A Specialized Tool to Assist in the Creation of Basic Object-Oriented Java Programs

Max Gastelum
Senior Design

The Problem

- Advanced programming tools like, generative AI disincentivizes the process of actually learning how to program.
- Will this lead to extinction of skill programmers?
- Character Amnesia
  - A phenomenon where many Chinese and Japanese speakers are no longer able to handwrite words that they can easily read and produce electronically (1).
  - Some believe the most appropriate solution is acceptance (1).
- Will reliance on advance technologies like ChatGPT for programming result in a similar scenario?

Java

C

Python

?

The Method

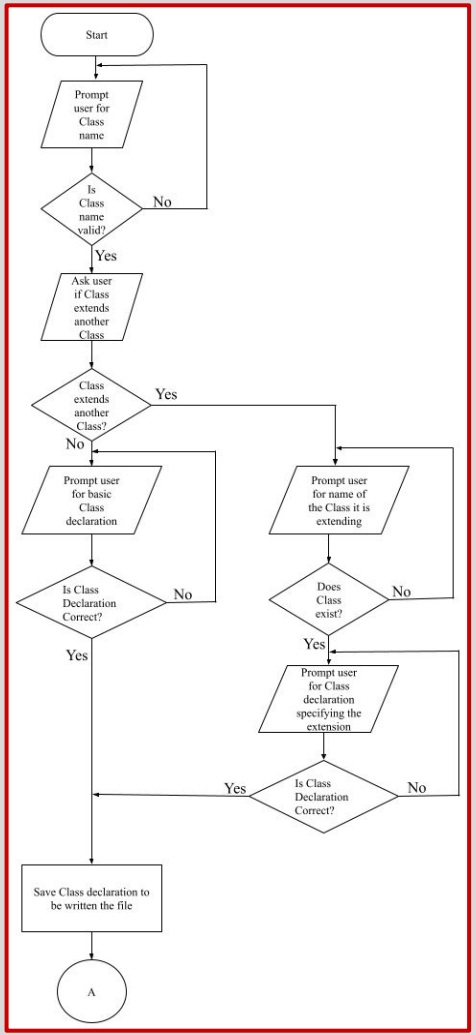
- A simple Java program that will assist users in the creation of object-oriented Java programs.
- The program will prompt users for the type of Java Class they would like to create and walk them through the process of creating it.
- The program will require the users to provide all the code themselves.
- The program will enforce adherence to common conventions, rules, and best practices.
- The main goal of the program is to offer beginners with a means to learn object-oriented programming while highlighting its strength in breaking down large problems and emulating the real world.

