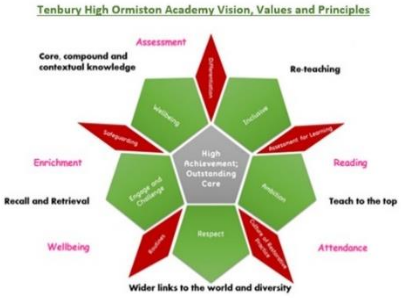

Curriculum Overview for Computer Science
Year 11

Half Term 4
Logic, Defensive Design and Testing

Substantive Knowledge:

- Understand why logic is important in computer programming
- Understand what AND means
- Understand what OR means
- Understand what NOT means
- Understand what a Truth Table is used for
- Understand how to combine logic gates into a logic diagram
- Understand the purpose of testing
- Understand the purpose of iterative testing
- Understand the purpose of final/terminal testing
- Understand the types of errors that happen in programming
- Understand the concept of test data
- Understand the need for boundary test data
- Understand the need for Erroneous test data
- Understand the concept of defensive design
- Understand why a program may be misused
- Understand the need for authentication
- Understand the need for code maintainability
- Understand the use of subprograms
- Understand the difference between a subroutine and a function
- Understand the need for naming conventions
- Understand the need for indentation
- Understand the need for comments

Disciplinary Knowledge:

Model reading
Reading out loud
Skim and Scan of source information
Decoding terms
Etymology of key terms

Logic
AND gate
OR gate
NOT gate
Truth Table
Logic diagram
Testing
Iterative Testing
Terminal Testing
Errors
Syntax Errors
Runtime Errors
Logic Errors
Normal Data
Boundary data
Erroneous data
Defensive Design
Misuse
Authentication
Maintainability
Subprograms
Subroutine
Function
Arguments
Naming Convention
Camel Case
Snake Case
Indentation
Comments

Formative assessment
Knowledge checks
Smart Revise
Practice questions
Summative assessment
End of unit assessment

Tenbury High Ormiston Academy Vision, Values and Principles

• Be able to identify the gate symbols for AND, OR and NOT gates • Be able to complete a truth table for each gate • Be able to produce a logic diagram • Be able to complete a truth table for a logic diagram • Be able to identify syntax errors, runtime errors and logic errors • Be able to identify Normal, Boundary and Erroneous test data • Be able to design a program anticipating misuse • Be able to design a program to include authentication • Be able to write a program that includes a subroutine • Be able to run a subroutine • Be able to write a program that includes a function • Be able to run a function • Be able to include arguments into subprograms • Be able to comment code in Python	🏠	Practice questions Revision tasks Research tasks