

# EECS1022 Programming for Mobile Computing (Winter 2021)

Q&A - Lectures W9

Monday, March 22

# Prog-Test-4

- Review Session Thur/Fri. (M&N).
- Practice Test.
- OOP
  - ↳ 3 classes (closer to practice).
  - ↳ 2D arrays X
  - ↳ static attributes ✓

Variable shadowing occurs when a variable declared within a certain scope (e.g., method) has the same name as a variable declared in an outer scope (e.g., class).

The outer variable (at the class level) is said to be shadowed by the inner variable (at the method level), while the inner identifier is said to mask the outer identifier.

```
class Student {  
    private String name;  
    public void setName (String name) {  
        name  
    }  
}
```

Annotations:  
- attribute points to the class-level `name`.  
- is shadowed by points from the class-level `name` to the method-level `name`.  
- mask points from the method-level `name` to the class-level `name`.  
- parameter points to the method-level `name`.  
- wiki points to the is shadowed by text.

This question has been running over my mind for a long time now. I understand that when we program something a compiler will help us to compile and make that program work.

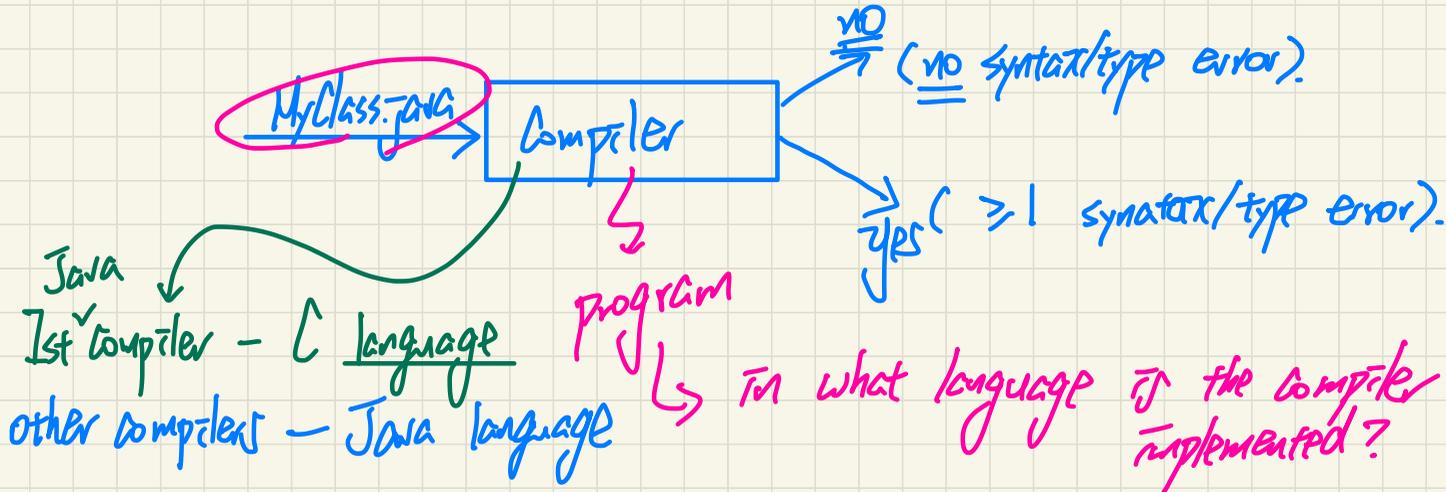
How about the compiler itself?

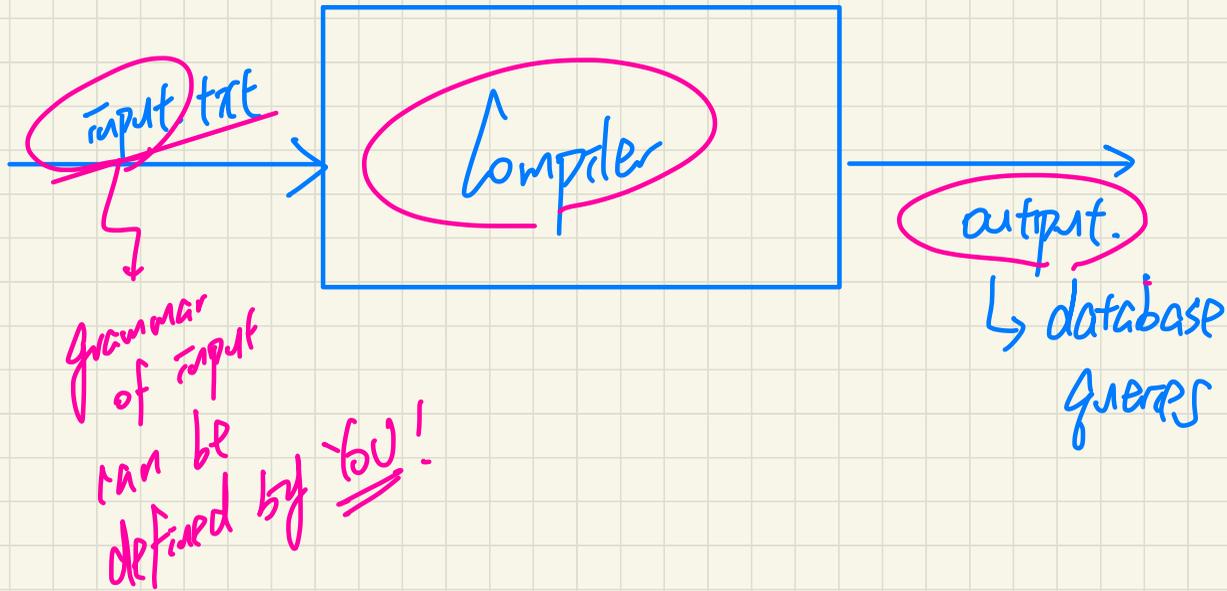
✓ EES 4302 Compilers &

If a compiler is being compiled by something else then what's the starting point of this?

Interpreter

How was the first programming language built?







Is it possible to create a global variable that's visible, accessible and modifiable across different classes, and if so, is it practical?

- You cannot declare a static variable outside all classes

```
public class A {  
    public static int i = 2;  
}
```

```
class B {  
    A.i.  
}
```

single copy  
A.i

