Programming Paradigms

Names, Scopes, and Bindings (Part 2)

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart
Summer 2020
Overview

- Object lifetime and storage management
- Scopes
- Aliasing and overloading
- Binding of referencing environments
Object Lifetime

Every memory object has a lifetime

- Global variables: Entire program execution
- Local variables: Function execution

Object lifetime vs. binding lifetime

- A single object may be bound to multiple names
- Bindings may be concurrent
Example 1

fun f(x) {
    x = 3
    return x
}

y = f(x)
Example 2:

- program execution

- object

- binding

Usually a bug ("dangling reference")
Is use-after-free attack in C
Three kinds of memory objects

- **Static**
  - Absolute address retained throughout execution

- **Stack**
  - Usually within subroutines
  - Allocation/deallocation on call/return

- **Heap**
  - Allocation and deallocation at arbitrary times
Statically Allocated Memory

Depending on the PL, used, e.g., for

- Global variables
- Constant literals
- Symbol tables
- Program code itself
- Compile-time constants
  - Even if local to function
Stack-based Allocation

```
fun c() {
    ...
}

fun b() {
    if ...
    b()
    else
    c()
}

fun a() {
    b()
}

// main
a()
```
Heap-based Allocation

- For dynamically allocated data structures and objects whose size is statically unknown
  - E.g., objects in Java

- Some PLs: Managed memory
  - Unreachable objects: Implicitly deallocated
    - Unreachable = No active binding
  - Less control but fewer bugs
    - E.g., no use-after-free
class Person {
    int pid;
    String name;

    // constructor
}

public class Driver {
    public static void main(String[] args) {
        int id = 23;
        String pName = "John";
        Person p = null;
        p = new Person(id, pName);
    }
}

Please vote via Ilias.
Quiz: Memory Allocation

Where are the following data objects stored (Java)?

- The integer 23
- The string "John"
- The Person object
- The reference variable p

```java
class Person {
    int pid;
    String name;
    // constructor
}

public class Driver {
    public static void main(String[] args) {
        int id = 23;
        String pName = "John";
        Person p = null;
        p = new Person(id, pName);
    }
}
```

Please vote via ilias.
Quiz: Memory Allocation

Where are the following data objects stored (Java)?

- The integer 23
- The string "John"
- The Person object
- The reference variable p

Stack (in allocation frame of main)

Please vote via Ilias.
Quiz: Memory Allocation

Where are the following data objects stored (Java)?

- The integer 23
- The string "John"
- The Person object
- The reference variable \( p \)

```
class Person {
    int pid;
    String name;

    // constructor
}

public class Driver {
    public static void main(String[] args) {
        int id = 23;
        String pName = "John";
        Person p = null;
        p = new Person(id, pName);
    }
}
```

Please vote via Ilias.