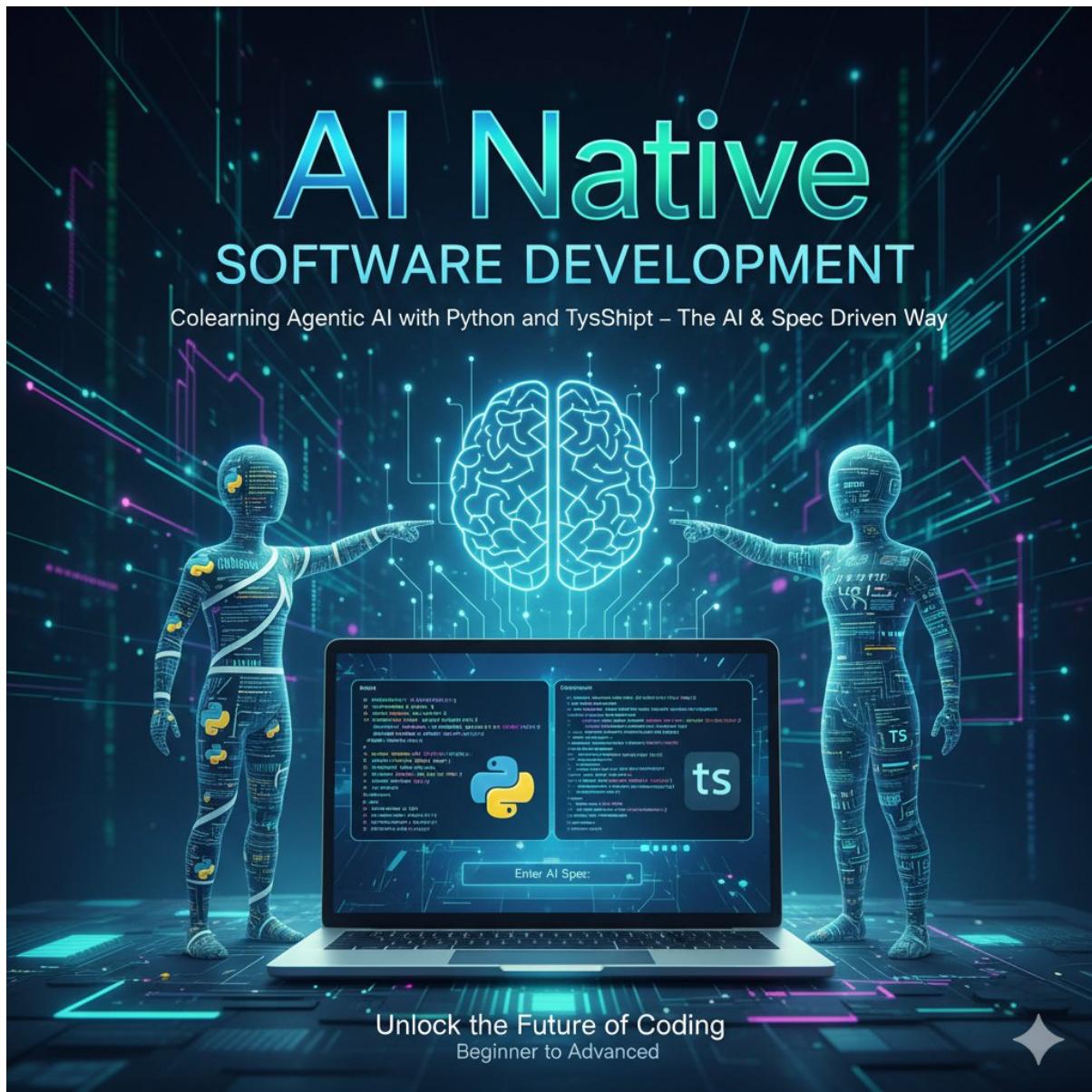


# AIDD (AI-Driven Development) 30-Days Challenge - Task 8 Submission

**Participant:** Anum Kamal (00326783)

**Slot:** Friday - 6:00 PM to 9:00 PM

**Instructor:** Sir Hamzah Syed



# Simple Calculator using SpecKitPlus

## Task Overview:

Build a calculator using SpecKitPlus, and complete all 5 phases:

