DatalytIQs Academy

Cambridge International AS and A Level Computer Science 9618

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. Information representation
2. Communication
3. Hardware
4. Processor fundamentals
5. System software
6. Security and integrity
7. Ethics and ownership
8. Databases
9. Algorithm design
10. Data types and structures
11. Programming
12. Software development
13. Advanced data representation
14. Advanced networking
15. Virtual machines
16. Advanced system software
17. Cryptography
18. Artificial intelligence
19. Computational thinking
20. Object-oriented programming
21. Further data structures
22. Algorithms
23. Recursion
24. Programming paradigms

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.

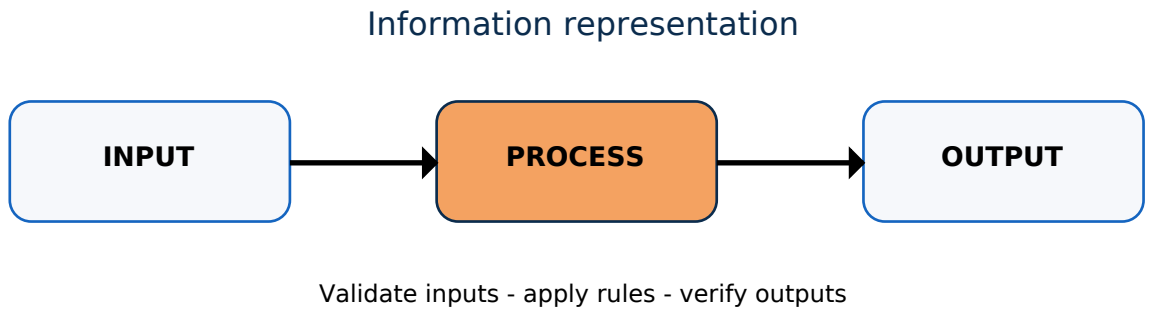
1. Information representation

Learning outcomes

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

Concept development

Information 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.



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 Information 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 Information representation, complete the worked method with new data, identify a limitation and justify an improvement.

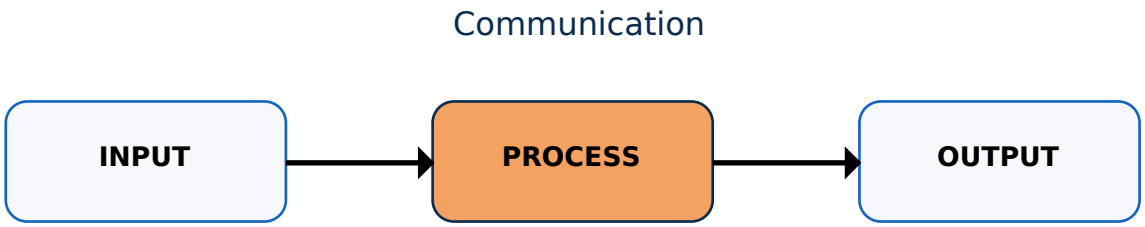
2. Communication

Learning outcomes

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

Concept development

Communication 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 Communication. 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 Communication, complete the worked method with new data, identify a limitation and justify an improvement.

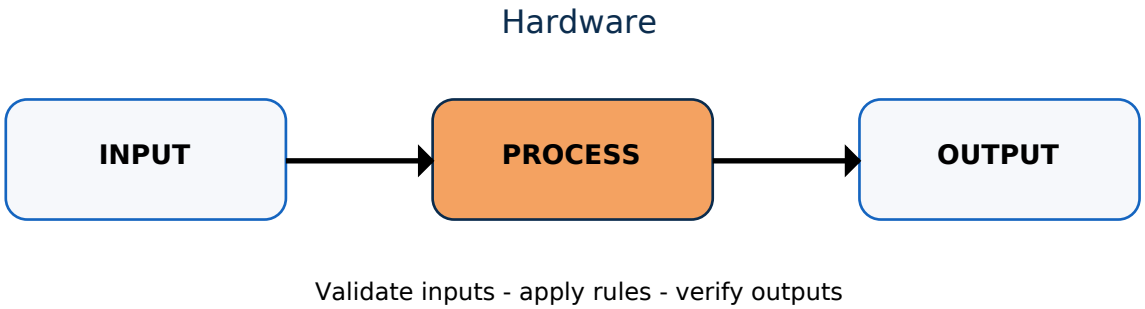
3. Hardware

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.



