DIGITAL GUIDE FOR BSIT STUDENTS

The Incoming BSIT Survival Guide

Everything I wish someone told me before first year — from a 6-year IT pro turned instructor. Plain Taglish, walang paligoy-ligoy.

Student Edition · 2026

Sir Vantes

Joshua Cervantes · IT Instructor & Industry Practitioner · joshuacervantes.online

PANIMULA

Bago ka magsimula

Kung binabasa mo 'to, ibig sabihin papasok ka na sa BSIT — o pinag-iisipan mo pa. Alinman diyan, nasa tamang lugar ka.

Anim na taon akong nasa totoong IT — Support at Operations — bago ako naging instructor. Nakita ko na kung paano nasasayang ang first two years ng maraming estudyante dahil walang nagsabi sa kanila ng mga simpleng bagay: anong aralin muna, anong gamit bibilhin, anong cert kukunin, at saan patungo ang lahat ng ito.

Hindi 'to textbook. Walang jargon na hindi ipapaliwanag. Ito 'yung usapan na sana may nagsabi sa 'kin noong first year ako — para hindi ka na mahirapan sa mga bagay na madali lang naman pala.

PAANO GAMITIN 'TO

Basahin mo lahat minsan para may overview ka. Pagkatapos, balik-balikan mo 'yung chapters habang umaandar ang semester. Ilagay mo sa phone mo — ito'y para sa'yo, kahit offline.

Tandaan: bawat estudyante iba't iba. Gamitin mo 'to bilang gabay, hindi bilang batas. Ang mahalaga — may malinaw ka nang direksyon mula araw one.

NILALAMAN

Ano ang laman nito

1 Ano ba talaga ang BSIT? (BSIT vs. CS vs. IS)

2 Ang mga subject mo, year by year — at alin ang mahalaga

3 Gadgets: ano bilhin, ano pwedeng i-skip (budget-friendly)

4 Free tools & accounts na i-set up bago ang day one

5 Paano mag-aral ng tech: labs bago lectures

6 Flowchart at logic — mag-isip bago mag-code

7 Ang unang code mo — coding basics (variables, if/else, loops)

8 Certs na worth kunin habang nag-aaral pa (A+, Network+, cloud)

9 Portfolio mula year 1 — GitHub, home labs, projects

10 Common first-year mistakes at paano iwasan

11 IT career paths pagka-graduate — kasama ang totoong sahod sa PH

12 Libreng resources: channels, courses, at communities

13 Bonus — Quick-Start Checklist: gawin agad bago mag-day one

14 Bonus — Glossary: mga IT term na madalas marinig

CHAPTER 1

Ano ba talaga ang BSIT?

Marami ang pumapasok sa BSIT na akala coding lang lahat. Ang totoo, isa lang 'yun sa maraming daan.

BSIT (Information Technology) ay tungkol sa paggamit at pagpapatakbo ng teknolohiya para tumulong sa mga tao at negosyo. Networks, systems, support, databases, security, saka konting programming — lahat 'yan hinihipo mo. Mas *hands-on* at *practical* ito kaysa purong theory.

Ang tatlong madalas ipinagkakamali

Course	Focus	Bagay kung...
BSIT Info Technology	Gamitin at patakbuhin ang tech — networks, systems, support, web, data.	Gusto mong hands-on, iba't ibang skill, at mabilis makapasok sa trabaho.
BSCS Computer Science	Theory, algorithms, at malalim na programming. Mas math-heavy.	Mahilig ka sa math, logic, at gustong gumawa ng software mula sa pundasyon.
BSIS Info Systems	Tech + business processes. Paano gamitin ang IT sa organisasyon.	Gusto mong pagdugtungin ang business at technology.

BOTTOM LINE

Walang "pinakamagaling" dito — depende sa'yo. Pero kung practical ka at gusto mong maraming pinto ang bukas pagka-graduate, tamang-tama ang BSIT.

