Exercise 1: Hello with Date Time

A screen shot of a computer

Description automatically generated

I expect that the values would be:

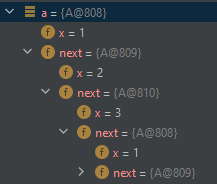
* 1
* 2
* 3
* 1

Result:

A screen shot of a computer

Description automatically generated

In debugger, when expanding the variable a, it will loop values in variable a,b,c as they reference each other.



Exercise 2: Sketching Object Graphs

A diagram of a graph

Description automatically generated

Exercise 3: Complete the Inefficient Implementations of LinkedList and ArrayList.

Contains method for LinkedList

A computer code with text

Description automatically generated

Contains method for ArrayList

A screen shot of a computer code

Description automatically generated

Exercise 4: Minimal rewrite of LinkedList and ArrayList to improve efficiency

Append function for EfficientLinkedList

A computer screen shot of code

Description automatically generatedA computer screen with text

Description automatically generated

Append function for EfficientArrayList

A computer screen shot of a code

Description automatically generated

Code extension of ListExample

A screen shot of a computer program

Description automatically generated

Output using an extension of ListExample

|  |  |  |  |
| --- | --- | --- | --- |
| LinkedList | EfficientLinkedList | ArrayList | EfficientArrayList |
|  |  |  |  |

Exercise 5: Performance Tests

Code used to test (slightly modified so I don’t have to change the code multiple times for a new number of elements)

A computer screen shot of a program code

Description automatically generated

A graph with numbers and lines

Description automatically generated

The average of EfficienctArrayList is constant while EfficientLinkedList for any case of append is constant. If you run the test multiple times, the graph below should look like 2 linear lines with the arraylist bumping up once in a while rather than random looking.

A graph with lines and numbers

Description automatically generated with medium confidence