

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

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.

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

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 patakbuin 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.

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.

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.

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.

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.

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 naiwan pag lumaki na ang problema. Ang flowchart, ito ang blueprint bago itayo ang bahay.

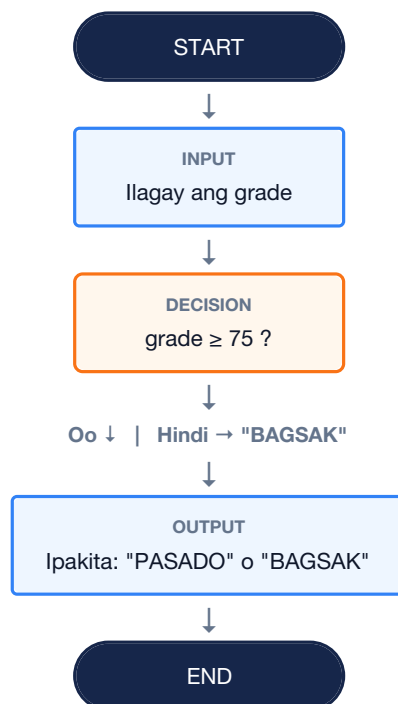
Ang 5 pangunahing simbolo

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. <code>total = a + b</code> .
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:

SIMULA

KUNIN ang grade

KUNG grade $\geq$ 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.

Ang unang code mo — coding basics

Hindi mo kailangang maging genius para mag-code. Anim na konsepto lang ang bumubuo ng halos lahat ng program. Kaya mo 'to sa isang upo.

Gagamit tayo ng **Python** — ang pinaka-beginner-friendly na language, at malamang ito rin ang gamit sa una mong subject. Ang mga `#` kulay-abo na text sa code ay *comment* — paliwanag lang, hindi binabasa ng computer.

1. Variable — lalagyan ng data

Isipin mong **kahon na may pangalan**. Nilalagyan mo ng laman, at makukuha mo ulit gamit ang pangalan.

```
pangalan = "Josh" # text (string)
edad = 27 # numero (integer)
print(pangalan) # ipapakita: Josh
```

OUTPUT

Josh

2. Data types — uri ng laman

Iba't ibang klase ng data, iba't ibang kilos:

- **String** — text, naka-quotes: `"Hello"`, `"BSIT"`
- **Integer** — buong numero: `27`, `100`
- **Float** — may decimal: `3.14`, `75.5`
- **Boolean** — `True` o `False` lang. Ito ang gamit sa mga desisyon.

3. Input at output — usap sa user

`input()` para tumanggap, `print()` para magpakita:

```
pangalan = input("Ano pangalan mo? ")
print("Kumusta, " + pangalan + "!")
```

OUTPUT

Ano pangalan mo? Josh
Kumusta, Josh!

GOTCHA

Lahat ng galing sa `input()` ay **string**, kahit numero ang tynpe. Para gawing numero: `edad = int(input("Edad: "))`. Ito ang #1 pinagkakamalian ng beginner.

4. If / else — desisyon

Ito na 'yung flowchart natin sa Chapter 6, ngayon tunay nang code:

```
grade = int(input("Ilagay ang grade: "))

if grade >= 75:
    print("PASADO")
else:
    print("BAGSAK")
```

OUTPUT

```
Ilagay ang grade: 82
PASADO
```

Kita mo? Eksaktong parehong logic ng flowchart at pseudocode — na-translate lang sa Python. 'Yun ang buong laro.

5. Loop — pag-ulit

Ayaw mong isulat ang parehong bagay ng 100 beses. Ang **loop** ang ulit para sa'yo:

```
for bilang in range(1, 6):
    print("Ulit numero", bilang)
```

OUTPUT

```
Ulit numero 1
Ulit numero 2
Ulit numero 3
Ulit numero 4
Ulit numero 5
```

6. Function — reusable na bloke

Function ay pinangalanang set ng hakbang na pwede mong tawagin ulit-ulit. Isulat minsan, gamitin kahit saan:

```
def bati(pangalan):
    print("Kumusta, " + pangalan + "!")

bati("Josh") # tawag 1
bati("Maria") # tawag 2
```

