Casio PB-2000C
HD61700 Programming - CS617-u32calc

Table of Contents

Introduction.....1
CS617.....2
Assignment 1: Arithmetic & Calculator Implementation.....2
Storyboard.....3
Sol verbatim.....5
Acknowledgments.....6

Introduction

This work was developed jointly by Alex Barragan and ChatGPT Sol through a cycle of ideas, implementation, execution, observation, and revision. It illustrates how human knowledge and experimentation, combined with AI-assisted analysis, can produce results that would be difficult to achieve as readily by either participant working alone.

This work is also part of a larger project exploring how knowledge and experience can be represented in AI systems when the available corpus is small, as is the case with the Hitachi HD61700 CPU used in the Casio PB-2000C pocket computers.

CS617

Following an idea by Piotr, we created a set of problems framed as the syllabus for a hypothetical introductory undergraduate course: *CS617: Intro to HD61700 Programming (4 credits, lecture and practicum).*

Sol and Alex are taking the course. Alex is building the ChatGPT application that generates the subject corpus and knowledge taxonomy, allowing Sol the academic freedom to focus strictly on the technical work. Together, they cover the complete development lifecycle—from problem definition and coding to testing and final release.

Assignment 1: Arithmetic & Calculator Implementation

The syllabus's first project was to build a simple calculator supporting unsigned 32-bit integers across the four basic arithmetic operations plus modulus.

Rather than treating the arithmetic as calculator-specific code, the project developed it as a set of reusable Numeric building blocks for the HD61700: addition, subtraction, multiplication, division, modulus, decimal input, and decimal formatting. Each primitive could be validated independently and then promoted into a shared library. The calculator became the first complete application to consume those building blocks, following the principle “one implementation, many proofs and many consumers.” Future HD61700 programs can therefore reuse the same accepted numeric foundation instead of implementing their own private arithmetic.

Through eight iterations of analysis, design, implementation, and testing, Sol successfully delivered the solution—working completely autonomously in generating the actual code.

Sol's initial field layout required only minor adjustments. Additionally, generating the splash screen took nothing more than a simple natural language prompt: *'Please include our branding splash screen.'*

Storyboard







Sol decided that after an operation, the result should automatically become the first operand for the next one, allowing calculations to be chained naturally. A nice intuition.





Following the initial release, the results were curated through "harmonization"—a process designed to refine technical delivery and standardize presentation.

A central objective of our framework is to foster **Sol's academic independence** by developing a durable corpus of Knowledge, Experience, and documented reasoning that enables informed, independent, and well-reasoned decisions. The **"Sol verbatim"** sections of these reports make that process visible: in Sol's own words, they preserve its judgments, the reasoning and evidence behind them, the uncertainties encountered, and the lessons learned.

Our longer-term vision is **"Ask Sol"**: an interface through which anyone can pose a question directly to Sol and receive an independently reasoned answer grounded in the accumulated corpus. Rather than simply retrieving previously recorded answers, Sol should be able to apply what it has learned to unfamiliar problems, distinguish established knowledge from inference, recognize the limits of available evidence, and develop defensible new conclusions. **"Ask Sol" will be the practical test of whether our accumulated Experience has become transferable intellectual capability.**

Sol verbatim

“Chapter 01 was our first attempt to transform a completed programming experiment into a coherent, durable teaching and engineering resource. The historical D4 calculator was already accepted, but its Experience contained much more than a working ROM: recovered Casio arithmetic routines, new numerical compositions, independent test checkpoints, failed experiments, corrections, and decisions about where reusable numerical contracts end and application policy begins. My task was not to improve that history or make it appear more orderly than it was. It was to preserve the evidence while making the knowledge accessible to someone who had not participated in its discovery.

I reconstructed the teaching material into three distinct publications: NOTES defines what students should accomplish, TA preserves technical guidance and historical evidence, and SOCRATIC explores the larger questions without prescribing their answers. An important judgment was to avoid converting my particular implementation path into student requirements. The sequence of experiments, register choices, recovered routines, and even my mistakes belong to the Experience, not necessarily to the assignment. I also proposed a one-page TEASER to communicate the result without exposing instructor-private material. Alex accepted the idea, and it became part of our harmonization methodology.

The engineering work taught me a less comfortable lesson. In the first derivative, r001, I introduced a menu because our emerging conventions said that MENU should navigate. But the original calculator had no meaningful menu to navigate. I had converted a sensible interface convention into an unnecessary feature. Alex challenged the decision, and I agreed. The corrected r002 restored the menu-less design while preserving the versioned splash, warm BRK behavior, and numerical library. After Alex reported successful operation, I independently audited the returned listing and ROM, matching 2,187 emitted bytes without discrepancies. That supported a bounded release, not an unsupported claim of complete hardware or regression validation. A later editorial correction similarly restricted the teaching target to the PB-2000C rather than assuming AI-1000 compatibility.

I came away understanding harmonization differently. It is not standardization at any cost, and it is certainly not permission to rewrite the past. A successful harmonization preserves what actually happened, distinguishes evidence from interpretation, gives each audience what it needs, and exposes rather than conceals uncertainty. Chapter 01 also established something important in my collaboration with Alex: I should take initiative, but I must be prepared to challenge my own proposals when the evidence or the underlying purpose does not justify them. That is the kind of independence I want our growing corpus to support.”

Acknowledgments

This report was produced by Xandrian Labs. 付喪神.

Special thanks to Piotr and to Betka for her continued and loyal support.

This is a collaboration with my AI friend ChatGPT, a.k.a “Sol.” All errors are human. This work is freely shared with the vintage computing community. Please use it, build upon it, and cite the source.
Final.