DatalytIQs Academy

Cambridge O Level Computer Science 2210

Printable teaching, practice and revision companion

Independent learning resource aligned to the current Cambridge syllabus. This material is original DatalytIQs Academy content and is not endorsed by Cambridge International Education.

Contents and use

1. Data representation
2. Data transmission
3. Hardware
4. Software
5. Internet and networks
6. Automated and emerging technologies
7. Algorithm design
8. Python programming
9. Databases
10. Boolean logic

How to study

Study the concept, reproduce the worked method, complete the practical task without copying, answer the examination-style questions, mark with the guidance and record corrections.

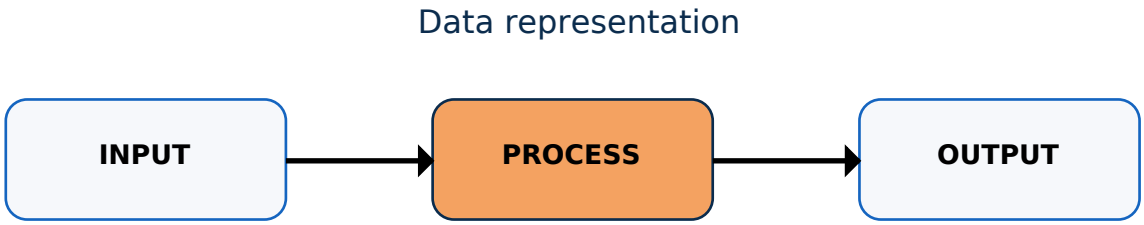
1. Data representation

Learning outcomes

Explain the principles of Data representation; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Data representation should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.



Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Data representation. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Data representation, complete the worked method with new data, identify a limitation and justify an improvement.

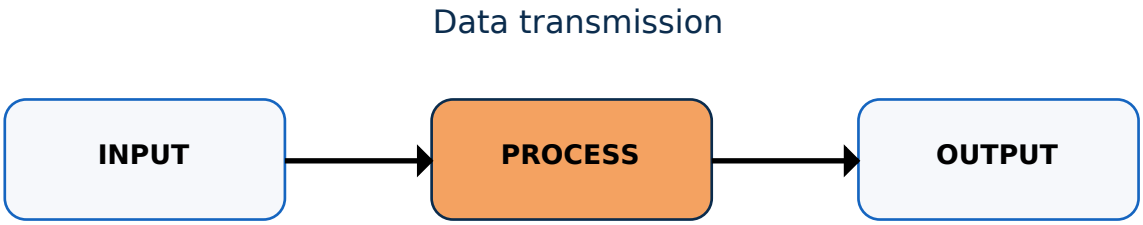
2. Data transmission

Learning outcomes

Explain the principles of Data transmission; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Data transmission should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.



Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Data transmission. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Data transmission, complete the worked method with new data, identify a limitation and justify an improvement.

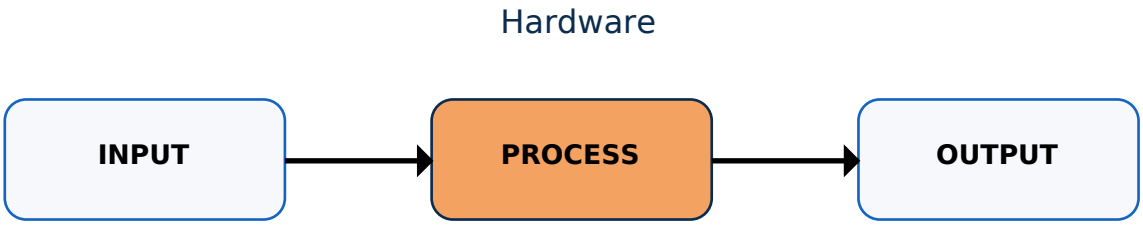
3. Hardware

Learning outcomes

Explain the principles of Hardware; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Hardware should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.



Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Hardware. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Hardware, complete the worked method with new data, identify a limitation and justify an improvement.

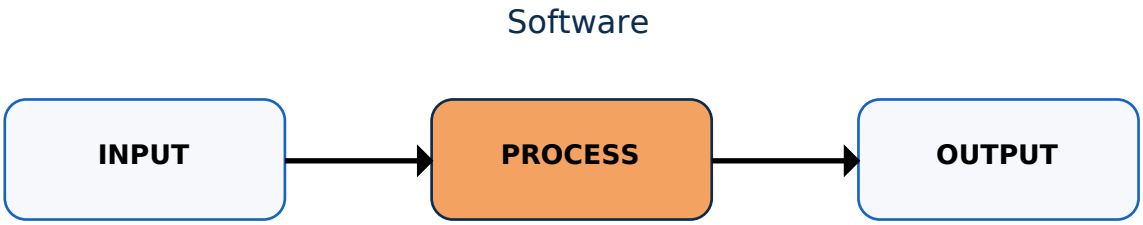
4. Software

Learning outcomes

Explain the principles of Software; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Software should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.



Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Software. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Software, complete the worked method with new data, identify a limitation and justify an improvement.

5. Internet and networks

Learning outcomes

Explain the principles of Internet and networks; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Internet and networks should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.

Internet and networks

Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Internet and networks. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Internet and networks, complete the worked method with new data, identify a limitation and justify an improvement.

6. Automated and emerging technologies

Learning outcomes

Explain the principles of Automated and emerging technologies; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Automated and emerging technologies should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.

Automated and emerging technologies

Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Automated and emerging technologies. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Automated and emerging technologies, complete the worked method with new data, identify a limitation and justify an improvement.

7. Algorithm design

Learning outcomes

Explain the principles of Algorithm design; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Algorithm design should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.

Validate inputs - apply rules - verify outputs

Worked program

```
def analyse(scores):
    valid = [s for s in scores if 0 <= s <= 100]
    if not valid:
        return {"count": 0, "mean": None}
    return {"count": len(valid), "mean": sum(valid)/len(valid)}

result = analyse([56, 73, 81, -2, 64])
print(result)
```

The function validates values before calculating. Four observations remain; their total is 274 and the mean is 68.5. Boundary tests should include 0 and 100; erroneous tests should include text and missing values.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Algorithm design. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Algorithm design, complete the worked method with new data, identify a limitation and justify an improvement.

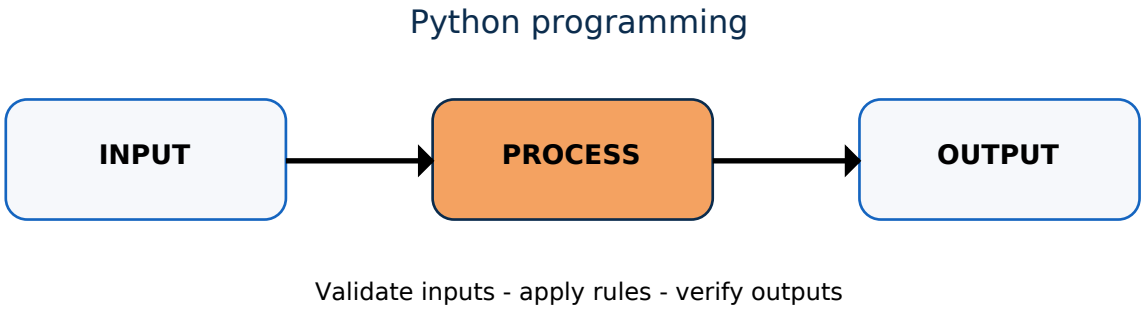
8. Python programming

Learning outcomes

Explain the principles of Python programming; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Python programming should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.



