

Year 12 SL Study Guide for Half-Yearly

System fundamentals (from topic 1): System life cycle (feasibility study, analysis, design, implementation, operation, maintenance), feasibility report (technical, economic, legal, operational and schedule feasibility). Data gathering (unstructured interviews, questionnaires, Changeover methods (immediate, parallel, pilot, phased). Change management, legacy systems, business merger, SaaS, changeover methods, data migration. Types of testing (functional testing, data testing, dry run testing). Validation and verification. Kinds of documentation (internal, external, user documentation), good and bad documentation. Different ways of providing user documentation and user training. Data loss: causes and solutions. DFD, structure chart. Prototyping. Social and ethical issues.

Computer organisation (from topic 2): software, hardware. Input, process, storage, output model, data and information. CPU, CU, ALU. Data bus, address bus and control bus. Kilo, mega, giga, tera. RAM, ROM. Caches L1, L2. Registers (PC, CIR, Accumulator, MAR, MDR). Operating systems: peripheral communication, memory management, resource monitoring and multitasking, networking, disk access and data management, security. Software applications: word processors, spreadsheets, DBMSs, web browsers, email, CAD, graphic processing software. GUI, CLI. ASCII and Unicode. Secondary storage devices. Virtual memory.

Searching and sorting: sequential (linear) search. Bubble sort.

Computer logic: The AND, OR, NOT, NAND, NOR and XOR operations. Venn diagram, truth table and logic gate. Precedence rules of arithmetic and Boolean algebra. Logic expressions and logic circuits. Laws of Boolean algebra.

Number representation: decimal, binary, hex (with fractions) and conversions. Two's complement. Signed and unsigned numbers.

Computational thinking (from topic 4): flowchart, pseudocode, Gantt chart,

Java: algorithm, flowchart, pseudocode. Variables and types. Scanner class. Condition, conditional statement, if..then..else. For..loop. While..loop. Arrays, enter data, view data, find highest, search, modify. Strings, charAt(). Object-oriented programming: classes, objects, encapsulation, inheritance, class diagram. Exception, concurrency, modularisation.

Abstract data structures (from topic 5): Iteration, recursion. Stack (push, pop, isEmpty). Queue (enqueue, dequeue, isEmpty). One- and two-dimensional arrays (operations e.g. find maximum, add elements, search etc.).

Object-oriented programming (from topic Option D): class, object, encapsulation, inheritance, abstraction, polymorphism. UML, class diagram, association, inheritance, abstract class, decomposing a problem, dependency, aggregation.