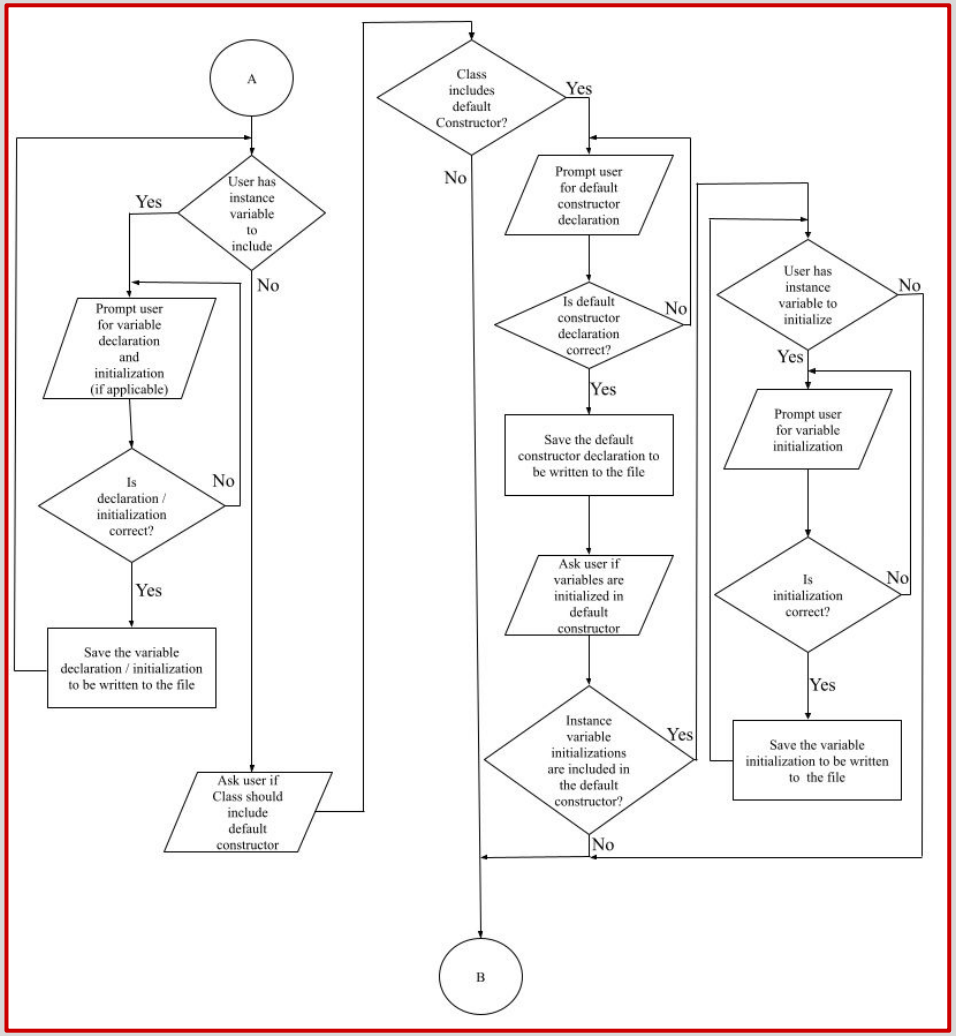
Comparison of Methods

- ChatGPT - College students produced significantly larger amount of code violations when completing object-oriented programming tasks with or without ChatGPT (2).
- Programming Video Games - Most of the games reviewed for in class use were short form and do not cover all the required topics (3).
- Block based programming - Often targeted towards young or first time programmers and especially helpful in teaching procedural programming.
- The goal of the proposed method is not to replace these solutions but fill some of the gaps left open and offer a specialized solution.