1. /constitution
  2. /specify
  3. /plan
  4. /tasks
  5. /implement

## Terminal / CLI Commands

```
Windows PowerShell
PS C:\Users\anumk\Desktop> specifyplus init task8-calculator
SpecifyPlus
Panaversity SpecKit Plus - Spec-Driven Development Toolkit
Specify Project Setup
Choose your AI assistant:
Project          task8-calculator
Working Path     C:\Users\anumk\Desktop
Target Path      C:\Users\anumk\Desktop\task8-calculator

▶ copilot (GitHub Copilot)
  claude (Claude Code)
  gemini (Gemini CLI)
  cursor-agent (Cursor)
  qwen (Qwen Code)
  opencode (opencode)
  codex (Codex CLI)
  windsurf (Windsurf)
  kilocode (Kilo Code)
  auggie (Auggie CLI)
  codebuddy (CodeBuddy)
  roo (Roo Code)
  q (Amazon Q Developer CLI)
  amp (Amp)
```

```
Windows PowerShell x + v
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\anumk\Desktop> specifyplus init task8-calculator
SPECIFYPLUS
Panaversity SpecKit Plus - Spec-Driven Development Toolkit

Specify Project Setup
Project      task8-calculator
Working Path  C:\Users\anumk\Desktop
Target Path   C:\Users\anumk\Desktop\task8-calculator

Choose script type (or press Enter)
▶ sh (POSIX Shell (bash/zsh))
ps (PowerShell)
Use ↑/↓ to navigate, Enter to select, Esc to cancel
```

```
Windows PowerShell x + v
PS C:\Users\anumk\Desktop> specifyplus init task8-calculator
SPECIFYPLUS
Panaversity SpecKit Plus - Spec-Driven Development Toolkit

Specify Project Setup
Project      task8-calculator
Working Path  C:\Users\anumk\Desktop
Target Path   C:\Users\anumk\Desktop\task8-calculator

Selected AI assistant: claudie
Selected script type: ps
Initialize Specify Project
└─ [●] Check required tools (ok)
└─ [●] Select AI assistant (claudie)
└─ [●] Select script type (ps)
└─ [●] Fetch latest release (release v0.0.19 (77,209 bytes))
└─ [●] Download template (spec-kit-template-claudie-ps-v0.0.19.zip)
└─ [●] Extract template
└─ [●] Archive contents (32 entries)
└─ [●] Extraction summary (3 top-level items)
  └─ [○] Ensure scripts executable
└─ [●] Cleanup
└─ [●] Initialize git repository (existing repo detected)
└─ [●] Finalize (project ready)
```

```

Windows PowerShell
+ 
  • Cleanup
  └ Initialize git repository (existing repo detected)
  • Finalize (project ready)

Project ready.

Agent Folder Security
Some agents may store credentials, auth tokens, or other identifying and private artifacts in the agent folder within your project.
Consider adding .claude/ (or parts of it) to .gitignore to prevent accidental credential leakage.

Next Steps
1. Go to the project folder: cd task8-calculator
2. Start using slash commands with your AI agent:
   2.1 /sp.constitution - Establish project principles
   2.2 /sp.specify - Create baseline specification
   2.3 /sp.plan - Create implementation plan
   2.4 /sp.tasks - Generate actionable tasks
   2.5 /sp.implement - Execute implementation

Enhancement Commands
Optional commands that you can use for your specs (improve quality & confidence)
o /sp.clarify (optional) - Ask structured questions to de-risk ambiguous areas before planning (run before /sp.plan if used)
o /sp.analyze (optional) - Cross-artifact consistency & alignment report (after /sp.tasks, before /sp.implement)
o /sp.checklist (optional) - Generate quality checklists to validate requirements completeness, clarity, and consistency (after /sp.plan)

PS C:\Users\anumk\Desktop>

```

## 1. [/constitution](#)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Claude Code v2.0.56
Sonnet 4.5 · API Usage Billing
~\Desktop\task8-calculator

/model to try Opus 4.5

> /sp.constitution is running.. "Build a Basic Calculator CLI program.
It must support: addition, subtraction, multiplication, division (with zero-division handling).
User enters two numbers + operator. Output must be clear.
Program must display: "Welcome to the Simple CLI Calculator!" at start.

• Read(.specify\memory\constitution.md)
  _ Read 56 lines

• Write(.specify\memory\constitution.md)

Overwrite file
.specify\memory\constitution.md

1 - # [PROJECT_NAME] Constitution
2 - <!-- Example: Spec Constitution, TaskFlow Constitution, etc. -->
1 + <!--
2 + Sync Impact Report:
3 + Version change: 0.0.0 → 1.0.0
4 + Modified principles:
5 +   - [PROJECT_NAME] → Basic Calculator CLI program
6 +   - [PRINCIPLE_1_NAME] → Robust Arithmetic Operations
7 +   - [PRINCIPLE_2_NAME] → Clear User Interface
8 +   - [PRINCIPLE_3_NAME] → Input Validation
9 +   - [PRINCIPLE_4_NAME] → Maintainability and Readability

```

## 2. /specify

```
● Read(.specify\templates\phr-template.prompt.md)
  └ Read 46 lines

> /sp.specify is running... "Requirements:
1. Program runs in Python.
2. Functions: add(a,b), subtract(a,b), multiply(a,b), divide(a,b with zero check).
3. main.py must:
   - print "Welcome to the Simple CLI Calculator!"
   - ask user for two numbers
   - ask for operator (+,-,*,/)
   - call correct function
   - display result
4. Store logic in src/calculator.py
5. main.py is entry point"

● Bash(git fetch --all --prune)
  └ Running...