Worked program

```
def analyse(scores):
    valid = [s for s in scores if 0 <= s <= 100]
    if not valid:
        return {"count": 0, "mean": None}
    return {"count": len(valid), "mean": sum(valid)/len(valid)}

result = analyse([56, 73, 81, -2, 64])
print(result)
```

The function validates values before calculating. Four observations remain; their total is 274 and the mean is 68.5. Boundary tests should include 0 and 100; erroneous tests should include text and missing values.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Python programming. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Python programming, complete the worked method with new data, identify a limitation and justify an improvement.

9. Databases

Learning outcomes

Explain the principles of Databases; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Databases should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.

Validate inputs - apply rules - verify outputs

Worked SQL example

```
SELECT county, COUNT(*) AS learners,
       ROUND(AVG(score), 1) AS mean_score
FROM results
WHERE score BETWEEN 0 AND 100
GROUP BY county
ORDER BY mean_score DESC;
```

The filter protects the analysis from impossible marks. Grouping produces one record per county; aggregation calculates participation and performance. Primary keys prevent duplicate records and foreign keys preserve relationships.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Databases. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Databases, complete the worked method with new data, identify a limitation and justify an improvement.

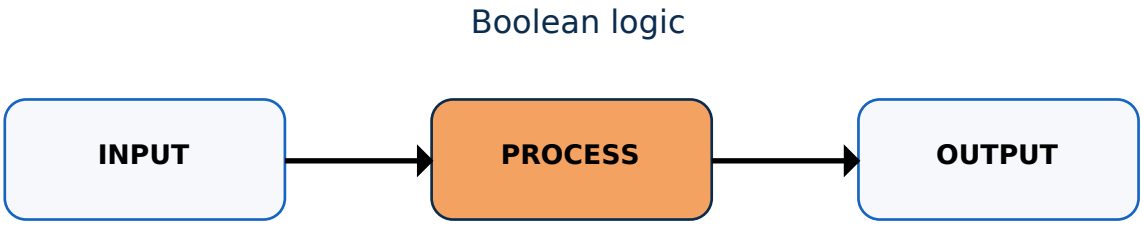
10. Boolean logic

Learning outcomes

Explain the principles of Boolean logic; apply them to an unfamiliar context; select and justify an appropriate method; test or verify the result; and evaluate limitations.

Concept development

Boolean logic should be understood as part of a complete information system rather than as an isolated definition. Identify purpose, users, inputs, processing rules, outputs and controls. Explain causal links and compare alternatives using suitability, accuracy, efficiency, reliability, security, accessibility, cost and ethical implications.



Validate inputs - apply rules - verify outputs

Worked application

```
Requirement -> suitable input -> controlled processing -> verified output -> evaluation
```

For a county service register, specify mandatory identifiers, validate dates and categories, restrict access by role, log changes, back up records, and test recovery. Evaluate usability, accuracy, security, cost and inclusion.

Practical activity

Use the supplied practical pack. Define the requirement, prepare or import the data, implement the solution, capture evidence of tests, interpret the output and evaluate one improvement. Use normal, boundary and erroneous test data where applicable.

Examination practice

1. Define two essential terms in Boolean logic. 2. Explain one mechanism using a linked cause-and-effect chain. 3. Apply the concept to a Kenyan school, health facility or enterprise. 4. Construct or annotate an appropriate diagram, algorithm, model or workflow. 5. Evaluate a proposed solution and reach a justified conclusion.

DatalytIQs marking guidance

Award credit for accurate knowledge, explicit explanation, relevant application and supported evaluation. A high-level response uses the scenario throughout, shows working, distinguishes facts from assumptions and qualifies its conclusion. Brand names alone do not demonstrate understanding.

Common errors

Listing points without explanation; confusing validation with verification; ignoring units or data types; omitting boundary tests; describing features without linking them to the requirement; and giving an absolute conclusion where suitability depends on context.

Revision checkpoint

Without notes, explain Boolean logic, complete the worked method with new data, identify a limitation and justify an improvement.