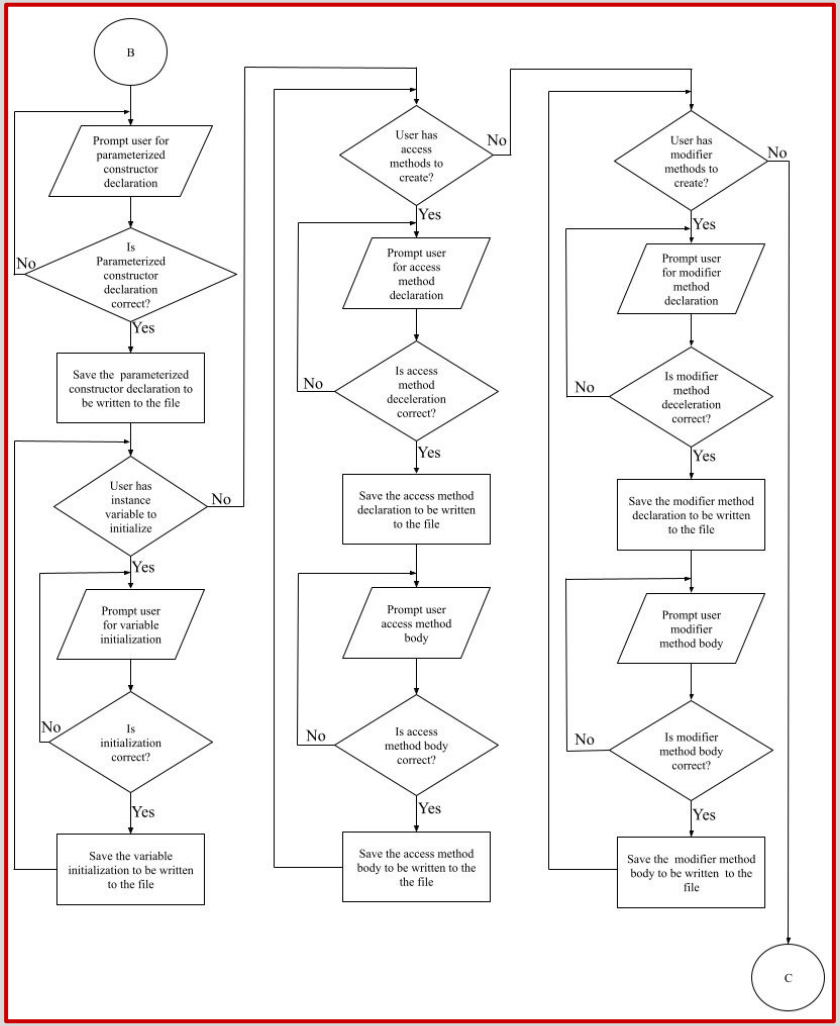
Algorithm

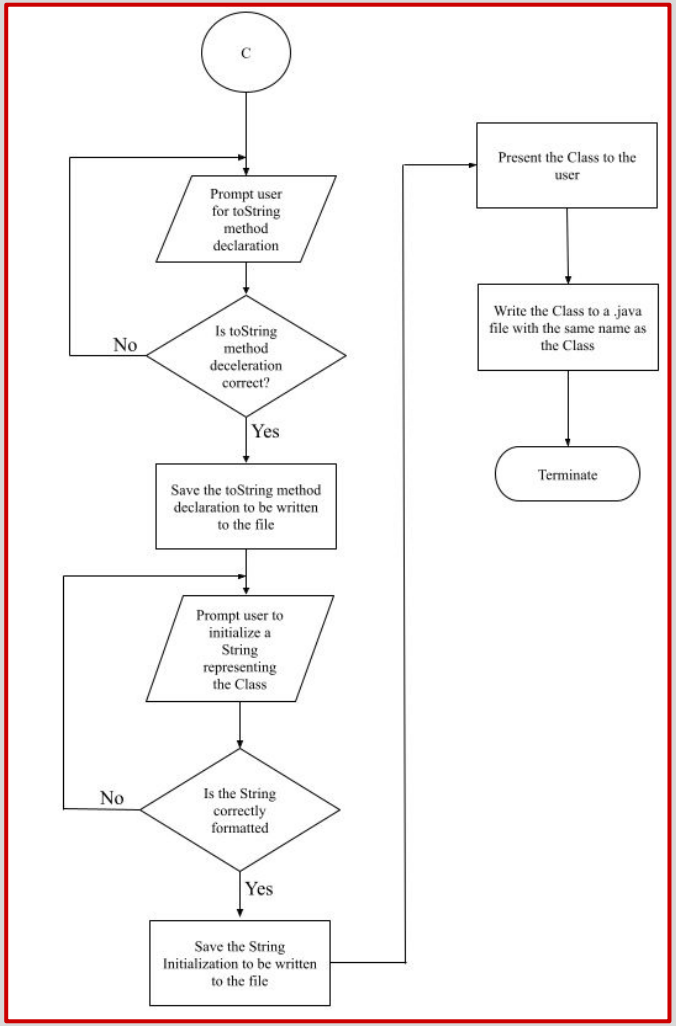
:


