

Introduction to Metaprogramming (Tutorial)

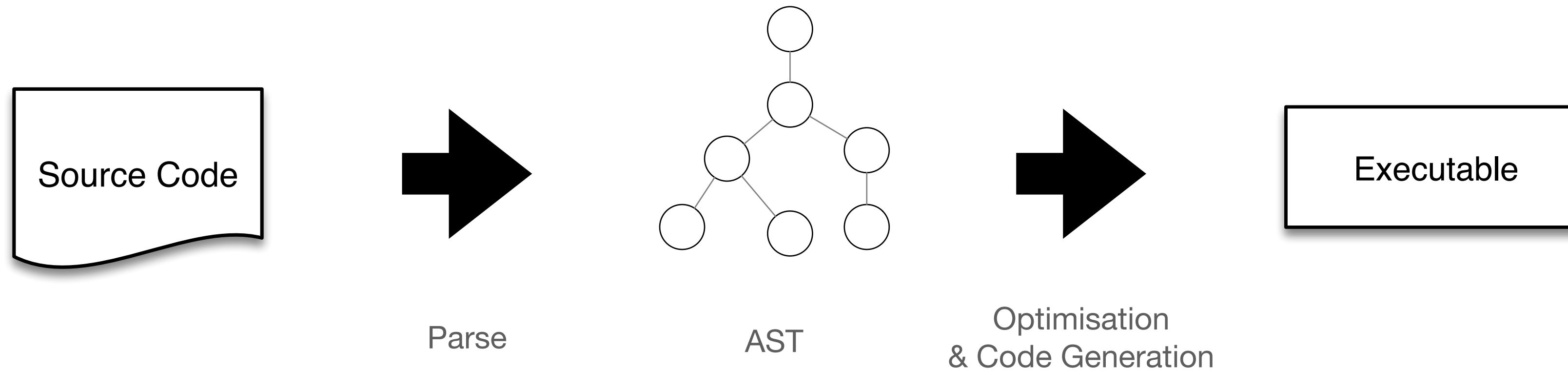
**CS453 Automated Software Testing / CS454 AI-Based Software
Engineering**

Shin Yoo

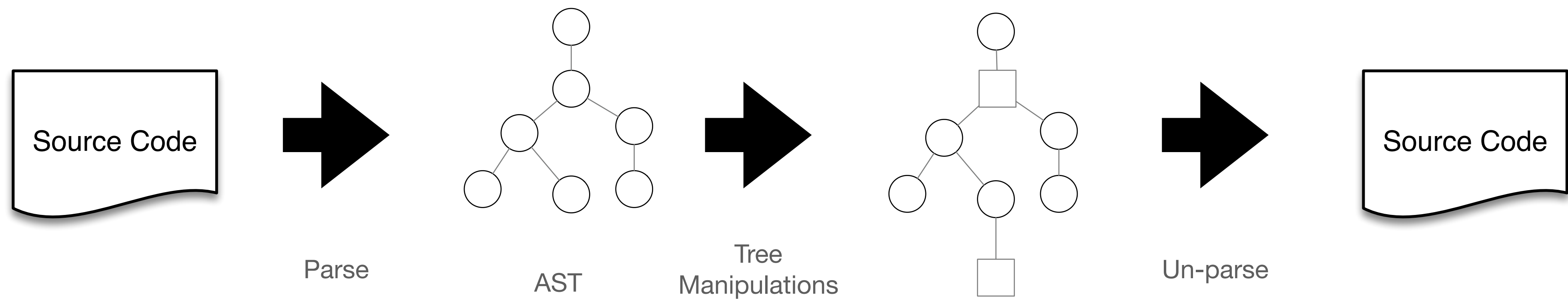
Metaprogramming

- Programming technique in which computer programs have the ability to take other programs as data: a program that can read, generate, analyze, or transform other programs (from Wikipedia)
 - A program may even modify itself!
- Von Neumann Architecture: a stored-program computer means 1) a separation of CPU and memory and 2) same mechanism being used to fetch both instructions and data. From this, we can derive two principles, code-as-data and data-as-code, opening our way to meta programming.

Manipulating AST in Python



Compilers



Metaprogramming

Tutorial Example 1

Programmatically modify from foo to bar

```
def foo(x):  
    if x > 0:  
        print('positive')  
    elif x == 0:  
        print('zero!')  
    else:  
        print('negative')
```

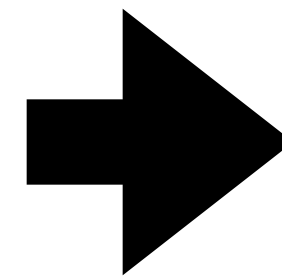
```
def change_function_name():  
    # what can we do here so that this program runs?
```

```
if __name__ == '__main__':  
    change_function_name()  
    bar(3) # DO NOT CHANGE
```

Tutorial Example 2

Programmatically add a function bar and write the source code

```
def foo(x):  
    if x > 0:  
        print('positive')  
    elif x == 0:  
        print('zero!')  
    else:  
        print('negative')
```



```
def bar():  
    print('Yeah!')  
  
def foo(x):  
    if x > 0:  
        print('positive')  
    elif x == 0:  
        print('zero!')  
    else:  
        print('negative')
```