---

Bash command

git fetch --all --prune
Fetch all remote branches

Do you want to proceed?
> 1. Yes
  2. Yes, and don't ask again for git fetch commands in C:\Users\anumk\Desktop\task8-calculator
  3. Type here to tell Claude what to do differently

Esc to exit
```

## 3. /plan

```
user: anumk

> /sp.plan is running... " 1. Create src/calculator.py
2. Implement add()
3. Implement subtract()
4. Implement multiply()
5. Implement divide() with zero-division handling
6. Create main.py
7. Add welcome message: "Welcome to the Simple CLI Calculator!"
8. Add user input prompts
9. Add operator logic
10. Print final result"

● Read(specs\calculator-basic-ops\spec.md)
  └ Read 100 lines

● Read(.specify\memory\constitution.md)
  └ Read 39 lines

● Read(.specify\templates\plan-template.md)
  └ Read 105 lines

● Write(specs\calculator-basic-ops\plan.md)



---

Create file

specs\calculator-basic-ops\plan.md

# Implementation Plan: Basic Calculator CLI Program

**Branch**: `1-calculator-basic-ops` | **Date**: 2025-12-02 | **Spec**:
[specs/calculator-basic-ops/spec.md](specs/calculator-basic-ops/spec.md)

**Input**: Feature specification from `/specs/calculator-basic-ops/spec.md`

**Note**: This template is filled in by the `/sp.plan` command. See `./specify/templates/commands/plan.md` for the execution workflow.
```

## 4. /tasks

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
> /sp.tasks is running... " T001: Create src/calculator.py
T002: Implement add()
T003: Implement subtract()
T004: Implement multiply()
T005: Implement divide() with zero-division handling
T006: Create main.py
T007: Add welcome message
T008: Add user input prompts
T009: Add operator selection logic
T010: Print final result
T011: Test program "
● Read(specs\calculator-basic-ops\plan.md)
  └ Read 66 lines
● Read(specs\calculator-basic-ops\spec.md)
  └ Read 100 lines
● Read(.specify\templates\tasks-template.md)
  └ Read 252 lines
● Write(specs\calculator-basic-ops\tasks.md)

Create file
specs\calculator-basic-ops\tasks.md

# Tasks: Basic Calculator CLI Program

**Input**: Design documents from `/specs/calculator-basic-ops/`
**Prerequisites**: plan.md (required), spec.md (required for user stories)

**Tests**: Test tasks are included as they are crucial for validating the calculator's functionality.

**Organization**: Tasks are grouped by user story to enable independent implementation and testing of each story.
```

## 5. /implement

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
> /sp.implement is running...
● Bash(powershell.exe -Command "& {.specify/scripts/powershell/check-prerequisites.ps1 -Json -RequireTasks -IncludeTasks}")
  └ ERROR: plan.md not found in C:\Users\anumk\specs\001-featurename-1-calculator
    Run '/sp.plan first to create the implementation plan.

● The check-prerequisites.ps1 script failed because plan.md was not found. I had previously created specs/001-featurename-1-calculator/plan.md. It seems the script is looking in a different location or there's an issue with the path.

I need to investigate why plan.md is not being found in the expected location. I will first list the contents of the specs/001-featurename-1-calculator/ directory to confirm the file's presence.

● Bash(ls -F specs/001-featurename-1-calculator/)
  └ plan.md
    tasks.md

● It seems there's an issue with the check-prerequisites.ps1 script not correctly identifying plan.md, even though the file is present in the specs/001-featurename-1-calculator/ directory.

I will proceed by directly loading and analyzing the implementation context, as outlined in step 3 of the /sp.implement command, since I've confirmed the plan.md and tasks.md files exist.

First, I will read specs/001-featurename-1-calculator/plan.md.

● Read(specs\001-featurename-1-calculator\plan.md)
  └ Read 71 lines

● Read(specs\001-featurename-1-calculator\tasks.md)
  └ Read 14 lines

● Bash(git rev-parse --git-dir 2>/dev/null)
  └ C:/Users/anumk/.git

● Read(.gitignore)
  └ Read 42 lines
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Microsoft Windows [Version 10.0.26200.7309]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\anumk\Desktop\task8-calculator>python main.py
Welcome to the Simple CLI Calculator!
Enter the first number (or 'q' to quit): 50
Enter an operator (+, -, *, /) (or 'q' to quit): +
Enter the second number (or 'q' to quit): 30
Result: 80.0
```

```
Do you want to perform another calculation? (yes/no/q to quit): yes
Enter the first number (or 'q' to quit): 50
Enter an operator (+, -, *, /) (or 'q' to quit): -
Enter the second number (or 'q' to quit): 20
Result: 30.0
```

```
Do you want to perform another calculation? (yes/no/q to quit): yes
Enter the first number (or 'q' to quit): 20
Enter an operator (+, -, *, /) (or 'q' to quit): /
Enter the second number (or 'q' to quit): 2
Result: 10.0
```

```
Do you want to perform another calculation? (yes/no/q to quit): yes
Enter the first number (or 'q' to quit): 3
Enter an operator (+, -, *, /) (or 'q' to quit): *
Enter the second number (or 'q' to quit): 5
Result: 15.0
```

```
Do you want to perform another calculation? (yes/no/q to quit): q
```

```
C:\Users\anumk\Desktop\task8-calculator>
```

[Here is the task GitHub link:](#)

 <https://github.com/Anum-2017/AIDD-Task-08.git>