Start
Prompt user for Class name
Is Class name valid?
No
Yes
Ask user if Class extends another Class
Class extends another Class?
Yes
No
Prompt user for basic Class declaration
Is Class Declaration Correct?
No
Yes
Prompt user for name of the Class it is extending
Does Class exist?
No
Yes
Prompt user for Class declaration specifying the extension
Is Class Declaration Correct?
Yes
No
Save Class declaration to be written the file
A


A
User has instance variable to include
Yes
No
Prompt user for variable declaration and initialization (if applicable)
Is declaration / initialization correct?
No
Yes
Save the variable declaration / initialization to be written to the file
Ask user if Class should include default constructor
Class includes default Constructor?
Yes
No
Prompt user for default constructor declaration
Is default constructor declaration correct?
No
Yes
Save the default constructor declaration to be written to the file
Ask user if variables are initialized in default constructor
Instance variable initializations are included in the default constructor?
Yes
No
User has instance variable to initialize
No
Yes
Prompt user for variable initialization
Is initialization correct?
No
Yes
Save the variable initialization to be written to the file
B

Algorithm Continued:





Project Design Elements - Implemented Classes

ClassBuilder

-className: String
-extendedClassName: String
-classDeclaration: String
-unitializedInstanceVariables: ArrayList<InstanceVariable>
-directlyItializedInstanceVariables: ArrayList<InstanceVariable>
-instanceVariableHash: HashMap<String, InstanceVariable>
-allInstanceVariables: ArrayList<InstanceVariables>
-instanceVariableDeclarations: ArrayList<String>
-instanceVariableDirectInitializations: ArrayList<String>
-defaultConstructor: DefaultConstructor
-parameterizedConstructors: ArrayList<ParameterizedConstructor>
-setters: ArrayList<Setter>
-getters: ArrayList<Getters>
-objectsToString: ToString

+ClassBuilder(): ClassBuilder
+setClassName(in inputtedClassName: String) : boolean
+getClassName(): String
+setClassDecleration(in inputtedClassDecleration: String) : boolean
+setExtendedClassName(in inputtedExtendedClassName: String) : boolean
+setExtendedClassDeclaration(in inputtedExtendedClassDeclaration: String) : boolean
+instanceVariaableNameExists(in inputtedInstanceVariableName: String) : boolean
+addUnitializedInstanceVariable(in inputtedInstanceVariable: InstanceVariable) : boolean
+addInstanceVariableDeclaration(in inputtedInstanceVariableDeclaration: String, in inputtedInstanceVariable: InstanceVariable) : boolean
+addDirectlyInitializedInstanceVariable (in inputtedInstanceVariable: InstanceVariable) : void
+addInstanceVariableDirectInitialization(in inputtedDirectInitialization: String, in userInstanceVariable: InstanceVariable) : boolean
+getUninitializedVariables() : ArrayList<InstanceVariable>
+getAllInstanceVariables() : ArrayList<InstanceVariable>
+setDefaultConstructor(in inputtedDefaultConstructor: DefaultConstructor) : void
+setParameterizedConstructor(in inputtedParameterizedConstructor: ParameterizedConstructor) : void
+addSetter(in inputtedSetter: setter) : void
+addGetter(in inputtedGetter: getter) : void
+setToString(in toString: ToString) : void
+toString() : String

InstanceVariable

-type: String
-name: String
-value: String

+setType(in type: String) : boolean
+setName(in name: String) : boolean
+setValue(in value: String) : boolean
+getType() : String
+getName() : String
+getValue() : String

Project Design Elements - Implemented Classes Continued...