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.

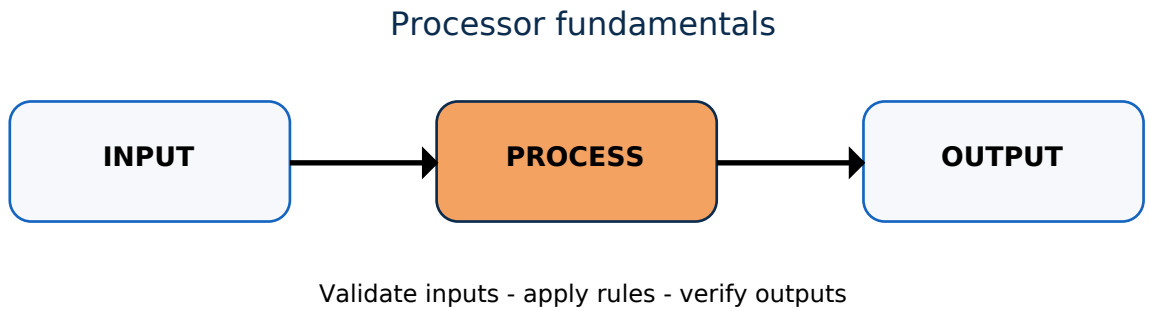
4. Processor fundamentals

Learning outcomes

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

Concept development

Processor fundamentals 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 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 Processor fundamentals. 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 Processor fundamentals, complete the worked method with new data, identify a limitation and justify an improvement.

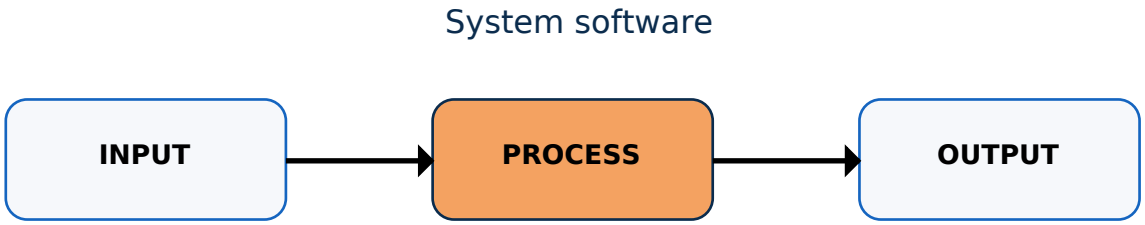
5. System software

Learning outcomes

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

Concept development

System 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 System 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 System software, complete the worked method with new data, identify a limitation and justify an improvement.

6. Security and integrity

Learning outcomes

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

Concept development

Security and integrity 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 Security and integrity. 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 Security and integrity, complete the worked method with new data, identify a limitation and justify an improvement.

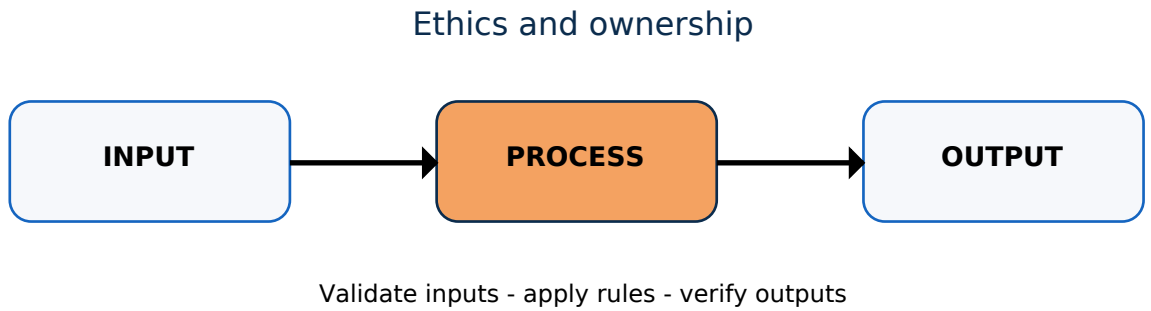
7. Ethics and ownership

Learning outcomes

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

Concept development

Ethics and ownership 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 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 Ethics and ownership. 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 Ethics and ownership, complete the worked method with new data, identify a limitation and justify an improvement.

