LESSON 90

Data Processing and Algorithm Implementation

240 minutes (6 periods) | Applied Learning Tasks (ALTs) | Outcome 3.4 | Outcome 3.5 | Outcome 3.6 | Outcome 3.7 | Outcome 1.22 | Outcome 1.23 | Outcome 2.7

WHAT THIS LESSON DOES

In this lesson, you'll learn to process data from your previous work and implement algorithms to analyse it. Follow guided steps to review, clean, and explore your data, code algorithms, evaluate results, and document findings for initial insights.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.4	develop algorithms that can find the frequency, mean, median and mode of a data set
3.5	structure and transform raw data to prepare it for analysis
3.6	represent data to effectively communicate in a graphical form
3.7	use algorithms to analyse and interpret data in a way that informs decision-making
1.22	read, write, test, and modify computer programs
1.23	reflect and communicate on the design and development process
2.7	implement algorithms using a programming language to solve a range of problems

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the process of reviewing and cleaning data to ensure accuracy and relevance for analysis.
- Develop skills in exploratory data analysis to identify patterns and trends using statistical methods and visualisations.
- Implement algorithms in Python to calculate key statistics and analyse datasets effectively.
- Evaluate and interpret analytical results to draw meaningful conclusions related to a hypothesis.
- Document processes, code, and findings clearly to support project development and communication.

BEFORE THE LESSON

- Confirm Python runs on every machine and that each student's own dataset file (CSV or list) from earlier work is saved where they can reach it in class.
- Check whether matplotlib is installed behind the school filter. If it is not, the lesson works with the text-based histogram, so plan around that rather than assume plots.
- Have a small clean sample dataset ready so students can test their statistics functions before running them on their real, messier data.

Data Processing and Algorithm Implementation

HOW THE LESSON RUNS

Introduction
Review Collected Data
Clean the Data
Perform Exploratory Data Analysis
Implement Algorithms
Evaluate Initial Results
Document Progress

TEACHING MOVES

- 2. Review Collected Data. Push students to load and print a summary before touching anything. Have them write the actual issues they find into a text file now, not from memory. Missing entries and mixed date formats are the usual culprits.
- 3. Clean the Data. Make them justify each cleaning choice in writing. Deleting rows versus filling with an average changes the result, and the project is marked on reasoning. Warn against silently dropping data that makes the hypothesis look better.
- 4. Perform Exploratory Data Analysis. Insist they say what each statistic tells them about their hypothesis, not just print a number. A mean with no interpretation is not analysis. The text histogram is fine if matplotlib is unavailable.
- 5. Implement Algorithms. Have them test each function on the known sample first, where they can predict the answer by hand. A median function that fails on an even-length list is common and invisible until checked.
- 6. Evaluate Initial Results. Ask directly: does this support your hypothesis, or not? Results that contradict the hypothesis are valid and worth full marks when honestly reported. Watch for students bending the analysis to fit.
- 7. Document Progress. Treat the log as a graded deliverable, not an afterthought. Have them record issues fixed, algorithms used and insights gained, with commented code snippets. This is their evidence trail for the externally assessed project.

WHAT TO WATCH FOR

- Students assume median is just the middle element and forget the even-length case, so the function returns a single value instead of the average of the two middle values. Have them test on a four-item list where they know the correct answer by hand, then on a five-item list, before running on real data.
- Students think cleaner data means better results, and delete rows or fill gaps until the numbers match their hypothesis. Require a written reason for every removed or filled value. Stress that an honest result that refutes the hypothesis earns marks, a manipulated one does not.
- Students treat calculating a statistic as the analysis itself and print a mean with no interpretation. For every number produced, ask them to write one sentence linking it back to the hypothesis before moving on.

DIFFERENTIATION

- Emerging: If Python or the dataset file is fighting a student, pair them and get one clean statistic printed first, then build from there. Give them the tested sample dataset so they can get functions working before facing their own messier data.
- Developing: Have them explain aloud what their cleaning choice does to the data before they run it, so the reasoning is separate from the code.
- Proficient: Push toward Higher Level treatment: have them consider correlation between two variables, or the effect their cleaning choice had on the final mean. Have them write the evaluation section as project evidence, stating clearly whether results support the hypothesis and why.