DELTA

Computer Science Year 11
Half Term 1

Computing Strands:
Computer Science

welcome

YEAR 11

WHERE HAVE YOU BEEN?

Intro to Comp Sci

Computational thinking – Abstraction / Decomposition & algorithm thinking

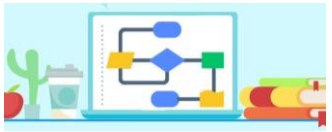
Writing algorithms – create, interpret, correct, complete and refine algorithms

Understanding algorithms - create, interpret, correct, complete and refine algorithms

2.1 Algorithms
2.2 Programming 2.3 Robust Programming

ecification requirements. Overview of course expectations.

Programming skills – Variables/Constants



Programming skills – Data types

Trace tables – create and use of trace tables to follow algorithms

Algorithm Search – Understanding steps of binary/linear search

Interpreting flowcharts – Produce diagrams to show problem solving

Interpreting flowcharts – Produce diagrams to show problem solving

2.1 Algorithms
2.2 Programming 2.3 Robust Programming

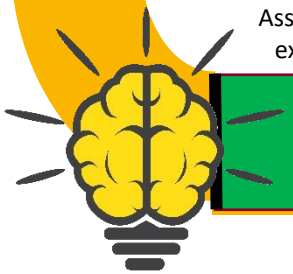
Programming skills –IF / ELSE / ELIF statements (Selection)

Programming skills –IF / ELSE statements (Selection)

SKILLS TAUGHT:

- Computational thinking
- Thinking abstractly
- Concepts of decomposition
- Sequence / Selection and Iteration in programming
- Debugging programs
- Designing programs
- Writing programs

Assessment preparation – exam retrieval practice



Assessment

NEXT Y11

HALF TERM 2

Iterative Assessment COMP 1 // 2.1 // 2.2

Why are we learning this? To be able to understand and apply fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic and algorithms.

DELTA

Computer Science Year 11
Half Term 2

Computing Strands:
Computer Science

welcome

YEAR 11

WHERE HAVE YOU BEEN?

Assessment Recap

Assessment feedback. Responding to assessment

2.1 Algorithms

Algorithm Sort – Understanding steps of bubble/merge/insertion sort

2.2 Programming

Programming skills – For loops (Iteration)

Programming skills – While loop (Iteration)

2.3 Robust Programs

Defensive design – Understanding issues a programmer should consider to ensure program caters for all input values

Input validation – Understanding how to deal with invalid data in a program

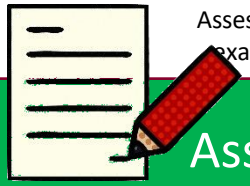
SKILLS TAUGHT:

- Data validation
- Defensive programs
- Program maintenance
- Sequence / Selection and Iteration in programming
- Debugging programs
- Designing programs
- Writing programs

Assessment preparation – exam retrieval practice

Assessment preparation – exam retrieval practice

Assessment preparation – exam retrieval practice



Assessment

Component One Assessment
Component Two Assessment
(Mocks)

Assessment preparation – exam retrieval practice

Maintainability – Purpose of naming conventions, indentations and commenting and how they apply to programs

2.3 Robust Programs

2.2 Programming

Programming skills – 1D Arrays (Data Structure)

NEXT Y11

HALF TERM 3

Why are we learning this? To be able to analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.

Computer Science Year 11
Half Term 3

Computing Strands:
Computer Science

welcome

YEAR 11

WHERE HAVE YOU BEEN?

Testing – Purpose of testing including final/iterative testing.
What are the differences between logic and syntax errors?
How do they apply to programs (correcting errors)

Suitable test data – Selecting and using suitable test data normal/boundary and erroneous

2.3 Robust Programs

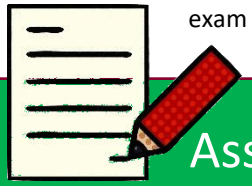
2.2 Programming

Programming skills – 2D Arrays (Data Structure)

Programming skills – Functions / Procedures

Assessment preparation – exam retrieval practice

Assessment feedback. Responding to assessment



Assessment preparation – exam retrieval practice

Assessment

Iterative Assessment
COMP 1 // 2.1 // 2.2 // 2.3

Assessment preparation – exam retrieval practice

SKILLS TAUGHT:

- Sequence / Selection and Iteration in programming
- Debugging programs
- Designing programs
- Writing programs
- Thinking logically
- Sub routine programs

Logic gates – producing logic diagrams including truth tables for AND/OR/NOT gates

Logic gates – producing combined logic gates inc expressions

2.4 Boolean Logic

2.2 Programming

Programming skills – Functions / Procedures

NEXT Y11

HALF TERM 3

Why are we learning this? To be able to analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.

Computer Science Year 11
Half Term 4

Computing Strands:
Computer Science

welcome

Languages – Characteristics and purpose of high level / low level languages. Purpose of translators including differences between compiler and interpreter.

YEAR 11

WHERE HAVE YOU BEEN?

2.5 Programming Languages and IDE

IDE – Tools and facilities available in integrated development environment.

HT4 to focus on revision on Mock preparations using exam retrieval practice

Assessment preparation – exam retrieval practice

Assessment preparation – exam retrieval practice

Assessment preparation – exam retrieval practice

Assessment

Component One Assessment
Component Two Assessment (Mocks)

1.1 Revision

Assessment preparation – exam retrieval practice

1.2 Revision

COMPONENT ONE REVISION

2.2 Programming

Programming skills – file handling

Programming skills – file handling

NEXT Y11

HALF TERM 4

SKILLS TAUGHT:

- Types of languages
- Differences between languages
- IDE tools and how they can be used
- Debugging programs
- Designing programs
- Writing programs

Why are we learning this? To be able to understand various types of programming languages to develop understanding of how programs are executed and translated.

Computer Science Year 11
Half Term 5

Computing Strands:
Computer Science

welcome

YEAR 11

WHERE HAVE YOU BEEN?

1.3 Revision

COMP 1

Assessment practice – walking talking exam questions

2.2 Programming

Programming skills – SQL statements

1.4 Revision

COMP 1

1.5 Revision

1.6 Revision

HT5 to focus on revision of component one/two and recapping programming skills based on QLA

Assessment practice – walking talking exam questions

Exam Retrieval

2.2 Programming

Programming Fundamentals – programming skills

Exam Retrieval

Assessment practice – walking talking exam questions

2.2 Programming

Programming Fundamentals – programming skills

NEXT Y11

GCSE EXAM!

SKILLS TAUGHT:

- Computational thinking
- Thinking abstractly
- Concepts of decomposition
- Sequence / Selection and Iteration in programming
- Debugging programs
- Designing programs
- Writing programs

Why are we learning this? To be able to understand and apply fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic and algorithms. To be able to analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.