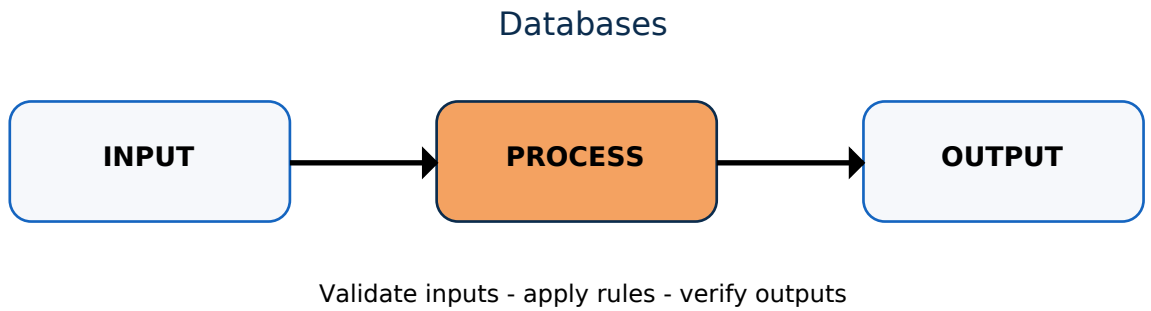
8. Databases

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.

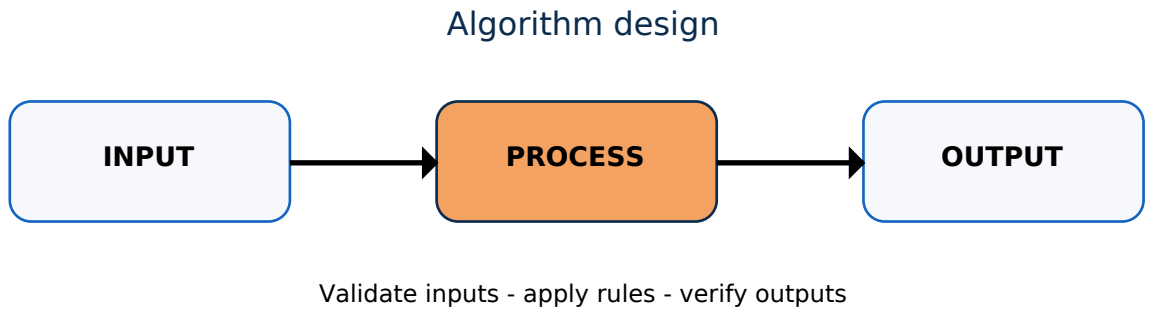
9. Algorithm design

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.

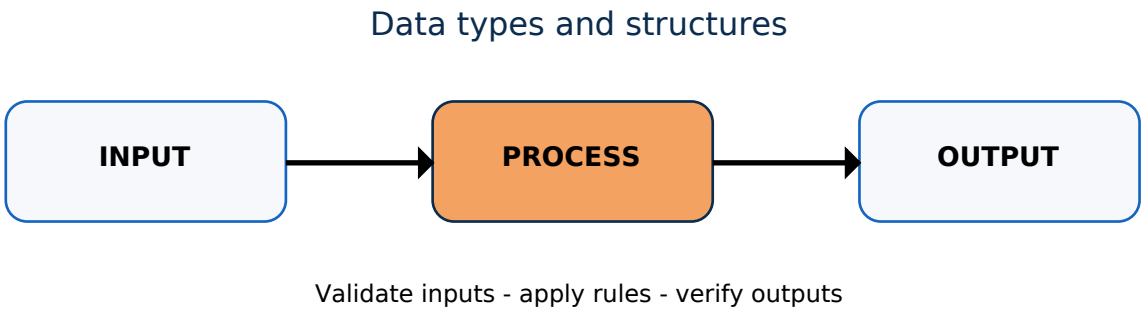
10. Data types and structures

Learning outcomes

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

Concept development

Data types and structures 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 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 types and structures. 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 types and structures, complete the worked method with new data, identify a limitation and justify an improvement.