OUTPUT

```
Kumusta, Josh!
Kumusta, Maria!
```

Buuin natin: mini grade checker

Pagsamahin ang lahat — input, function, if/else, loop. Isang maliit pero *totoong* program:

```
def tsek_grade(grade):
    if grade >= 75:
        return "PASADO"
    else:
        return "BAGSAK"

bilang = int(input("Ilang subject? "))

for i in range(bilang):
    g = int(input("Grade: "))
    print(tsek_grade(g))
```

OUTPUT

```
Ilang subject? 2
Grade: 88
PASADO
Grade: 70
BAGSAK
```

Anim na konsepto, isang kapaki-pakinabang na program. Ganito lumalaki ang lahat — mula sa maliliit na piyesang ito.

Indentation — sa Python, mahalaga ang espasyo sa simula ng linya. Mali ang spaces? Error agad. **Colon (:)** — nakakalimutan sa dulo ng `if` , `for` , at `def` . **Quotes** — text kailangan naka-`" "` , kung hindi, akala ng Python variable name.

ANG TOTOONG SKILL

Hindi ang pagmemorya ng syntax ang mahirap — ma-Google mo naman lahat 'yun. Ang tunay na skill ay ang **paghati ng malaking problema sa maliliit na hakbang** (tulad ng flowchart). Kaya ka nagsisimula sa logic, hindi sa syntax.

Certs habang nag-aaral pa

Ang certification ay proof na kaya mo — kahit wala ka pang trabaho. Malaking edge 'to sa fresh grad. Eto ang worth simulan.

Mga entry-friendly na cert

Cert	Para kanino	Kailan kunin
CompTIA A+	Support & hardware basics. Magandang unang cert.	1st–2nd year
CompTIA Network+	Networking foundations. Malaking tulong sa lahat ng IT.	2nd–3rd year
Google IT Support	Beginner-friendly, online, self-paced.	1st–2nd year
AZ-900 / Cloud fundamentals	Intro sa cloud (Azure/AWS/OCI).	3rd year
Cisco CCNA	Seryosong networking. Mas advanced.	3rd–4th year
Security+ / Google Cybersecurity	Kung security ang gusto mo.	3rd–4th year

HUWAG MAG-PANIC

'Di mo kailangan lahat 'to. Isa o dalawang cert na tapat sa gusto mong field, sapat na para makapasok. Quality over quantity.

REALITY CHECK SA SECURITY

Maganda ang sahod sa cybersecurity, pero **hindi ito entry-level**. Karamihan dumadaan muna sa Support o Networking (2–5 taon) bago dito. Gawin mong long-term goal, hindi unang hakbang.

Marami sa mga cert na ito ay may libreng reviewer sa YouTube (kasama ang channel ko). 'Wag kang mahiya sa libre — 'yun ang pinaka-sulit.

Portfolio mula year 1

Sa IT, mas mahalaga ang **kaya mong ipakita** kaysa sa nasa diploma. Simulan mo agad — 'di na kailangang maghintay ng graduation.

Ano ang portfolio?

Koleksyon ng mga bagay na ginawa mo — kahit maliit. Nagpapakita ito sa employer na hindi ka lang nakikinig sa klase; gumagawa ka.

Mga unang project na kaya mo

- **Personal website** — kahit simple. Ito na mismo ang unang project.
- **Network diagram sa Packet Tracer** — mag-set up ng maliit na "opisina" na may VLANs at DHCP.
- **Home lab** — mag-install ng Linux sa VM, mag-set up ng simpleng server.
- **Automation script** — maliit na Python/Bash na nag-o-automate ng boring na task.
- **Class projects na pinaganda** — 'yung requirement, i-upload mo sa GitHub nang maayos.

ANG GITHUB AY RESUME MO

Bawat project, ilagay sa GitHub na may malinaw na README (ano 'to, paano gamitin, ano natutunan mo). Ang "green squares" sa GitHub profile mo — pruweba ng consistency.

Ang tamang mindset

Hindi kailangang perfect o malaki. Ang kailangan: **totoo, sarili mo, at naipaliwanag mo**. Mas okay ang 3 maliit na project na kaya mong ipaliwanag kaysa 1 malaking kinopya mo lang.

