LP	Computer Science
8-9	Students will partially decompose a problem into its sub-problems **e.g. to make use of structured flowcharts to represent it.** Students will analyse and present an algorithm for a given task **e.g. Small Basic, Scratch, Python.** Students will explore the effects of changing the variables in a model or program. Students will try out and refine sequences of instructions Students will be able to manipulate strings and select appropriate data types **e.g. defines data types: real numbers and Boolean.** Students will design and use simple data structures **e.g. select correct data type for a program.** Students will recognise similarities between simple problems and algorithms **e.g. write algorithm for given program** Students will reflect critically on programs in order to make improvements in further programming exercises Students will know that computers use binary to represent all data **e.g. 0,1**
6-7	Students will understand how bit patterns represent numbers and images. Students will understand & explain that iteration is the repetition of a process such as a loop e.g. smoke alarm **e.g. algorithm/flowchart.** Students will have practical experience of a high-level textual language **e.g. python programming.** Students will use a range of operators and expressions e.g. Boolean, and apply them in the context of program control **e.g. python programming.** Students will recognise and understand the function of the main internal parts of basic computer **e.g. dismantling a PC.** Students will understand the concepts behind the fetch-execute-cycle **e.g. routine of instructions.** Students will be able to suggest ways in which search engines rank search results **e.g. use of keywords, site structure and quality of links.**
4-5	Students will analyse and represent symbolically a sequence of events. e.g. create an algorithm using symbols **e.g. data flow diagram** Students will recognise different types of data **e.g. text, number, real/integer, instructions and string.** Students will understand the need for care and accuracy when programming **e.g. errors, debugging.** Students will give instructions involving selection and repetition **e.g. loop, if, else.** Students will 'think through' an algorithm and predict an output showing an awareness of inputs. Students will present data in a structured format suitable for processing. Students will recognise tasks completed by humans or computers **e.g. data entry (register) can be automated or by humans, CAD/CAM – car production line.** Students will design solutions by breaking down a problem and create a sub-solution for each of these parts. Students will design, write and debug structured (modular) programs using steps (procedures) **e.g. lots of mini procedures which make up a program.** Students will understand why and when computers are used **e.g. work (production), social (communicating), efficiency & effectiveness of tasks.** Students will understand how to effectively use search engines **e.g. knowing how to use advanced search to narrow down specific data.**

LP	Computer Science
2-3	Students will recognise similarities between storyboards of everyday activities **e.g. brushing teeth, cooking, walking to school**. Students will present data in an orderly way e.g. storyboard, set of instructions **e.g. method for recipe, making a sandwich or smoothie.** Students will design a simple linear (non-branching) sequence of instructions to make things happen **e.g. Scratch, Small Basic, IF statement.** Students will design simple algorithm flowchart to show solutions using repetition and two-way selection **e.g. a flowchart with IF, then and else.** Students will use algorithms to develop, improve instructions and create programs to achieve given goals. Students will recognise, state and assign variables. Students will use post-tested loop e.g. 'until', and a sequence of selection statements in programs, including and if, then and else statement **e.g. Scratch – character to dance to music until it ends.** Students will know that computers collect data from various input devices **e.g. sensors – doors/lights and application software.** Students will understand the difference between hardware and application software and their roles within a computer system **e.g. labelling peripherals & components of computer system and what software to use for a particular purpose/task.**
0-1	Students will draw a storyboard or other diagram of processes of activities **e.g. algorithm showing wearing socks before shoes**. Students will organise items in simple sets of data **e.g. text, numeric, Boolean, binary.** Students will plan and give direct commands to make things happen **e.g. Big Track, Scratch.** Students will solve simple problems using software **e.g. Big Track, Scratch Formulas and Queries.** Students will understand that algorithms are a set of instructions that are carried out on digital devices as programs **e.g. smoke alarm.** Students will design simple algorithms using loops and choices i.e. if statements. **e.g. brushing teeth.** Students will be able to look at an algorithm, identify errors and correct them **e.g. debugging, in algorithms.** Students will identify and correct simple **semantic errors (words, phrases, signs, and symbols)** Students will use arithmetic operators **(+ / * -)** if statements, and loops, within programs **e.g. Turtle/Small Basic creating a shape i.e. "square" & Big track.** Students will recognise and can use a range of input and output devices **e.g. digital camera, scanner, webcam, headphones, monitor, mouse, keyboard.**