

CS2030S Recitation

Week 8: Problem Set 5

brian

2026-03-12

National University of Singapore

Recap

Recap: Stack and heap

- Recall that the Stack contains frames for the active method calls
- heap contains the objects in memory
- There's a 3rd space called the *metaspace*
 - stores class level information
 - static fields

Recap: Variable capture

- variables will disappear when the stack frame is popped
- What if an object requires the value of that variable?
 - Capture the variable in the object instance
 - Something like a “hidden” field (not accessible by programmer)

Recap: Immutability

- Setting the stage for another paradigm
- A DS is mutable if it can be changed
- Immutable means that it cannot be changed
 - `String` is immutable
- In this course we relax this definition
 - Immutable objects must not have *observable* changes on the *outside*

Recap: Why make things immutable

- Easier to reason about
 - Guarantees that whatever you are referring to has not changed
- Sharing objects
 - Multiple objects can refer to something without worry about it changing
- Sharing internals
 - Possible to reuse some internals (see notes example on `ImmutableSeq`)
- Safer concurrency
 - Guarantees would still hold even if different interleaving of instructions (not important now, will learn in the future)

Recap: Fully qualified name

- Consider the following code

```
1 class B{
2     int x = 1;
3
4     class A {
5         int x = 0;
6
7         int f() {
8             int x = 3;
9             return x; // which x?
10        }
11    }
12 }
```

- What is `x` referring to?
- Somewhat ambiguous, esp from the compilers pov
- Fully qualify the name to remove ambiguity

Recap: Fully qualified name

- Consider the following code

```
1 class B{
2     int x = 1;
3
4     class A {
5         int x = 0;
6
7         int f() {
8             int x = 3;
9             return B.this.x; // oh this x
10        }
11    }
12 }
```

- What is `x` referring to?
- Somewhat ambiguous, esp from the compilers pov
- We can fully qualify the name to remove ambiguity

Recap: Fully qualified name

- Idea:
 - ▶ If it's a field add `this`
 - ▶ If it's some outer class, add the class name e.g. `B`
 - ▶ we can chain these 2 e.g. `B.this` to access outer class `B`'s fields

Q1:

We have the following code

```
1 B b = new B();  
2 B.f();
```

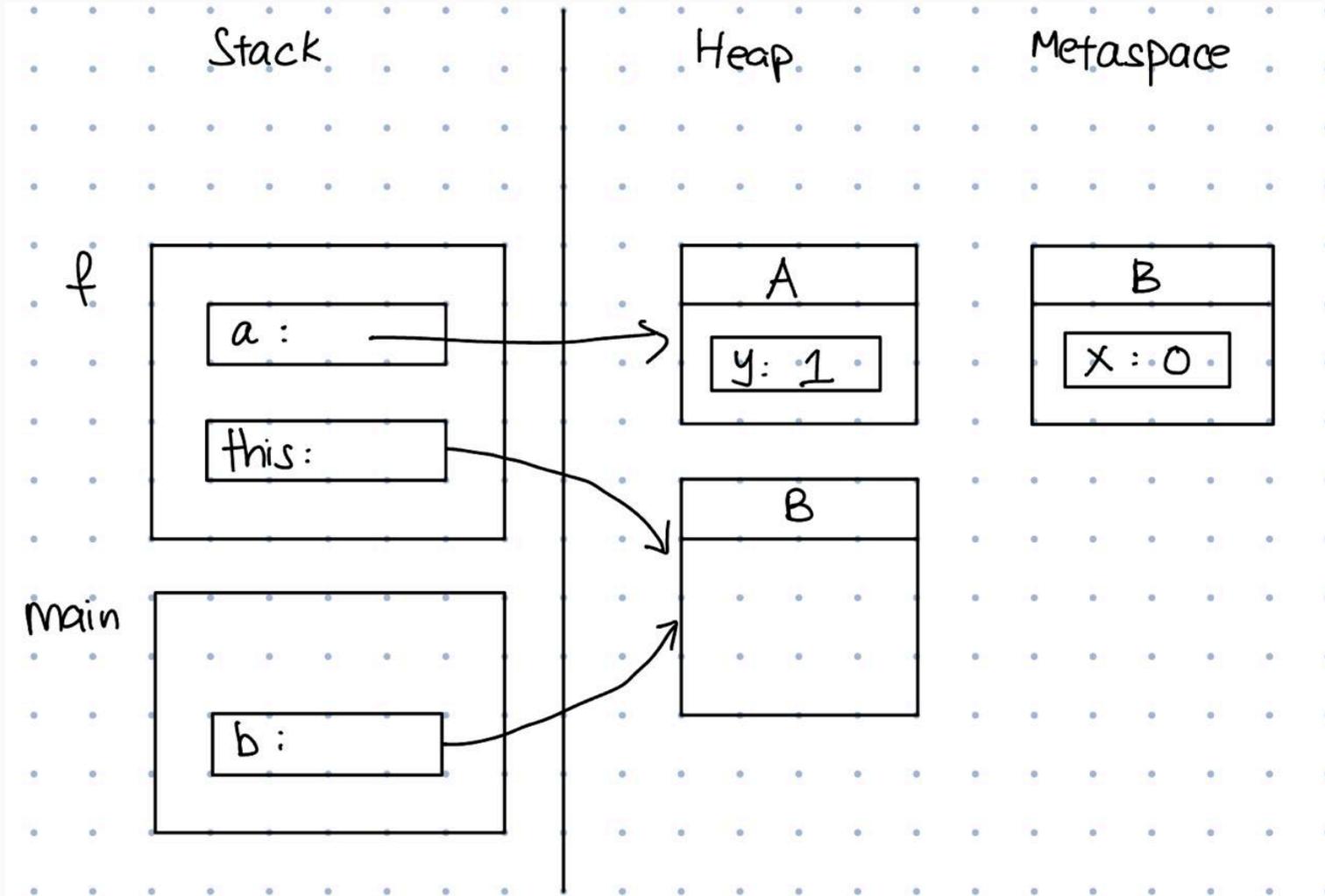
Q1a: Fully qualify and draw stack and heap

```
1 class B {
2     static int x = 0;
3     void f() {
4         A a = new A();
5         // Line A
6     }
7     static class A {
8         int y = 0;
9         A () {
10            y = x + 1;
11        }
12    }
13 }
```