11. Programming

Learning outcomes

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

Concept development

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.



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 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 Programming, complete the worked method with new data, identify a limitation and justify an improvement.

12. Software development

Learning outcomes

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

Concept development

Software development 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 development. 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 development, complete the worked method with new data, identify a limitation and justify an improvement.

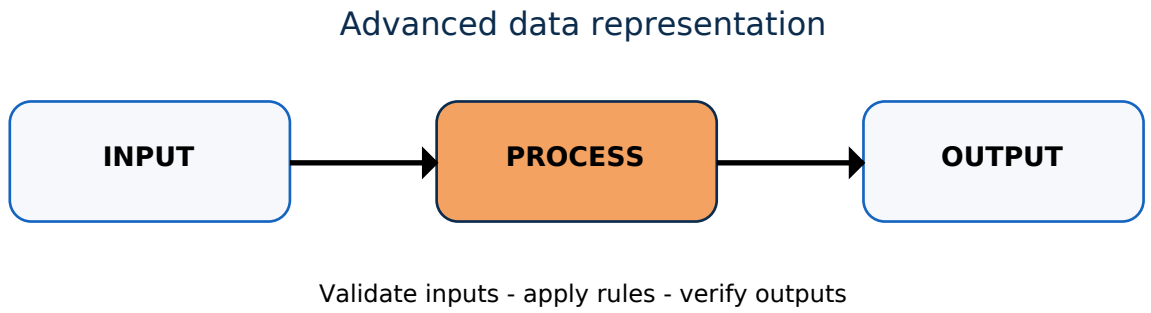
13. Advanced data representation

Learning outcomes

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

Concept development

Advanced 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.



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 Advanced 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 Advanced data representation, complete the worked method with new data, identify a limitation and justify an improvement.

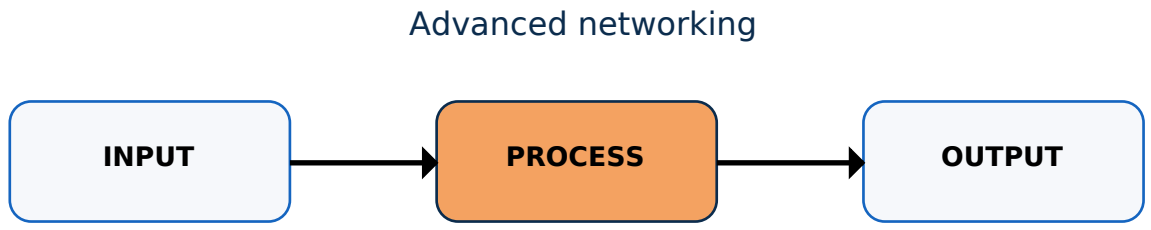
14. Advanced networking

Learning outcomes

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

Concept development

Advanced networking 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 Advanced networking. 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 Advanced networking, complete the worked method with new data, identify a limitation and justify an improvement.

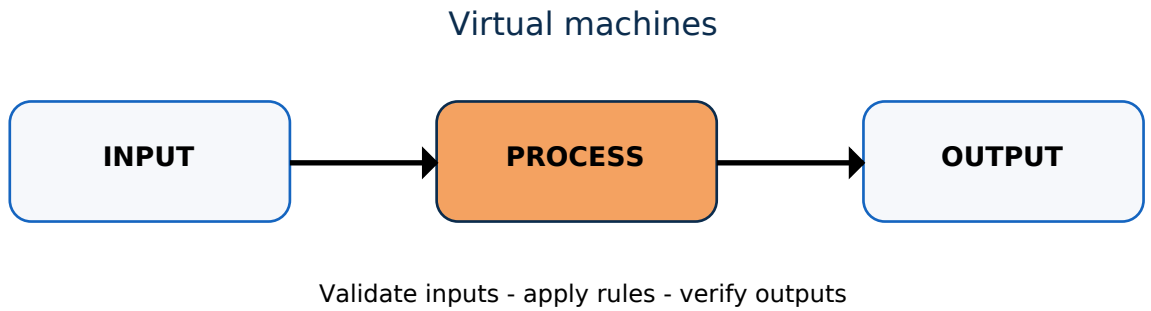
15. Virtual machines

Learning outcomes

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

Concept development

Virtual machines 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 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 Virtual machines. 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 Virtual machines, complete the worked method with new data, identify a limitation and justify an improvement.

