**BCPR301 – Advanced Programming**

**Assessment 4 Marking Sheet**

Student Name: Adam Sellars

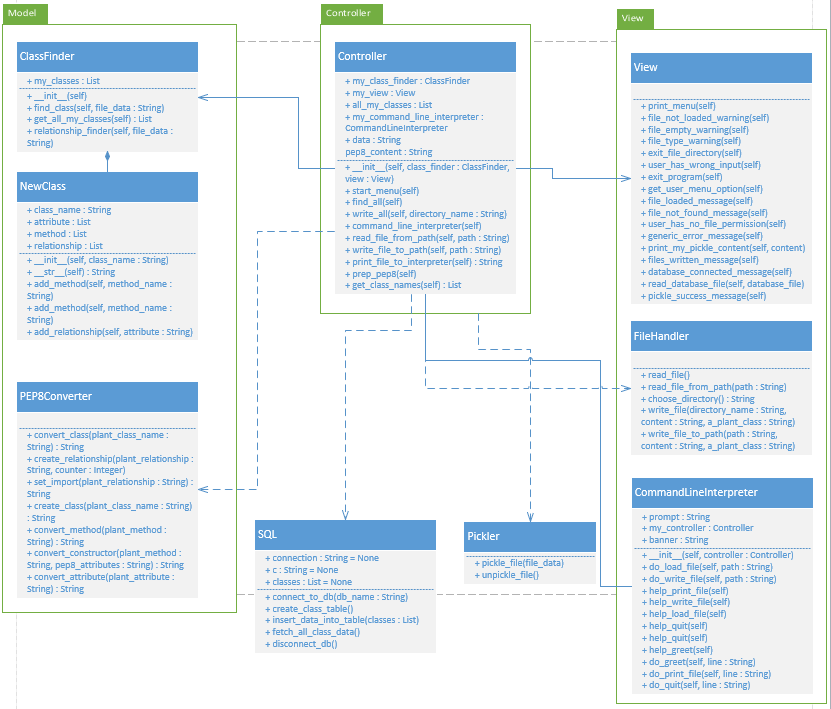
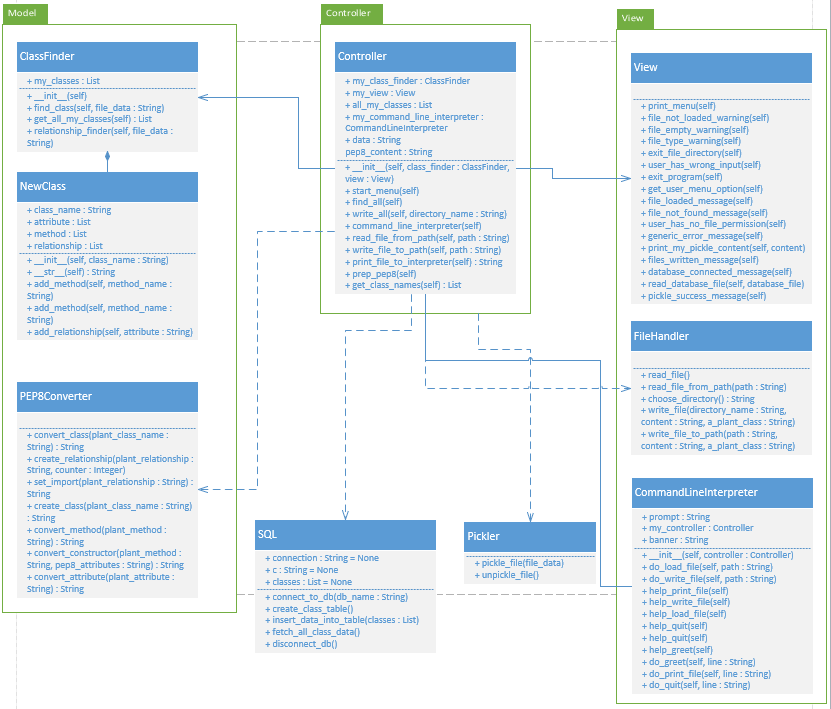
# The compulsory (i.e., ZERO mark if not being provided):

1. You MUST submit a filled self-marking sheet to indicate how many marks you think you can get for each section in the marking guide provided below.

