

# FUNDAMENTALS OF AI-AUGMENTED ENGINEERING

## CHAPTER 1 / PART 1

### Engineering Homework

#### One Model, Two Computational Tools

Excel → Python → Compare

#### **Suggested Assessment · 10 points**

target effort ≈ 20–30 minutes · type all answers and paste all screenshots directly into this document



Figure: AI First Engineering (AIFE) course logo

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

**AI use is allowed and encouraged — as an assistant.** But every screenshot must show YOUR OWN Excel worksheet and YOUR OWN Python session, running on your computer — not an AI transcript. Assessment rewards evidence + reasoning + verification, not polished prose.

#### THE ENGINEERING MODEL — ideal free-fall time

$$t = \sqrt{(2h / g)}$$

$t$  = predicted fall time (seconds) ·  $h$  = drop height (meters) ·  $g = 9.81 \text{ m/s}^2$

In this homework you will implement the SAME engineering model in two computational tools — Microsoft Excel and Python. You will generate predictions over a range of heights, create a graph in each tool, and compare the two implementations. **No new physical experiment is required.**

*Time guide: Part A ≈ 8–10 min · Part B ≈ 5 min · Part C ≈ 8–10 min · Part D ≈ 3–5 min · Part E ≈ 2–3 min. If you are far past 30 minutes, submit what you have with honest notes — do not expand the assignment.*

### PART A · Excel — Build the Calculation (≈ 8–10 min)

Create a new, blank Excel worksheet and build the free-fall table from scratch:

1. At the top of the worksheet, enter the gravitational constant in three adjacent cells, like this:

| Variable | Value | Units            |
|----------|-------|------------------|
| <b>g</b> | 9.81  | m/s <sup>2</sup> |

2. Below it, create a two-column data table with the headers shown here:

| h (m) | t (s)                    |
|-------|--------------------------|
| 1.0   | ← your formula goes here |
| 1.1   |                          |
| ⋮     |                          |
| 2.0   |                          |

3. In the h column, enter heights 1.0, 1.1, 1.2, ... , 2.0 m — increments of 0.1 m, 11 rows in total. (Tip: type 1.0 and 1.1, select both cells, and drag the fill handle down.)
4. In the first t cell, enter the free-fall equation as an Excel formula (it must start with =). Use SQRT( ) and reference the cell that contains g — do NOT type 9.81 directly into the formula.
5. Fill/copy the formula down the column so every height has a calculated fall time. Do not type a separate calculation into each row.
6. Check your table: fall times should increase as height increases (from about 0.45 s up to about 0.64 s).

**About absolute references:** when you copy a formula down a column, Excel normally shifts every cell reference down with it. Writing the reference to g with dollar signs — for example **\$B\$2** — “locks” that reference so every copied row still points at the single cell holding g. This is called an absolute (fixed) cell reference, and it is one convenient way to complete step 5.

**A1 — Excel Evidence.** Paste ONE screenshot of your Excel worksheet. It must clearly show: the  $g = 9.81 \text{ m/s}^2$  entry, the h column, the t column, all heights from 1.0 through 2.0 m, and the calculated fall times.

**PASTE EXCEL TABLE SCREENSHOT HERE**

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

**A2 — Formula.** What Excel formula did you use to calculate fall time? Type or paste the actual formula, exactly as entered in Excel (first t cell).

**ANSWER A2:**

## PART B · Excel — Create the Graph (≈ 5 min)

---

Using your Part A table, create an **XY Scatter chart**:

1. Select the h and t columns and insert an XY Scatter chart (Insert → Charts → Scatter).
2. Horizontal axis: Height, h (m). Vertical axis: Predicted Fall Time, t (s). Axis labels must include units.
3. Give the chart a meaningful title, for example: Predicted Fall Time vs. Height.  
*No trendline, curve fitting, or other advanced analysis is required.*

**B1 — Excel Graph Evidence.** Paste a screenshot of your completed Excel graph here.

**PASTE EXCEL GRAPH HERE**

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

**B2 — Observe the Graph.** As height increases from 1.0 to 2.0 m, does fall time increase, decrease, or remain constant? Does the graph appear perfectly linear? (No more than 2 sentences.)

**ANSWER B2:**

---

## PART C · Python — Modify and Run the Program (≈ 8–10 min)

---

Copy the starter program below into your Python environment (IDLE, Spyder, Jupyter, VS Code — any is fine). The starter program already runs. Your job is only to make the small changes marked **MODIFY** and **ADD**. The objective is not Python mastery — it is: open Python → modify working code → run it → inspect the result.

```

# free_fall.py – One Model, Two Tools (Chapter 1)
import math
import matplotlib.pyplot as plt

g = 9.81                                # gravitational acceleration, m/s^2

# MODIFY: make this list contain 1.0 through 2.0 m in steps of 0.1 m
#         (11 values – typing all 11 numbers is perfectly fine)
heights = [1.0, 2.0]

times = []                               # predicted fall times will be stored here

for h in heights:
    t = math.sqrt(2 * h / g)             # the free-fall model:  $t = \sqrt{2h/g}$ 
    times.append(t)
    print(f"h = {h:.1f} m      t = {t:.3f} s")

# ADD: uncomment the five lines below (delete each leading #) to make the graph
# plt.plot(heights, times, "o-")
# plt.xlabel("Height, h (m)")
# plt.ylabel("Predicted Fall Time, t (s)")
# plt.title("Predicted Fall Time vs. Height")
# plt.show()

```

**Your three changes:** (1) MODIFY the heights list so it contains 1.0 through 2.0 m in steps of 0.1 m; (2) run the program and check the printed fall times against your Excel table; (3) ADD the graph by uncommenting the five plotting lines and running again.

