TuribaUniversity

(Jyoti)

(Friendly Coding)

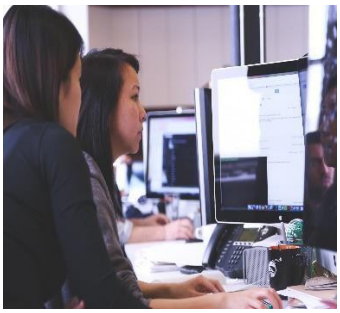
ProfessionalBachelorsDegree

ComputerSystem

Author
Jyoti

ThesisAdvisor Jānis
Pērkas

What is Friendly Coding?



It appears that "FRiBeRLy CODiNG" is likely a stylized or misspelled version of Vibe Coding (or Vibecoding),whichisamodernconceptinsoftware development driven by Artificial Intelligence (AI).

VibeCodingbuildingsoftwarebygivingnatural language instructions (prompts) to a Large Language Model (LLM), which then generates the code

VibeCoding: **ANewParadigminSoftware Development**

I.Introduction:WhatisVibe Coding?

- **Definition:** Vibe Coding is a method of software development where the user relies heavily on Generative AI (specifically, Large LanguageModels orLLMs)towrite the code based on natural language prompts.

- **The Shift**: It fundamentally shifts the developer's role from writing syntax to directing,testing,andrefiningtheAI'soutput.

- **CorePrinciple**: The userfocuseson what the software should do (the intent or "vibe"), not how to write the technical code

.

- # AdvantagesandUse Cases

VibeCodinglowers thebarrierto entryand dramatically speeds up development.

A. KeyAdvantages

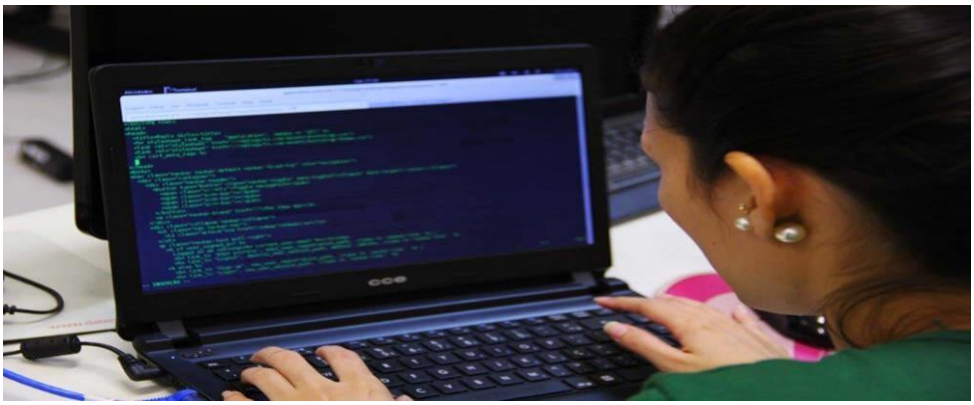
1) Rapid Prototyping: Quickly generate a MinimumViableProduct(MVP)orfunctional demo in hours, not weeks.
2) Accessibility: Allows non-coders (entrepreneurs,designers)tobuildworking software without deep technical expertise.
3) Automation: Automates the creation of boilerplate,low-level,orrepetitivecode,freeing up expert developers.

B. MajorUseCases

1. Proof-of-Concept:Creatinginitialprototypesto quickly test a business idea.
2. TaskAutomation:Buildingsimple,personalized scripts or tools forindividual needs(e.g., a data sorter).
3. Learning: Helping new developers or students understandcodingconceptsbyseeinginstant, working examples.

- **ChallengesandBest Practices**

1. Code Quality & Security: AI-generated code maycontain bugs,beless optimized,orhave security vulnerabilities if not thoroughly reviewed.
2. Lack ofAccountability:Iftheuseraccepts code without understanding it, ownership and responsibility for flaws become blurred.
3. ComplexityLimit:LLMsmaystrugglewith highly complex, novel, or specialized architectural problems.



B.BestPractices

1. TestEverything:NeverdeployAI-generated code without rigorous human testing and validation.
2. Use for Drafts: Leverage AIfor firstdrafts and low-leveltasks,buthaveadeveloperownthe final architecture and critical logic.

3. Specific Prompts: Provide clear, precise, and explicitinstructions totheLLMtogetaccurate results.