# Marking guide (20 \* N marks in total where N = 2):

Builder

1. The class diagram before your modification (2 \* N marks)

The locations of code involved (i.e., code reference) **in your assignment 2 solution** (2 \* N marks)

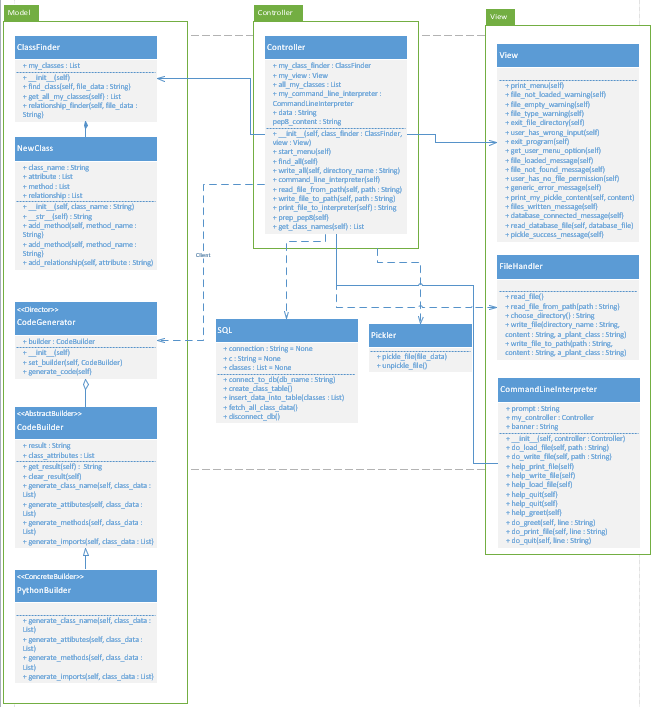
1. controller → interpreter\_controller → InterpreterController → lines 113 – 132
2. model → code\_generator → CodeGenerator → all code
3. model → code\_builder → CodeBuilder → all code
4. model → python\_builder → PythonBuilder → all code
5. The name of the design pattern applied (2 \* N marks)

Builder

1. The reasons why applying this design pattern is suitable; the reasons **MUST be specified** for the particular situation you try to apply, i.e., **do not just give general reasons** why using that design pattern is good. (2 \* N marks)

* Implementing builder made the code more extendable satisfying the open/closed SOLID principle. New Concrete builders can be easily added to extend the program.
* The Code before implementing Builder had the construction process controlled in the InterpreterController class which also had many other responsibilities, by encapsulating the construction process in the builder pattern the single responsibility process is being adhered to better.

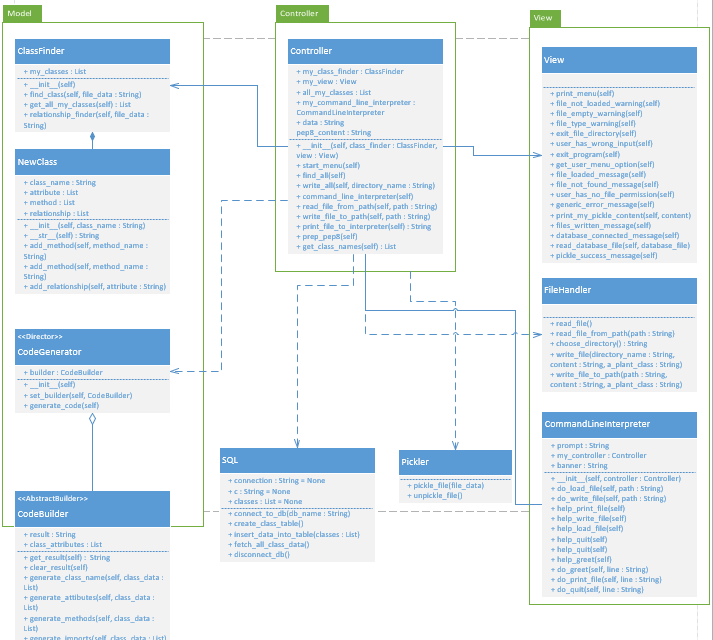
1. The class diagram after your modification; all the components in the design pattern class diagram provided in our textbook should be **explicitly labelled** in your class diagram. (2 \* N marks)