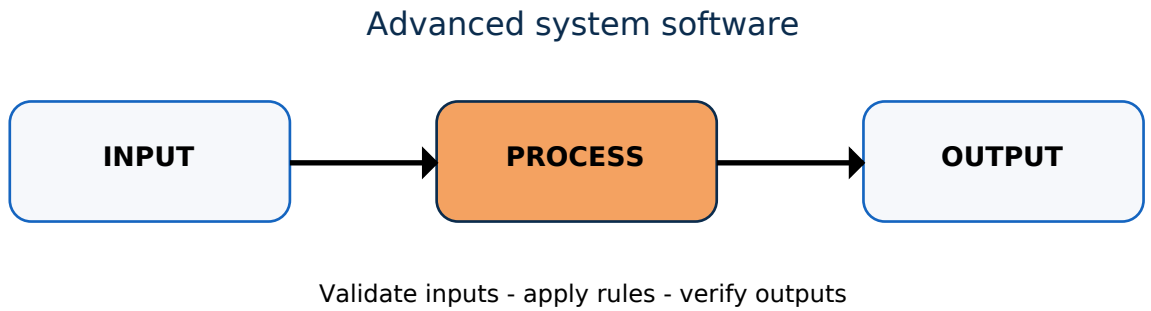
16. Advanced system software

Learning outcomes

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

Concept development

Advanced system 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 Advanced system 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 Advanced system software, complete the worked method with new data, identify a limitation and justify an improvement.

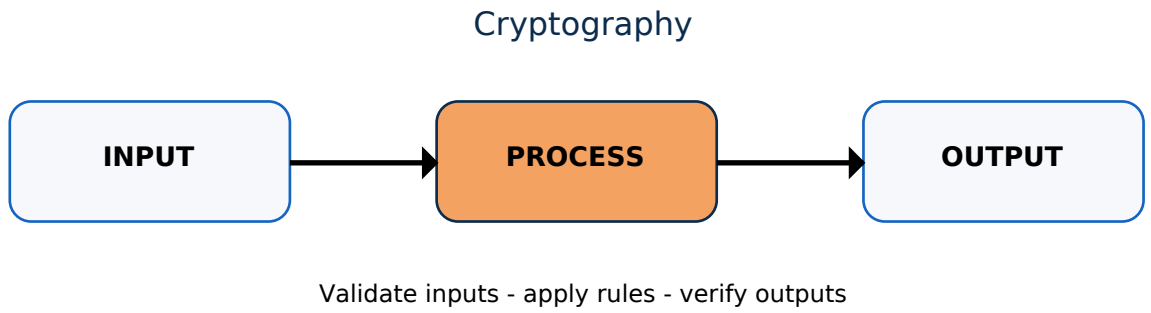
17. Cryptography

Learning outcomes

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

Concept development

Cryptography 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 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 Cryptography. 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 Cryptography, complete the worked method with new data, identify a limitation and justify an improvement.