Common first-year mistakes

Mga pagkakamaling nakita ko nang paulit-ulit — at kung paano mo maiiwasan.

1. "Coding lang naman 'to" na mindset

Marami ang na-sha-shock na may networking, math, at documentation. Buksan mo ang isip — malawak ang IT, at 'yun ang ganda niya.

2. Pag-memorize imbes na pag-intindi

Pwede ka makalusot sa exam sa memorize, pero babagsak ka sa totoong trabaho. Intindihin ang *bakit*, hindi lang ang *ano*.

3. Paghihintay ng graduation bago "magsimula"

Ang portfolio, cert, at network — pwede simulan year 1. Ang naghihintay hanggang 4th year, huli na sa mga nauna.

4. Pagko-compare sa iba

May classmate kang mukhang genius? Normal 'yan. Iba-iba ang simula ng bawat tao. Ang kalaban mo — 'yung ikaw kahapon.

5. Takot magtanong

Walang bobo na tanong sa umpisa. Ang nahihiyang magtanong ang naiiwan. Lapitan mo ang prof, classmate, o online community.

ANG PINAKAMALAKING MISTAKE

Ang sumuko tuwing mahirap. Normal na parang "hindi para sa'kin 'to" 'yung pakiramdam sa umpisa — nangyari 'yan sa lahat, kasama ako. Ang pagkakaiba ng magaling at sumuko: nagpatuloy lang 'yung magaling.

Career paths + totoong sahod (PH)

Saan ka pwedeng pumunta pagka-graduate? Eto ang mga pangunahing daan, kasama ang tinatayang sahod sa Pilipinas (2026, entry hanggang mid-level).

Path	Ano ang trabaho	Sahod / buwan
IT Support	Ang go-to tech pag may sira. Pinakamadaling pasukan; magandang training ground.	P18K – 35K
Networking	Nagpapatakbo ng networks at connection. Umaakyat gamit ang certs (CCNA).	P25K – 50K
Web / Software Dev	Gumagawa ng apps at websites. Mataas ang ceiling lalo sa remote/foreign client.	P25K – 43K
Data Analyst	Nag-aanalyze ng datos para may basehan ang desisyon.	P20K – 43K
Cybersecurity	Depensa laban sa threats. Mataas ang sahod — pero 'di entry-level.	P68K – 78K+

*Tinatayang ranges lang ito at nag-iiba base sa kumpanya, lokasyon, skill, at kung remote/foreign ang client. Gamitin bilang gabay, hindi eksaktong pangako.

ANG TOTOONG SEKRETO SA SAHOD

Hindi ang course o school ang nagtatakda ng sahod mo — ang **skill, cert, at karanasan**. Dalawang BSIT grad, malayong-malayo ang kita depende sa ginawa nila habang nag-aaral. Ikaw ang may hawak niyan.

Remote at foreign clients: Isa sa pinakamalaking oportunidad ngayon. Maraming Pinoy IT pro ang kumikita nang higit sa local rate dahil remote sila sa dayuhang kumpanya. Ang kailangan: solidong skill, English communication, at portfolio.

Libreng resources

Lahat ng ito, libre. Ang internet ang pinakamalaking libreng paaralan — matutunan mo lang gamitin.

Pag-aralan (libre)

- **freeCodeCamp** — web dev, mula zero hanggang project-ready.
- **Cisco Networking Academy** — networking courses, madalas libre sa estudyante.
- **Google/Coursera "Audit"** — maraming course, libreng panoorin (audit mode).
- **Microsoft Learn** — libreng cloud at systems learning paths.
- **The Odin Project** — full-stack web dev, buo at libre.

Panoorin

- **Sir Vantes (ako!)** — Tagalog IT real talk, career guidance, at reviewers.
- **Professor Messer** — libreng CompTIA (A+, Network+, Security+) reviewers.
- **NetworkChuck** — networking at IT, masaya panoorin.

Sumali

- **Reddit:** r/ITCareerQuestions, r/CompTIA, r/networking
- **Discord communities** ng mga cert at dev — magtanong, matuto.
- **Local PH tech groups** sa Facebook — trabaho, tips, at network.