*If Python reports that matplotlib is not installed, install the package using your Python environment's package installation method (for example: `pip install matplotlib`).*

**C1 — Python Evidence.** Paste ONE screenshot showing your modified Python code together with its successful printed output (the 11 height–time pairs), with no unresolved error.

### PASTE PYTHON CODE / OUTPUT SCREENSHOT HERE

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

**C2 — Python Graph.** Paste the Python-generated graph of Predicted Fall Time vs. Height. Axes and units should correspond to your Excel graph.

## PASTE PYTHON GRAPH HERE

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

### PART D · One Model, Two Tools (≈ 3–5 min)

**D1 — Numerical Comparison.** Choose ONE height from your table and report both predictions:

**Selected height:**  $h =$  \_\_\_\_\_ m

**Excel predicted fall time:**  $t =$  \_\_\_\_\_ s

**Python predicted fall time:**  $t =$  \_\_\_\_\_ s

**Do the Excel and Python values agree to the precision shown?**

YES ☐ NO ☐

*If NO — one sentence on what you checked:*

**D2 — Engineering Judgment.** Excel and Python give the same predicted fall times. Does this agreement prove that the physical free-fall model  $t = \sqrt{2h/g}$  correctly represents reality? In your answer, distinguish what the Excel–Python agreement DOES establish from what it DOES NOT establish. (2–3 sentences maximum.)

**ANSWER D2:**

### PART E · AI as a Coding Companion (≈ 2–3 min)

Pick **one line or feature** of the Python program or your Excel formula that you did not completely understand at first — for example append, the `print(f"...")` line, `$B$2`, or `SQRT`. Ask any capable AI assistant (your Engineering Companion) to explain it. Do not paste the full transcript here.

**E1 — What did you ask AI about?** (one sentence)

**STUDENT RESPONSE:**

**E2 — What did you learn?** (1–2 sentences)

**STUDENT RESPONSE:**

## SUBMIT · Evidence Checklist & Human Review (≈ 2 min)

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

- Excel table screenshot (g entry, all 11 heights, calculated fall times) — Part A1
- Actual Excel formula typed in ANSWER A2
- Excel graph screenshot — Part B1
- B2 graph observation ( $\leq 2$  sentences)
- Python code + output screenshot — Part C1
- Python graph — Part C2
- D1 Excel/Python numerical comparison completed
- D2 implementation-check vs. physical-validation answer (2–3 sentences)
- Part E AI coding-companion responses
- Human review statement signed and dated (below)

### SUBMIT ONE COMPLETED HOMEWORK DOCUMENT

Keep all required evidence in this document rather than as separate Excel, Python, or image files. 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 calculations, screenshots, plots, and written responses in this submission and confirm that they represent the work I performed."*

Human reviewed and approved: \_\_\_\_\_

Student name: \_\_\_\_\_ Date: \_\_\_\_\_

## Suggested Assessment Rubric — 10 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 Excel, Python, or image files.

| Criterion                                         | What full credit looks like                                                                                                                                                                                                                       | Pts |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Excel setup, formula, and calculated table        | Screenshot shows $g = 9.81$ entered once with units; $h = 1.0\text{--}2.0$ m in $0.1$ m steps (11 rows); plausible increasing $t$ values ( $\approx 0.45\text{--}0.64$ s). A2 formula uses SQRT (or $^{\wedge}0.5$ ) and references the $g$ cell. | 2   |
| Excel graph correctly created and labeled         | XY scatter of $t$ vs. $h$ ; both axes labeled with units; meaningful title; increasing, gently flattening shape.                                                                                                                                  | 1   |
| Python modification executes and produces results | Screenshot shows modified code and printed $h\text{--}t$ pairs for all 11 heights with no unresolved error.                                                                                                                                       | 2   |
| Python graph correctly created and labeled        | matplotlib plot of $t$ vs. $h$ with axis labels (units) and title, corresponding to the Excel graph.                                                                                                                                              | 1   |

| Criterion                                         | What full credit looks like                                                                                                                                                                                | Pts       |
|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Excel/Python numerical comparison (D1)            | One height selected; both values reported; agreement correctly assessed (YES expected — a justified NO with a checked explanation also earns credit).                                                      | 1         |
| Implementation check vs. physical validation (D2) | States that agreement shows both tools evaluate the same equation consistently (an implementation check) and does NOT prove the model matches reality; physical validation requires experimental evidence. | 2         |
| AI coding-companion reflection + human review     | One concrete question and a concise statement of what was learned; human-review statement signed and dated.                                                                                                | 1         |
|                                                   | <b>TOTAL</b>                                                                                                                                                                                               | <b>10</b> |