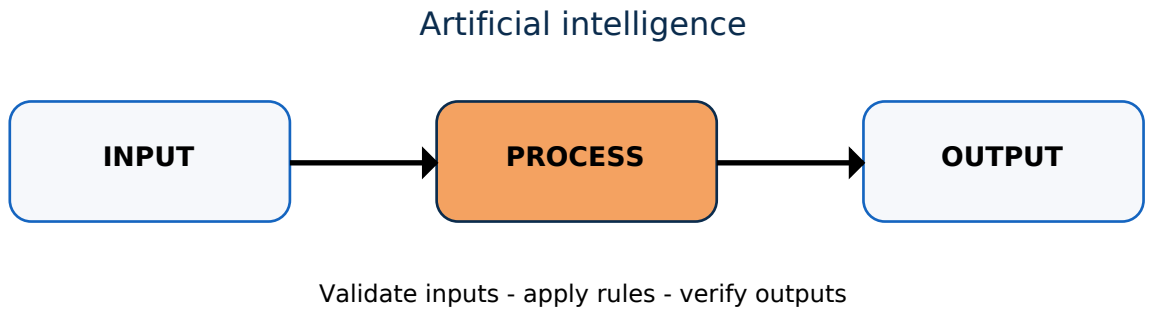
18. Artificial intelligence

Learning outcomes

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

Concept development

Artificial intelligence 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 Artificial intelligence. 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 Artificial intelligence, complete the worked method with new data, identify a limitation and justify an improvement.

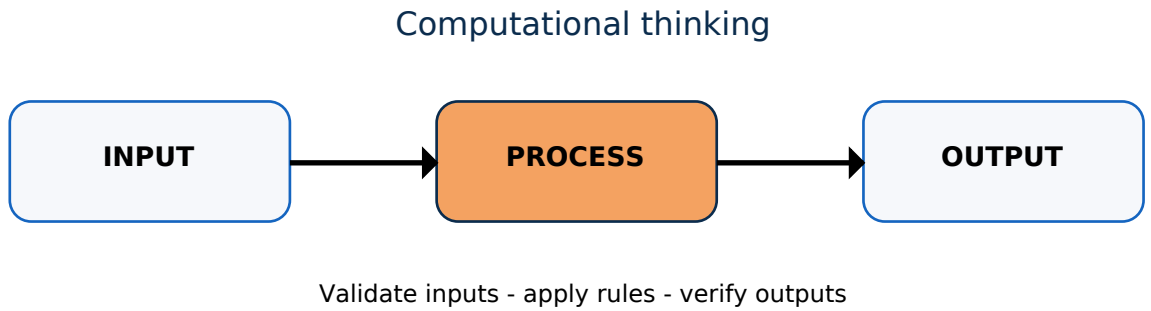
19. Computational thinking

Learning outcomes

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

Concept development

Computational thinking 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 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 Computational thinking. 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 Computational thinking, complete the worked method with new data, identify a limitation and justify an improvement.

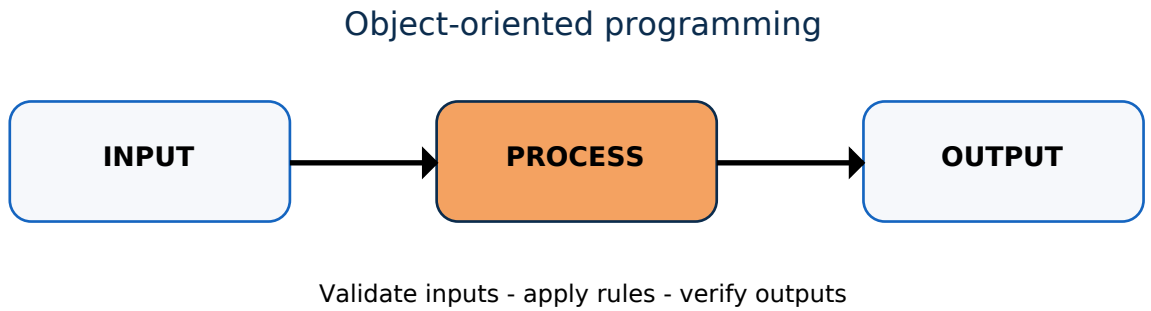
20. Object-oriented programming

Learning outcomes

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

Concept development

Object-oriented 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.



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 Object-oriented 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 Object-oriented programming, complete the worked method with new data, identify a limitation and justify an improvement.