DefaultConstructor
-header: String -instanceVariableInitializations: String -superStatement: String
+DefaultConstructor() : DefaultConstructor +setDefaultConstructorHeader(in inputtedHeader: String, in className: String) : boolean +setSuperStatement(in inputtedSuperStatement: String) : boolean +addInstanceVariableInitialization(in inputtedInitialization: String, in instanceVariable: InstanceVariable) : boolean +toString() : String

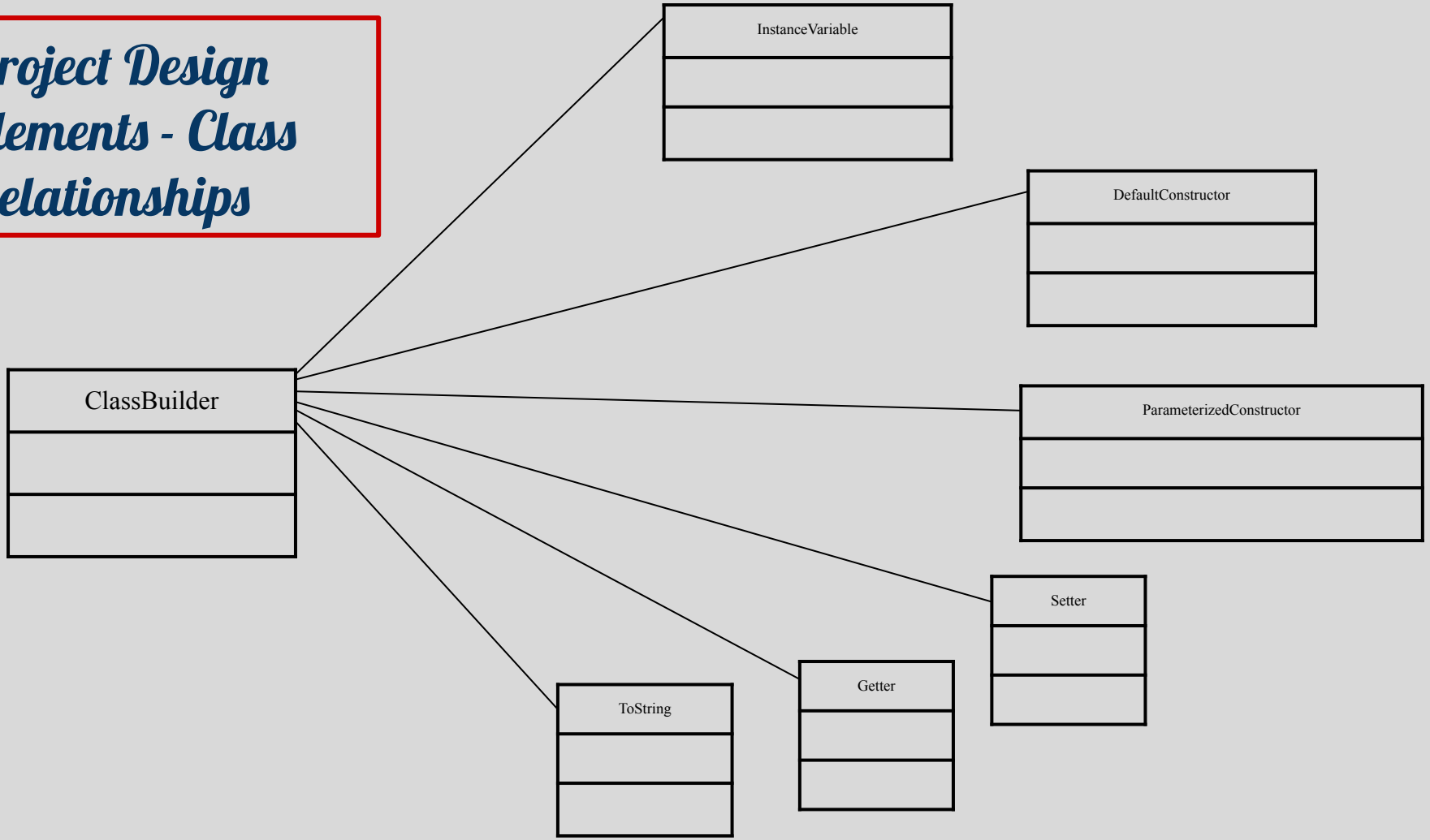
ParameterizedConstructor
-header: String -instanceVariableInitializations: String -superStatement: String
+ParameterizedConstructor() : ParameterizedConstructor +setParameterizedConstructorHeader(in inputtedHeader: String, in className: String, in parameterListInstanceVariables: ArrayList<InstanceVariable>) : boolean +setParameterizedConstructorHeaderForExtended(in inputtedHeader: String, in className: String, in parameterListInstanceVariables: ArrayList<InstanceVariable>) : boolean +setSuperStatement(in inputtedSuperStatement: String) : boolean +addConventionalInstanceVariableInitialization(in inputtedInitialization: String, in instanceVariable: InstanceVariable) : boolean +addFlexibleInstanceVariableInitialization(in inputtedInitialization: String, in instanceVariable: InstanceVariable) : boolean +toString() : String