1. Applying the design pattern proposed. Your source code needs to pass PEP8 check (10 \* N marks)

Strategy

1. The class diagram before your modification (2 \* N marks)



1. The locations of code involved (i.e., code reference) **in your assignment 2 solution** (2 \* N marks)

view → command\_line\_interpreter → CommandLineInterpreter → lines 26 – 49

View → help\_display → HelpDisplay → all code

View → abstract\_help\_message → AbstractHelpMessage → all code

View → greet\_help → GreetHelp → all code

View → load\_file\_help → LoadFileHelp → all code

View → print\_file\_help → PrintFileHelp → all code

View → quit\_help → QuitHelp → all code

View → write\_file\_help → WriteFileHelp → all code

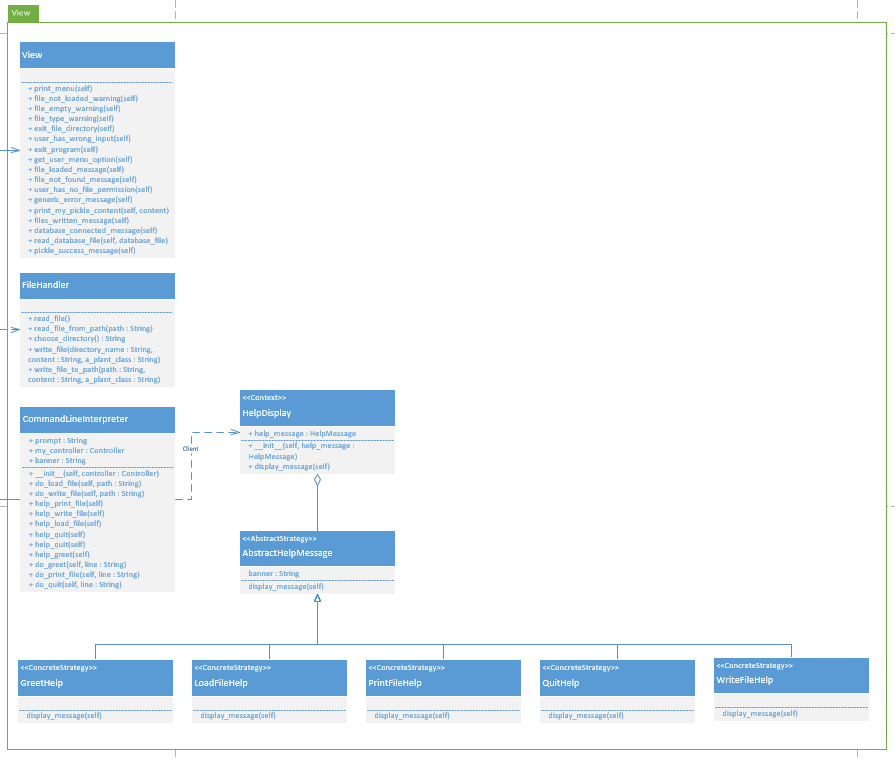
1. The name of the design pattern applied (2 \* N marks)

Strategy

1. The reasons why applying this design pattern is suitable; the reasons **MUST be specified** for the particular situation you try to apply, i.e., **do not just give general reasons** why using that design pattern is good. (2 \* N marks)

Original program had all Command line functionality in one class creating a large class with many responsibilities. The new program uses Strategy Design Pattern to separate out the specific functionalities of displaying command line help into a family of related classes. This adheres to the Single Responsibility Principle and helps prevent large class bad smell.

1. The class diagram after your modification; all the components in the design pattern class diagram provided in our textbook should be **explicitly labelled** in your class diagram. (2 \* N marks)



1. Applying the design pattern proposed. Your source code needs to pass PEP8 check (10 \* N marks)