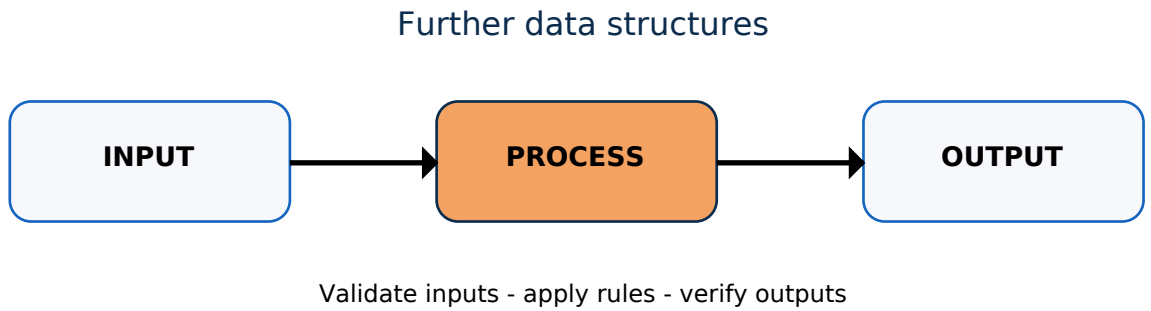
21. Further data structures

Learning outcomes

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

Concept development

Further data structures 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 Further data structures. 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 Further data structures, complete the worked method with new data, identify a limitation and justify an improvement.

22. Algorithms

Learning outcomes

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

Concept development

Algorithms 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 Algorithms. 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 Algorithms, complete the worked method with new data, identify a limitation and justify an improvement.

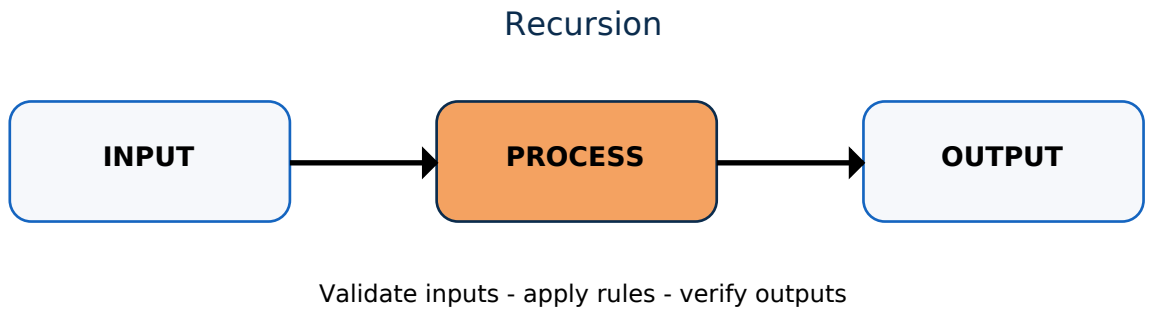
23. Recursion

Learning outcomes

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

Concept development

Recursion 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 Recursion. 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 Recursion, complete the worked method with new data, identify a limitation and justify an improvement.

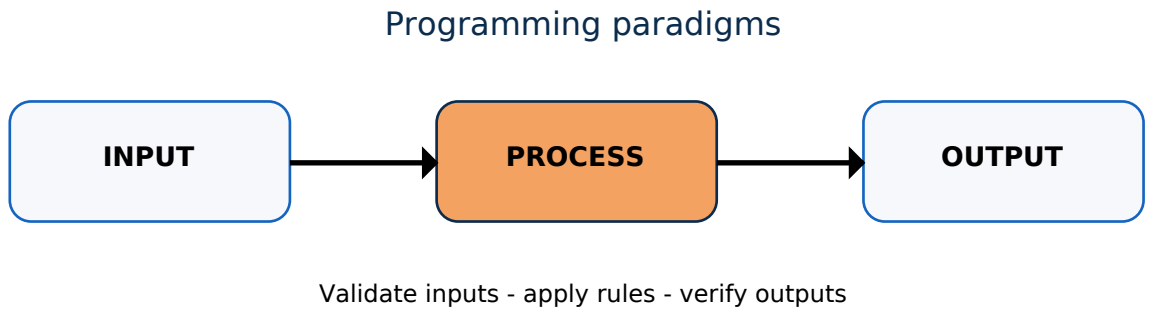
24. Programming paradigms

Learning outcomes

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

Concept development

Programming paradigms 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 Programming paradigms. 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 Programming paradigms, complete the worked method with new data, identify a limitation and justify an improvement.