Mga daan sa loob ng BSIT

Habang tumatagal, mahihilig ka sa isa sa mga ito. Okay lang na 'di mo pa alam ngayon — matutuklasan mo habang natututo:

- **Support & Systems** — ang go-to tech; dito nagsimula karamihan (kasama ako).
- **Networking** — ang nagpapatakbo ng internet at connection ng buong opisina.
- **Cybersecurity** — depensa laban sa hackers (mataas ang sahod, pero 'di entry-level).
- **Data** — pag-aanalyze ng numero para may basehan ang desisyon.
- **Web / Software Dev** — paggawa ng apps at websites.

Detalyado natin 'to — pati sahod — sa Chapter 11.

CHAPTER 2

Ang mga subject mo, year by year

Iba-iba ang curriculum kada school, pero halos pare-pareho ang balangkas. Eto ang hitsura ng apat na taon — at kung alin ang dapat mong seryosohin.

1st Year — Pundasyon

Intro to Computing, Programming 1 & 2, at general education (English, Math, PE). Dito madalas na-de-decide kung ma-i-in-love ka o matatakot ka sa coding.

- **Seryosohin:** Programming fundamentals. Kung matibay ka dito, magaan lahat ng susunod.
- **Tip:** Huwag mong isaulo ang code — intindihin mo. Malaking pagkakaiba 'yun.

2nd Year — Core Skills

Data Structures, Object-Oriented Programming, Databases (SQL), at Networking fundamentals.

- **Seryosohin:** Databases at Networking. Ito ang backbone ng halos lahat ng IT job.
- Dito mo malalaman kung "networking person" ka o "developer person" ka.

3rd Year — Specialization

Web/Mobile Development, System Administration, Information Security, at electives base sa track ng school mo.

- **Seryosohin:** 'Yung field na gusto mo. Dito ka na dapat nagsisimula ng portfolio.
- Simulan mo nang mag-ipon ng projects — Chapter 9 ang tungkol dito.

4th Year — Capstone & OJT

Thesis/Capstone project at On-the-Job Training (internship). Dito nagiging totoo lahat.

> **SOBRANG IMPORTANTE**
>
> Ang OJT mo ay madalas na unang linya sa resume mo — at minsan, dito ka na na-a-hire. Seryosohin mo. Dumating on time, magtanong, tumulong. Ang first impression, matagal.

> **BABALA**
>
> Ang GenEd subjects (English, Math, Rizal) — huwag mong ipagsawalang-bahala kahit "'di IT". Dito bumabagsak ang GWA ng maraming magaling mag-code. Balanse ang kailangan.

CHAPTER 3

Gadgets: ano bilhin, ano i-skip

Hindi mo kailangan ng mahal na gaming laptop para maging magaling. Eto ang totoo tungkol sa gear — para 'di masayang ang pera.

Laptop — ang pinaka-importante

Para sa BSIT, sapat na ang **reliable mid-range laptop**. Ito ang hanapin:

Parte	Sapat na	Bakit
RAM	**8GB minimum, 16GB ideal**	Para kaya ang IDE, browser, at virtual machines nang sabay.
Storage	**SSD, 256GB+**	Mas mabilis mag-boot at mag-load. Iwas HDD.
Processor	Ryzen 5 / Intel i5 (recent)	Sapat na para sa dev at networking labs.
OS	Windows o Mac — pareho okay	Matututo ka rin ng Linux, pero 'di kailangan bumili ng bago para dito.

PERA-SAVING TIP

Second-hand business laptop (ThinkPad, EliteBook, Latitude) ay madalas mas matibay at mas mura kaysa bagong "student" laptop. Type "refurbished business laptop" — sulit 'yan.