Project Design Elements - Implemented Classes Continued...

DefaultConstructor
-header: String -instanceVariableInitializations: String -superStatement: String
+DefaultConstructor() : DefaultConstructor +setDefaultConstructorHeader(in inputtedHeader: String, in className: String) : boolean +setSuperStatement(in inputtedSuperStatement: String) : boolean +addInstanceVariableInitialization(in inputtedInitialization: String, in instanceVariable: InstanceVariable) : boolean +toString() : String

ParameterizedConstructor
-header: String -instanceVariableInitializations: String -superStatement: String
+ParameterizedConstructor() : ParameterizedConstructor +setParameterizedConstructorHeader(in inputtedHeader: String, in className: String, in parameterListInstanceVariables: ArrayList<InstanceVariable>) : boolean +setParameterizedConstructorHeaderForExtended(in inputtedHeader: String, in className: String, in parameterListInstanceVariables: ArrayList<InstanceVariable>) : boolean +setSuperStatement(in inputtedSuperStatement: String) : boolean +addConventionalInstanceVariableInitialization(in inputtedInitialization: String, in instanceVariable: InstanceVariable) : boolean +addFlexibleInstanceVariableInitialization(in inputtedInitialization: String, in instanceVariable: InstanceVariable) : boolean +toString() : String

Project Design Elements - Class Relationships



Project Evaluation

- Successfully assisted user in creating valid Java object classes.
  - Classes were properly instantiated
- Allowed for the creation of subclasses of objects created with the program.
  - As seen in the demo, “Trim” extended “Car”
- Forced the user to produce the code themselves while providing constant checking and validation for their input.
- Superiority over ChatGPT for this purpose.
  - The program constantly interacted with the user.
  - After constant testing with ChatGPT...
    - ChatGPT was very successful in created similar and correct classes.
    - Lacked support for building the class with the user, instead it provided steps then instantly provided the entire code.

Findings and Future Work

- Generative AI tools like ChatGPT are very powerful but this power can hinder a programmers ability to learn concepts for themselves.

- Although the proposed solution successfully met the initial goals, there are a few ways it can be made more effective...
  - Offer validation for elements related to super class when creating subclasses.

  - More validation checks for unconventional returns and assignments.
    - For example, providing analysis of operations being assigned or returned.

  - Allow users to input entire blocks of code for validation.
    - This will improve versatility in regards to skill level as it will allow the user to better test their skills as they become more familiar with the patterns and conventions.

References

1. ALMOG, G. (2019). Getting out of Hand? Examining the discourse of 'character amnesia.' Modern Asian Studies, 53(2), 690–717. https://doi.org/10.1017/S0026749X1700035X

2. Haindl, P., & Weinberger, G. (2024). Does ChatGPT Help Novice Programmers Write Better Code? Results From Static Code Analysis. IEEE Access, 12, 114146–114156. https://doi.org/10.1109/ACCESS.2024.3445432

3. Lindberg, R. S. N., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K‐12: A review of programming curricula in seven countries and programming games. British Journal of Educational Technology, 50(4), 1979–1995. https://doi.org/10.1111/bjet.12685