated first minute · B3 loop choice · B4 stopping criterion and boundary rule · B5 termination argument
- ☐ Your pseudocode — written before any AI conversation (Part C)
- ☐ D1 recorded AI suggestion and D2 verdict with justification
- ☐ Python code + output screenshot (E1) and reported result (E2)
- ☐ BREAK THE AI: F1 hand trace · F2 observation · F3 diagnosis · F4 corrected line and retest
- ☐ G1 reference comparison · G2 step-size values and interpretation · G3 physical checks · G4 summer rerun
- ☐ G5 completed engineering-decision statement with evidence lines
- ☐ Human review statement signed and dated (below)

SUBMIT ONE COMPLETED HOMEWORK DOCUMENT

Keep all required evidence in this document rather than as separate Python or image files. A separate .py file or notebook is welcome as an optional attachment, but the document must stand on its own. If this homework is being used as part of a course, follow your instructor's submission and file-naming requirements.

HUMAN REVIEW

"I reviewed the predictions, calculations, pseudocode, code, screenshots, and written responses in this submission and confirm that they represent the work I performed, and that I can explain every line of the program I submitted."

Human reviewed and approved: _____

Student name: _____ Date: _____

Suggested Assessment Rubric — 20 points

Used by the instructor and, where applicable, by an AI grading agent. All required evidence must appear inside this document — the grader will not open separate Python or image files.

Criterion	What full credit looks like	Pts
Problem framing & committed predictions (A1–A4)	A1 states the computed quantity and the decision that depends on it. P1–P3 committed with a number/choice each; P3 carries one sentence of physical reasoning. Prediction correctness is NOT graded — commitment and reasoning are.	2
Model formation (B1–B5)	Roles correctly assigned (T and t are the state variables; the rest are parameters/inputs; t and final answer are outputs). B2 = 170.52 °C (± 0.1 , arithmetic shown). B3 chooses WHILE because the pass count is unknown and the condition is known. B4 states an explicit stop ($T \leq 45$ or $T < 45$) with a defended boundary rule. B5: each pass moves T toward T_amb, so the condition must eventually fail.	4
Pseudocode (Part C)	Complete and ordered: named inputs with units, the update formula, time accumulation, a WHILE with the student's explicit stopping criterion, a reported output with units. Written before AI critique.	3
AI critique & verdict (Part D)	One concrete suggestion recorded, plus an ACCEPT / MODIFY / REJECT verdict with independent justification. ACCEPT with verification earns full credit; manufactured disagreement does not.	2
Implementation (Part E)	cooling_time() function present and short; screenshot shows code + clean output; reported result ≈ 32 minutes for dt = 1.0; first computed minute consistent with B2.	3
BREAK THE AI (Part F)	Evidence-first trail: F1 = 171.90 °C vs 170.52 °C; F2 reports non-termination / temperature stalling just above 45 °C; F3 identifies the update line cooling toward	3

Criterion	What full credit looks like	Pts
	T_allow instead of T_amb — a Level-3 failure (wrong engineering model) — and explains why the stop is unreachable; F4 corrected and retested to ≈ 32 minutes.	
Validation & decision (Part G)	G1 reference ≈ 32.12 min compared honestly. G2 records ≈ 32.0 vs ≈ 32.1 vs 32.12 and attributes the gap to time-step resolution (not a bug). G3 physical checks tied back to P2/P3. G4 summer rerun ≈ 44 min with the shrinking ($T_{\text{allow}} - T_{\text{amb}}$) gap named. G5 decision: 30-minute slot insufficient, a margin-bearing recommendation, and the model-prediction caveat retained.	3
	TOTAL	20