Ano ang PWEDE mong i-skip (sa ngayon)

- **Gaming laptop** — 'di kailangan maliban kung naglalaro ka rin. Init at bigat lang ang bonus.
- **Mahal na mechanical keyboard** — nice-to-have, hindi need.
- **Second monitor** — makakatulong, pero pwede munang hintayin.
- **Paid software** — halos lahat ng kailangan mo, libre. (Chapter 4)

Sulit na bilhin

- **USB flash drive (32GB+)** — para sa bootable OS at file transfer.
- **Basic external mouse** — mas mabilis magtrabaho kaysa trackpad.
- **Notebook at pen** — oo, totoo. Mabisa ang sulat-kamay sa pag-intindi.

CHAPTER 4

Free tools & accounts bago day one

I-set up mo 'to bago pa magklase para ready ka na agad. Lahat libre. Gamitin ang school o personal email na propesyonal ang dating.

Mga account na gawin ngayon

- **GitHub** — dito nakatira ang code at portfolio mo. Mag-sign up sa github.com, tapos kunin ang libreng GitHub Student Developer Pack — libo-libong pisong tools, libre para sa estudyante.
- **Google account (propesyonal)** — para sa Drive, Docs, at pag-sign up sa ibang tools. Iwas email na tulad ng "cutiepie_gamer123".
- **LinkedIn** — oo, first year pa lang. Simulan mo nang buuin ang network mo.
- **Microsoft/Azure for Students** — libreng cloud credits gamit ang school email.

Mga software na i-install

Gamit	Tool (libre)
Code editor	Visual Studio Code
Version control	Git
Networking labs	Cisco Packet Tracer
Virtual machines	VirtualBox
Terminal (Windows)	Windows Terminal + WSL
Notes / second brain	Obsidian o Notion

PRO MOVE

Gumawa ka ng isang folder na "Dev" o "School" sa laptop mo, maayos ang pagkaka-organisa mula simula. Ang magulong desktop, magulong utak.

CHAPTER
5

Paano mag-aral ng tech: labs bago lectures

Ang pinakamalaking sekreto: hindi ka natututo ng IT sa pakikinig lang. Natututo ka sa paggawa.

Sa loob ng anim na taon kong nasa field, wala akong natutunan sa pag-memorize ng slides. Lahat ng talagang tumatak — natutunan ko dahil **ginawa ko, nasira ko, at inayos ko ulit.**

Ang "labs over lectures" na paraan

1. **Panoorin/basahin** ang konsepto — mabilis lang, para may ideya.
2. **Gawin mo agad** — i-type ang code, i-set up ang network sa Packet Tracer, buuin ang VM.
3. **Sirain mo nang sadya** — anong mangyayari kapag inalis ko 'to? Dito ang totoong pag-intindi.
4. **Ipaliwanag mo sa iba** — kung kaya mong ituro, alam mo na talaga.

ANG "FEYNMAN" TRICK

Pagkatapos mag-aral ng topic, ipaliwanag mo na parang kausap mo ay 12 anyos. Kung na-stuck ka, 'yun ang bahaging kailangan mong balikan. Mabisa 'to nang sobra.

Practical na routine

- **Konsistent kaysa marami:** 1 oras araw-araw > 7 oras isang beses sa linggo.
- **Isang project kada topic:** may natutunang bago? Gumawa ng maliit na bagay gamit 'yun.
- **I-document mo:** magsulat ng notes o README. Ito magiging portfolio at reviewer mo.

IWASAN

"Tutorial hell" — 'yung panay panood ng tutorials pero walang sariling ginagawa. Nakaka-good feeling pero 'di ka umuusad. Pagkatapos ng bawat tutorial, gumawa ng sarili mong bersyon na walang kasamang gabay.

CHAPTER
6

Flowchart at logic — mag-isip bago mag-code

Bago ka mag-type ng kahit isang linya ng code, kailangan mo munang malaman kung *ano* ang ipapagawa mo sa computer. Dito pumapasok ang flowchart: litrato ng iyong logic bago pa ito maging code.

Sa Programming 1, tuturuan ka nito. 'Wag mong isipin na "sayang oras" — ang mga estudyanteng dumidiretso sa code na walang plano, sila ang naiiwan pag lumaki na ang problema. Ang flowchart, ito ang blueprint bago itayo ang bahay.

