

Prompt█Powered Kickstart: Getting Started with Go by Building a Tiny CLI

****Duration:**** Monday 4th August – Friday 8th August\

****Author:****

1) Title and Objective

****Title:**** Hello Go – a beginner's toolkit to build and run a simple command█line app.\

****Objective:**** Learn the basics of Go, set up the toolchain, create a runnable CLI that prints a greeting with flags, and document the AI prompts that helped along the way.

2) Quick Summary of the Technology

****What is Go?****\

Go is an open█source programming language created at Google. It focuses on simplicity, speed, and easy concurrency.

****Where is it used?****\

Go powers network services, CLIs, APIs, and cloud tools. It is popular at companies like Google, Uber, and Cloudflare.

****Real█world example****\

Docker and Kubernetes use Go under the hood.

3) System Requirements

- ****OS:**** Windows 10 or later, macOS 12 or later, or a recent Linux distro
- ****Editor:**** VS Code or any editor you like
- ****Language runtime:**** Go 1.22+ (from <https://go.dev/dl>)
- ****Terminal:**** PowerShell (Windows), Terminal.app or iTerm2 (macOS), any standard Linux shell

4) Installation and Setup

Follow these steps exactly. Copy the terminal lines and run them in your shell.

4.1 Install Go

- Download and install Go from <https://go.dev/dl>
- Close and reopen your terminal after install
- Verify your install:

```
go version
```

You should see something like: `go version go1.22.x ...`

4.2 Create the project

```
# Create a folder and enter it
mkdir moringa-go-cli && cd moringa-go-cli

# Initialize a module (creates go.mod)
go mod init moringa-go-cli
```

4.3 Add code

Create a file named `main.go` with this content:

```
package main

import (
    "flag"
    "fmt"
    "time"
)

func main() {
    name := flag.String("name", "Moringa", "Name to greet")
    count := flag.Int("count", 1, "How many times to print")
    flag.Parse()

    for i := 0; i < *count; i++ {
        fmt.Printf("Hello, %s! Time: %s\n", *name, time.Now().Format(time.RFC3339))
    }
}
```

4.4 Run it

```
# Run without building a binary
go run .

# Run with flags
go run . -name "Shukri" -count 3
```

Expected output:

```
Hello, Shukri! Time: 2025-08-04T09:01:22+03:00
Hello, Shukri! Time: 2025-08-04T09:01:22+03:00
Hello, Shukri! Time: 2025-08-04T09:01:22+03:00
```

4.5 Build a binary

```
# Build a local executable in the current folder
# On Windows this produces moringa-go-cli.exe
# On macOS/Linux it produces moringa-go-cli

go build -o moringa-go-cli

# Run it
./moringa-go-cli -name "Amina" -count 2
```

4.6 Optional: Add a Makefile (Linux/macOS) or simple scripts

```
run:
\tgo run .

build:
\tgo build -o moringa-go-cli
```

5) Minimal Working Example

****What it does****\

A tiny command-line program that prints a greeting with the current timestamp. Two flags control behavior:

- ``-name`` sets who to greet
- ``-count`` repeats the message N times

****Key ideas****

- Reading flags with the standard ``flag`` package
- Printing formatted strings with ``fmt``
- Getting a timestamp with ``time``

****Expected output****\

Running ``go run . -name "Zainab" -count 2`` prints two lines like:

```
Hello, Zainab! Time: 2025-08-04T09:01:22+03:00
Hello, Zainab! Time: 2025-08-04T09:01:22+03:00
```

6) AI Prompt Journal

Record your exact prompts and a short note on what helped. Replace the placeholders with your own conversations. Add links to your ai.moringaschool.com chat threads if available.

1. ****Prompt:**** Explain Go at a beginner level and list what I need to install on Windows, macOS, and Linux.\
****AI helped with:**** A clean checklist for installs and verification.\
****Curriculum link:****
2. ****Prompt:**** Show me how to create a new Go module and explain `go.mod`.\
****AI helped with:**** Clear steps and the purpose of modules.\
****Curriculum link:****
3. ****Prompt:**** Write a tiny Go program that reads two flags, name and count, and prints a greeting with a timestamp.\
****AI helped with:**** Code scaffold I could paste and run.\
****Curriculum link:****
4. ****Prompt:**** I get "go: go.mod file not found" when I run go commands. What did I miss?\
****AI helped with:**** The fix was to run ``go mod init`` in the project folder.\
****Curriculum link:****
5. ****Prompt:**** How do I build a binary and run it on Windows versus macOS/Linux?\
****AI helped with:**** OS-specific run commands and file names.\
****Curriculum link:****
6. ****Prompt:**** Show me how to parse flags and print a helpful ``-h`` message.\
****AI helped with:**** Using ``flag.Usage`` and clear descriptions.\
****Curriculum link:****

7. **Prompt:** Suggest 3 small improvements to my program for better UX.\n**AI helped with:** Ideas like default values, examples in `-h`, and input validation.\n**Curriculum link:**
8. **Prompt:** How can a peer test my CLI quickly and give feedback?\n**AI helped with:** A simple test checklist and expected outputs.\n**Curriculum link:**

Add your own prompts as you iterate.

7) Common Issues and Fixes

- ```\``
Go is not on your PATH. Reinstall Go from <https://go.dev/dl> and make sure your terminal sees it. Close and reopen the terminal, then run `go version`.
- ```\``
You ran a Go command outside your project or you forgot to initialize the module. Use `cd` into your project folder and run `go mod init moringa-go-cli`.
- Antivirus blocks the binary on Windows**
Click “Allow” if prompted. Build again if needed.
- Permission to execute on Linux/macOS**
If needed, run `chmod +x moringa-go-cli` before executing the binary.
- Wrong folder**
VS Code opened the parent folder instead of the project. Open the exact folder that contains `go.mod` and `main.go`.

8) References

- The Go Programming Language docs – <https://go.dev/doc>
- Installing Go – <https://go.dev/dl>
- Command line flags – <https://pkg.go.dev/flag>
- Time formatting – <https://pkg.go.dev/time>
- Effective Go – https://go.dev/doc/effective_go

9) Repo Structure and README

Suggested structure:

```
moringa-go-cli/  
├── main.go  
├── go.mod  
├── README.md  
└── .gitignore
```

README.md example

```
# moringa-go-cli
```

A tiny Go commandline app that prints a greeting with a timestamp.

Prerequisites

- Go 1.22+

Run

go run .

Example

go run . -name "Shukri" -count 3

Build

go build -o moringa-go-cli

./moringa-go-cli -name "Amina" -count 2

Flags

- `-name`` Name to greet. Default: ``Moringa``
- `-count`` How many times to print. Default: ``1``

10) Peer Test Checklist

Ask a classmate to try the following and note what worked or failed.

- Can they run ``go version`` successfully
- Can they clone or unzip the project and run ``go run .``
- Do the flags work as expected
- Is the help text clear when running ``go run . -h``
- Can they build a binary and run it
- Notes on clarity of the README

11) Submission

- ****Docs:**** export this file as Markdown or PDF
- ****Code:**** push to GitHub or zip the folder
- ****Include:**** a README with run instructions and a link to your AI Prompt Journal

Good luck and happy building.