

# 50.054 Name Analysis

ISTD, SUTD

## Learning Outcomes

1. Articulate What Name Analysis is
2. Differentiate Static and Dynamic variable scoping

## Name Analysis

For each identifier defined (i.e. name), we want to check

1. Is it a variable name or a function name?
2. Is the variable name of type int or bool?
3. What is the scope of the variable?
4. Has the variable been declared before used?
5. Where is the defined variable used?

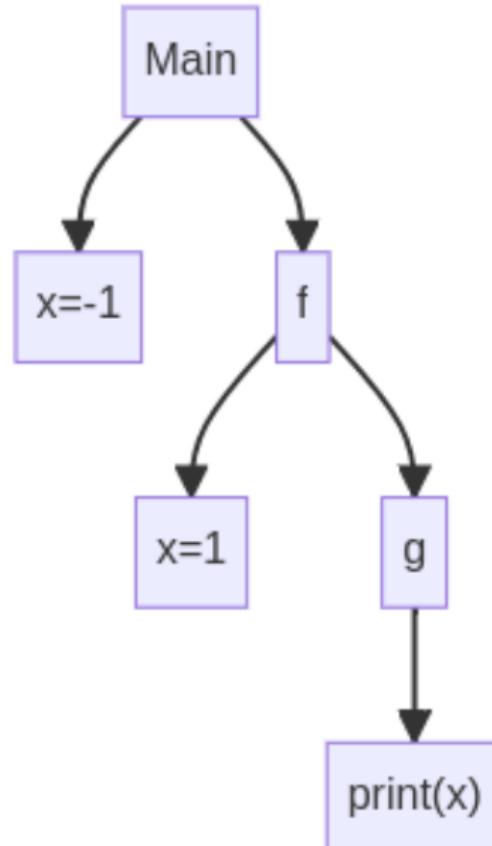
## Variable Scope

- ▶ Static Scope
- ▶ Dynamic Scope

# Dynamic Scoping

```
$x = -1;  
  
sub f {  
    local $x = 1;  
    return g();  
}  
  
sub g {  
    print $x;  
}  
f() # 1 is printed
```

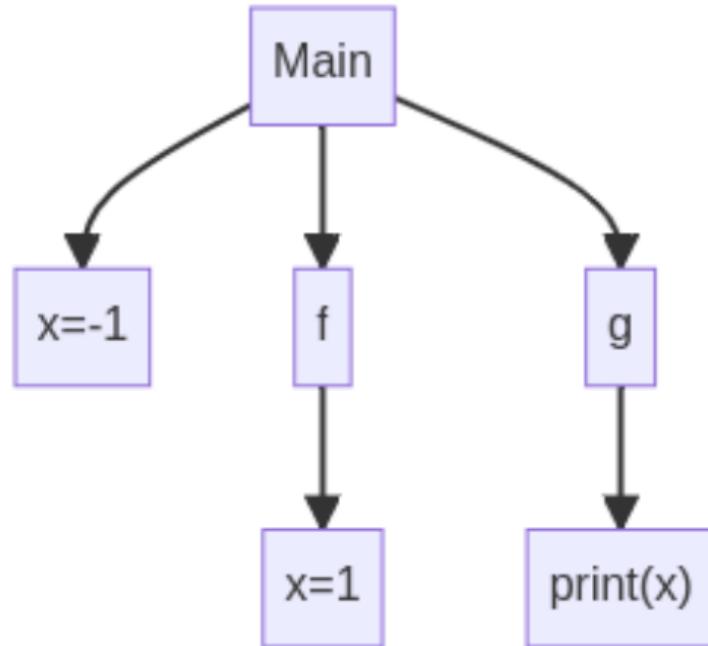
- ▶ variables are scoped according to the (run-time) call graph.



# Static Scoping in Python

```
x = -1  
  
def f():  
    x = 1  
    return g()  
  
def g():  
    print(x)  
  
f() # -1 is printed
```

- ▶ variables are scoped according to its declaration location.



## Static Scoping in Java vs Static Scoping in Python

```
class Main {  
    public static int main  
        (String[] argv) {  
        int x = 1;  
        if (argv.length > 0){  
            x = 2;  
        } else {  
            int y = 1;  
        }  
        println(y.toString());  
        return 1;  
    }  
}
```

```
def main(argv):  
    x = 1  
    if len(argv) == 0:  
        x = 2  
    else:  
        y = 1  
        print(y)
```

- ▶ All variables in the same function have the same scope.
- ▶ Run-time error
- ▶ Need more test cases (path-coverage testing)

- ▶ Variables from different nesting blocks have different scopes.
- ▶ Compilation error

## Back to SIMP

```
// SIMP_ERR1
x = 1;
if input == 0 {
    x = 2;
} else {
    y = 1;
}
return y; // y could be undefined.
```

- ▶ Want to detect error statically

## Analyse the Pseudo Assembly IR

```
// PA_ERR1
1: x <- 1
2: t <- input == 0
3: ifn t goto 6
4: x <- 2
5: goto 7
6: y <- 1
7: rret <- y // y could be undefined.
8: ret
```

# Quick Summary

- ▶ What is name analysis
  - ▶ What is the variable's reference
  - ▶ What is the scope of the variable
  - ▶ Where the variable is defined
  - ▶ Where the variable is used
- ▶ Static vs Dynamic Variable Scoping