Form 4 Computing Syllabus

- **Negative representations in Binary Numbers**

  2’s complement representation.

  Binary addition and subtraction (by complementation followed by addition) in 2’s complement.

- **Boolean Algebra**

  Evaluating a Boolean expression (propositional logic formula) – such as \((\bar{A}+B).C\) given the values of the Boolean variables, by converting to a logic circuit, *i.e. drawing the circuit from the Boolean expression or from the truth table.*

  *Construction of a truth table/logic circuit from a textual description of the problem.*

- **Software Applications - off the shelf, tailor made**

  Off-the-shelf, customisable and tailor-made packages.

  Advantages and disadvantages of each.

  *System and application software*

    *Installation, on-line help.*
• Inputting data- validation/verification

• Data entry errors – omission, substitution, transposition

Data capture forms (very simple).

Preparation and transcription of data with

their related errors *(transposition, omission, substitution)*.

Solution through data verification and validation; check digits; range check.

• Java

1. Valid meaningful identifiers, enforcing code convention rules, case sensitivity.

2. Variable and constant declarations

3. Data types *(primitive)*, compatibility and type casting

   byte, short, int, float, char, boolean.

4. The type String can be used for alphanumeric data. No knowledge of String class methods is required.

5. Initialisation and assignment operator –

   equal sign (=)

6. Comments
7. Arithmetic operators, precedence and expressions

   +, -, *, /, %, ++, --, +=, -=, /=, %=  

8. Printing and formatting of text-based output, escape characters:

   print(), println(), \, \n, \t (tutors may use printf() method for formatting purposes especially decimal numbers)

9. Inputting data from the keyboard through a third party class (the keyboard class).

10. Logical operators, simple and compound logical expressions

    !(unary not), && (and), || (or), == (equal to), != (not equal to), > (greater than), >= (greater or equal), < (smaller than), <= (smaller or equal).

11. Loops (including nested loops):

    while, do while and for

12. Conditional transfer (including at least 2 level nesting):

    if, if – else, switch

13. Math class – the Math class should be explained as one of the libraries/classes found in the JDK, and limited to the following methods:

    abs(), pow(), sqrt(), random(), round(), ceil(), floor().
14. Declaration and creation of single dimensional arrays of primitive data types.

15. Objects – Students create their own classes containing simple void methods and create instances through the use of the keyword ‘new’.

Syntax, logic and runtime errors.

Declaring and calling methods with and without arguments and simple methods which return a value.

- CPU – design

The processor and main memory.

The main components of the processor – control unit, ALU.

Registers: Accumulator, Program counter (PC), Instruction register (IR).

Main memory and memory addresses. The address bus and how its width relates to the size of the address space.

The data bus. The computer’s word size (or word length).

Control bus (read/write line only).

Concept of a stored program as a set of instructions.

Brief account of fetch and execute cycle.

Processor speed – cycles per second expressed in Hz, KHz, MHz and GHz.
Units of time measurements: milli, micro, nano seconds.

- Data Security and Privacy

Need of data security and integrity of data.

Backups (the generations of files: grandfather, father, son files).

*Parity checking.*

Physical security and software safeguards.


Software piracy and copyright.

Ethical and legal issues.

Hardware and software procedures which deter piracy - serial numbers and activation keys, hardware keys (dongles).

Software registration.

Access rights.

Privacy on multi-user/network systems.