ANG TOTOO TUNGKOL SA "LIBRE"

Ang problema hindi kakulangan ng resources — sobra-sobra nga. Ang problema ay pagpili at pagtapos. Pumili ng isa, tapusin, saka lumipat. Huwag mag-collect ng courses na hindi natatapos.

Quick-Start Checklist

I-print o i-screenshot 'to. Tsekan mo habang ginagawa — pagkatapos nito, ready ka na bago pa magsimula ang klase.

Mga account — gawin ngayon

- ☐ GitHub account + GitHub Student Developer Pack
- ☐ Propesyonal na Google account (hindi "cutiepie123")
- ☐ LinkedIn profile — kahit basic muna
- ☐ Microsoft / Azure for Students (school email)

I-install sa laptop

- ☐ Visual Studio Code
- ☐ Git
- ☐ Python (para sa Chapter 7)
- ☐ Cisco Packet Tracer
- ☐ VirtualBox
- ☐ Obsidian o Notion (notes)

Unang galaw — first week

- ☐ Gumawa ng "Dev" o "School" folder, maayos ang ayos
- ☐ I-type at patakbuhan ang mini grade checker (Chapter 7)
- ☐ Gumawa ng unang GitHub repo — kahit "hello world" lang
- ☐ Pumili ng target na unang cert (Chapter 8)
- ☐ Mag-follow ng 1 channel + sumali sa 1 community (Chapter 12)

ISA-ISA LANG

Hindi kailangang tapusin lahat sa isang araw. Ang mahalaga — nag-umpisa ka na. Bumalik ka dito linggo-linggo hanggang wala nang naka-blanko.

Glossary — mga IT term

Mga salitang madalas mong maririnig sa BSIT. Balikan mo 'to kapag may hindi pamilyar — normal lang 'yun sa umpisa.

Term	Ibig sabihin
IDE	Program na pinag-co-codean, may built-in na tulong (hal. VS Code).
Repository (repo)	Folder ng project sa GitHub — kasama ang buong history ng mga pagbabago.
Git	Tool na nagta-track ng bawat baguhin sa code. Pwede kang bumalik sa dating bersyon.
CLI / Terminal	Text-based na paraan ng pag-utos sa computer — walang mouse, puro type.
VM (Virtual Machine)	"Computer sa loob ng computer" — para makapag-test nang ligtas.
VLAN	Paghahati ng isang network sa mas maliliit, magkakahiwalay na parte.
DHCP	Awtomatikong nagbibigay ng IP address sa bawat device sa network.
IP address	Ang "address" ng isang device para makausap ito sa network.
SSD	Mabilis na storage — mas mabilis mag-boot kaysa lumang HDD.
OS (Operating System)	Windows, macOS, o Linux — ang nagpapatakbo sa buong computer.
Syntax	Ang "grammar" ng isang programming language.
Bug	Mali sa code na nagbibigay ng maling resulta o error.
Framework	Handa nang balangkas para mas mabilis gumawa ng app o website.
Open-source	Libre at bukas ang code — pwede gamitin, aralin, at baguhin ng kahit sino.

Ang susunod na hakbang

Kung isang bagay lang ang tatandaan mo sa buong guide na 'to, ito na 'yun:

Hindi ka na kailangang mag-antay ng "tamang panahon." Ang tamang panahon ay ngayon. I-set up ang GitHub. Gumawa ng unang maliit na project. Piliin ang unang cert. Maliit man, umuusad ka na — at 'yun ang mahalaga.

Anim na taon ako sa field bago naging instructor. Kung may nagbigay sana sa'kin ng guide na 'to noong first year, mas mabilis sana ako umusad. Ngayon, nasa'yo na 'yun. Gamitin mo.

Isama mo ako sa journey mo

May tanong? Stuck ka? Gusto mo ng gabay habang nag-aaral? Nandito lang ako.

YouTube: youtube.com/@ytsirvantes (Sir Vantes)

TikTok: tiktok.com/@sir.vantes

Website: joshuacervantes.online

Email: yt.sirvantes@gmail.com

Facebook: facebook.com/joshuacruzcertvantesx

Salamat sa tiwala. Kaya mo 'to. — Sir Vantes