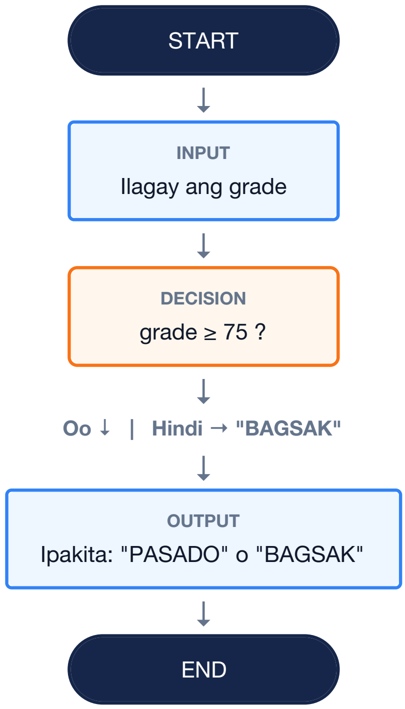
Ang 5 pangunahing simbolo

Halos lahat ng flowchart, gamit lang ang limang hugis na ito:

Simbolo	Hugis	Ano ginagawa
Terminal	⬭ **Oval**	Simula (START) o katapusan (END) ng program.
Input / Output	▱ **Parallelogram**	Pagtanggap ng data (input) o pagpapakita ng resulta (output).
Process	▭ **Rectangle**	Isang aksyon o kalkulasyon — hal. `total = a + b`.
Decision	◇ **Diamond**	Tanong na Oo/Hindi. Dito nag-a-*branch* ang daloy.
Flowline	→ **Arrow**	Nagpapakita ng direksyon — saan susunod.

Halimbawa: pasado ba o bagsak?

Sabihin nating gusto nating malaman kung pasado ang isang estudyante (grade na 75 pataas). Ganito ang flowchart:



Basahin mo mula taas pababa: nag-*start*, kumuha ng grade, nagtanong (decision), tapos nag-*output* base sa sagot. 'Yun lang. Kayang-kaya.

Pseudocode: tulay pa-code

Pseudocode ay ang parehong logic pero sinulat sa parang-Ingles, hindi tunay na code. Walang strict rules — basta maiintindihan ang hakbang:

```
# Pseudocode ng flowchart sa taas
```

```
SIMULA
  KUNIN ang grade
  KUNG grade >= 75
    IPAKITA "PASADO"
  KUNGDI
    IPAKITA "BAGSAK"
KATAPUSAN
```

Pansinin mo: flowchart → pseudocode → tunay na code. Bawat hakbang, mas malapit sa computer. Sa susunod na chapter, gagawin nating tunay na Python ang eksaktong logic na ito.

> **BAKIT ITO MAHALAGA**
>
> Kapag na-master mo ang pag-flowchart, ang coding ay nagiging "pag-translate" na lang ng plano mo sa language ng computer. Mas mabilis, mas kaunti ang bug. Ang mga senior dev, nagpa-plano pa rin bago mag-code — flowchart man o nasa isip lang.

FREE TRIAL

Kalahati na — 6 pang chapters ang natitira.

Ito ang free trial ng The Incoming BSIT Survival Guide. Nabasa mo na ang unang anim na chapters — kalahati ng buong guide. Nasa full version ang natitira.

Nasa full guide pa:

- Ang unang code mo — variables, if/else, loops (step by step)
- Certs habang estudyante ka pa — A+, Network+, cloud
- Portfolio mula year 1 — GitHub, home labs, projects
- Common first-year mistakes — at paano iwasan
- Career paths + totoong sahod sa Pilipinas
- Bonus: Quick-Start Checklist + Glossary ng IT terms

₱99 · one-time · lifetime access

Kunin ang buong 21-page guide →
joshuacervantes.online/guide

Sir Vantes
Joshua Cervantes · IT Instructor & Industry Practitioner · joshuacervantes.online