

INFORMATION TECHNOLOGY CHEAT SHEET

 MASTER INFORMATION TECHNOLOGY — SYSTEMS & NETWORKING (SIMPLIFIED)

IT theory covers hardware, software, networks and security — let's break it all down 🙌

◆ 1. Systems Technologies

📌 Hardware: physical components of a computer ✓ CPU, RAM, ROM, storage (HDD, SSD), input/output devices, motherboard

📌 Software: ✓ System software: operating systems (Windows, Linux, macOS) ✓ Application software: programs used for specific tasks

📌 Factors affecting performance: ✓ CPU speed (GHz), RAM size, storage type (SSD faster than HDD), bus speed, cache memory

📌 OS tools and utilities: ✓ Task manager, disk defragmenter, antivirus, file compression, backup tools

💡 Quick Tip: ✓ More RAM = more tasks can run simultaneously without slowing down ✓ SSD vs HDD: SSD is faster and more durable but more expensive

◆ 2. Networking — LAN and WAN

📌 LAN (Local Area Network): computers connected within a limited area (school, office)

📌 WAN (Wide Area Network): connects LANs over large distances (the Internet)

📌 Network devices: ✓ Router: connects networks and directs data packets ✓ Switch: connects devices within a LAN ✓ Modem: converts digital signal to/from analogue for transmission ✓ Access point: provides wireless connectivity (Wi-Fi)

📌 Network architectures: ✓ Client-server: central server manages resources ✓ Peer-to-peer: all devices share resources equally

💡 Quick Tip: ✓ IP address = unique address for each device on a network ✓ MAC address = physical address built into the network card (hardware level)

◆ 3. Cloud Computing and the Internet

✚ Cloud computing: storage and software accessed via the Internet (not local hardware) ✓
Advantages: accessible anywhere, scalable, cost-effective ✓ Disadvantages: requires internet connection, privacy/security risks

✚ Internet connectivity options: ✓ ADSL, fibre, LTE/5G, satellite, dial-up

✚ Deep web vs dark web: ✓ Deep web: unindexed content (databases, private emails) — mostly legitimate ✓ Dark web: requires special software (Tor) — often associated with illegal activity

💡 Quick Tip: ✓ Cloud = not a physical place — it's servers owned by providers like Google, Amazon, Microsoft ✓ SEO (Search Engine Optimisation) = making websites rank higher in search results

◆ 4. Security, Risks and Ethics

✚ Computer security threats: ✓ Malware (viruses, ransomware, spyware), phishing, hacking, DDoS attacks

✚ Security solutions: ✓ Firewalls, antivirus, encryption, strong passwords, two-factor authentication

✚ Social and ethical issues: ✓ Digital divide: unequal access to technology ✓ Cyberbullying: harassment using digital platforms ✓ Green computing: reducing environmental impact of technology ✓ Big data: large datasets collected and analysed for patterns ✓ IoT (Internet of Things): everyday devices connected to the internet

💡 Quick Tip: ✓ Encryption = converting data into unreadable code without a key ✓ Two-factor authentication = adds a second verification step beyond a password

🚀 Want to master IT theory and boost your exam marks? 📞 Book a session with MindShift Learning today: +27 63 945 4868

📖 MASTER INFORMATION TECHNOLOGY — PROGRAMMING & DATABASES (CLEAR GUIDE)

The practical side of IT — OOP, algorithms, SQL and the PAT — made simple 🙌

◆ 1. Object-Oriented Programming (OOP)

✚ Key concepts: ✓ Class: blueprint for creating objects ✓ Object: instance of a class ✓ Field (attribute): data stored in an object ✓ Method: an action the object can perform ✓ Constructor: initialises an object when it's created ✓ Accessor (getter): returns the value of a field ✓ Mutator (setter): changes the value of a field

✦ OOP principles: ✓ Encapsulation: hiding internal data (private fields, public methods) ✓ Inheritance: a subclass inherits properties from a superclass ✓ Polymorphism: same method name behaves differently in different classes

💡 Quick Tip: ✓ Private fields + public getters/setters = encapsulation ✓ Inheritance = "is a" relationship (e.g. Dog is an Animal)

◆ 2. Data Structures and Algorithms

✦ Data types: integer, string, boolean, double, char ✦ Arrays: fixed-size collection of elements of the same type ✦ Dynamic arrays (ArrayLists): resizable arrays — elements can be added or removed

✦ Algorithms: ✓ Selection: if/else, switch statements ✓ Looping: for, while, do-while loops ✓ Array manipulation: searching, sorting, summing elements ✓ String manipulation: length, substring, indexOf, toUpperCase, replace

💡 Quick Tip: ✓ Arrays use index starting at 0 (first element = array[0]) ✓ For a loop with arrays: always check you're not going out of bounds

◆ 3. Databases and SQL

✦ Relational database: data stored in tables with rows and columns ✦ Normalisation reduces data redundancy: ✓ 1NF: no repeating groups, each cell has one value ✓ 2NF: remove partial dependencies ✓ 3NF: remove transitive dependencies

✦ Key SQL commands: ✓ SELECT: retrieve data ✓ WHERE: filter records ✓ ORDER BY: sort results ✓ GROUP BY + aggregate functions: COUNT, SUM, AVG, MAX, MIN ✓ JOIN: combine data from two tables ✓ Subquery: query within a query

💡 Quick Tip: ✓ Primary key = unique identifier for each record ✓ Foreign key = links one table to another ✓ SQL is case-insensitive but keywords written in CAPS by convention

◆ 4. Data Representation and Boolean Logic

✦ Number systems: ✓ Binary (base 2): 0s and 1s — used by computers internally ✓ Hexadecimal (base 16): compact representation of binary ✓ ASCII/Unicode: character encoding standards (letters, symbols, numbers)

✦ Boolean logic: ✓ AND: both conditions must be true ✓ OR: at least one condition must be true ✓ NOT: reverses the condition

✦ Truth tables: show all possible outcomes of Boolean expressions

💡 Quick Tip: ✓ Convert binary to decimal: each bit is a power of 2 (from right: 1, 2, 4, 8, 16...) ✓ Hexadecimal digits: 0–9 then A=10, B=11, C=12, D=13, E=14, F=15

♦ 5. The PAT — Practical Assessment Task

✦ Your PAT demonstrates your ability to solve a real-world problem using IT

✦ PAT components: ✓ Problem specification: what problem are you solving? ✓ User interface design: how will the user interact with the system? ✓ Class design: what classes, fields and methods are needed? ✓ Coding: clean, commented, well-structured code ✓ Data validation: check all inputs before processing ✓ Testing: test plans, functional testing, identify and fix bugs ✓ Documentation: explain critical algorithms and sourced code

💡 Quick Tip: ✓ Defensive programming = anticipate errors before they happen (validate inputs) ✓ Separate UI code from working/business logic code — examiners check this ✓ Comments in code = marks for technical documentation

🚀 Want to master IT programming and databases for your Grade 12 exam? 📞 Book a session with MindShift Learning today: +27 63 945 4868