Q1a: Fully qualify and draw stack and heap

```
1 class B {
2     static int x = 0;
3     void f() {
4         A a = new A();
5         // Line A
6     }
7     static class A {
8         int y = 0;
9         A () {
10            this.y = B.x + 1;
11        }
12    }
13 }
```

Q1a: Fully qualify and draw stack and heap



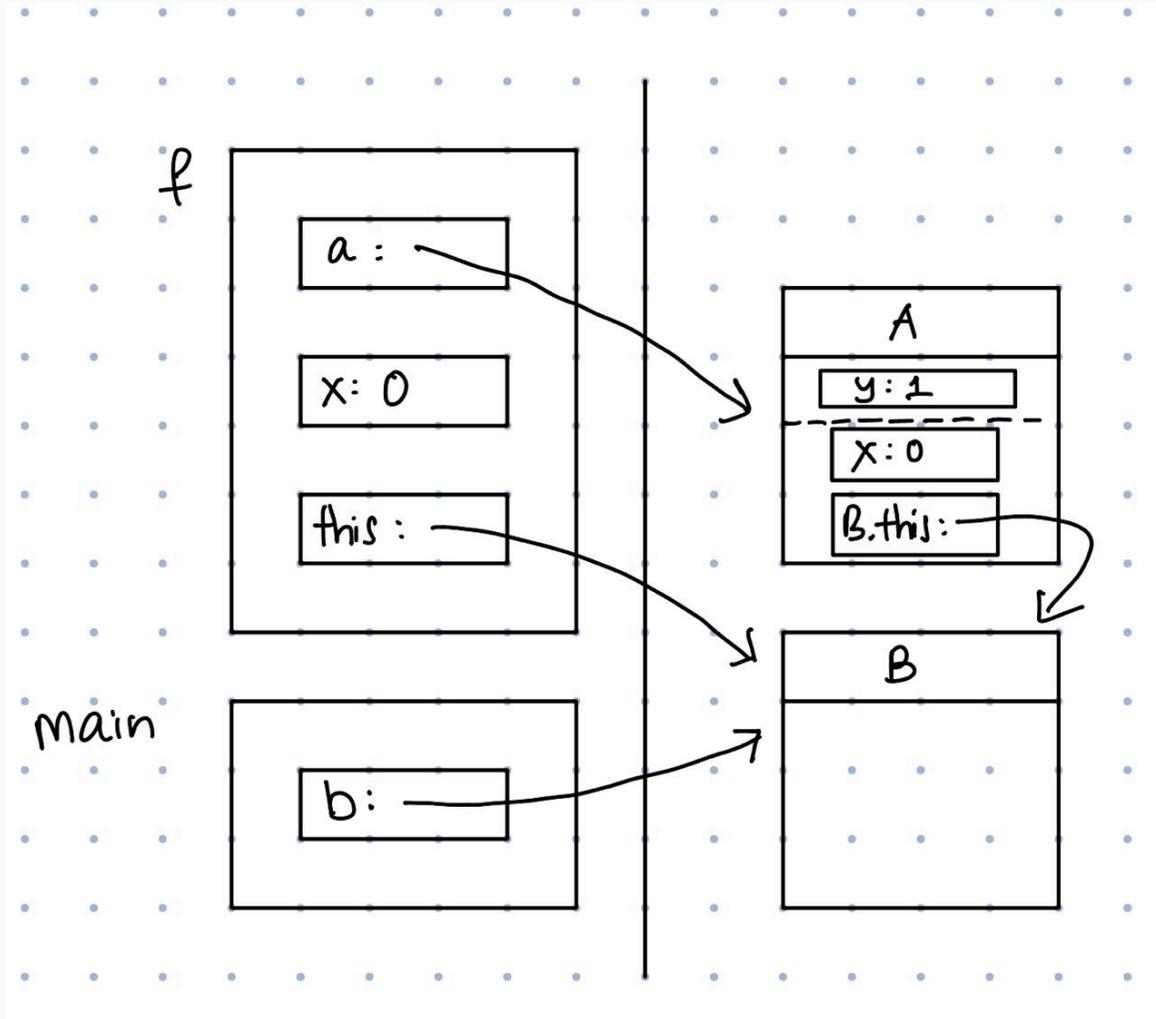
Q1b: Fully qualify and draw stack and heap

```
1 class B {
2     void f() {
3         int x = 0;
4         class A {
5             int y = 0;
6             A() {
7                 y = x + 1;
8             }
9         }
10        A a = new A();
11        // Line A
12    }
13 }
```

Q1b: Fully qualify and draw stack and heap

```
1 class B {
2     void f() {
3         int x = 0;
4         class A {
5             int y = 0;
6             A() {
7                 this.y = x + 1;
8             }
9         }
10        A a = new A();
11        // Line A
12    }
13 }
```

Q1b: Fully qualify and draw stack and heap



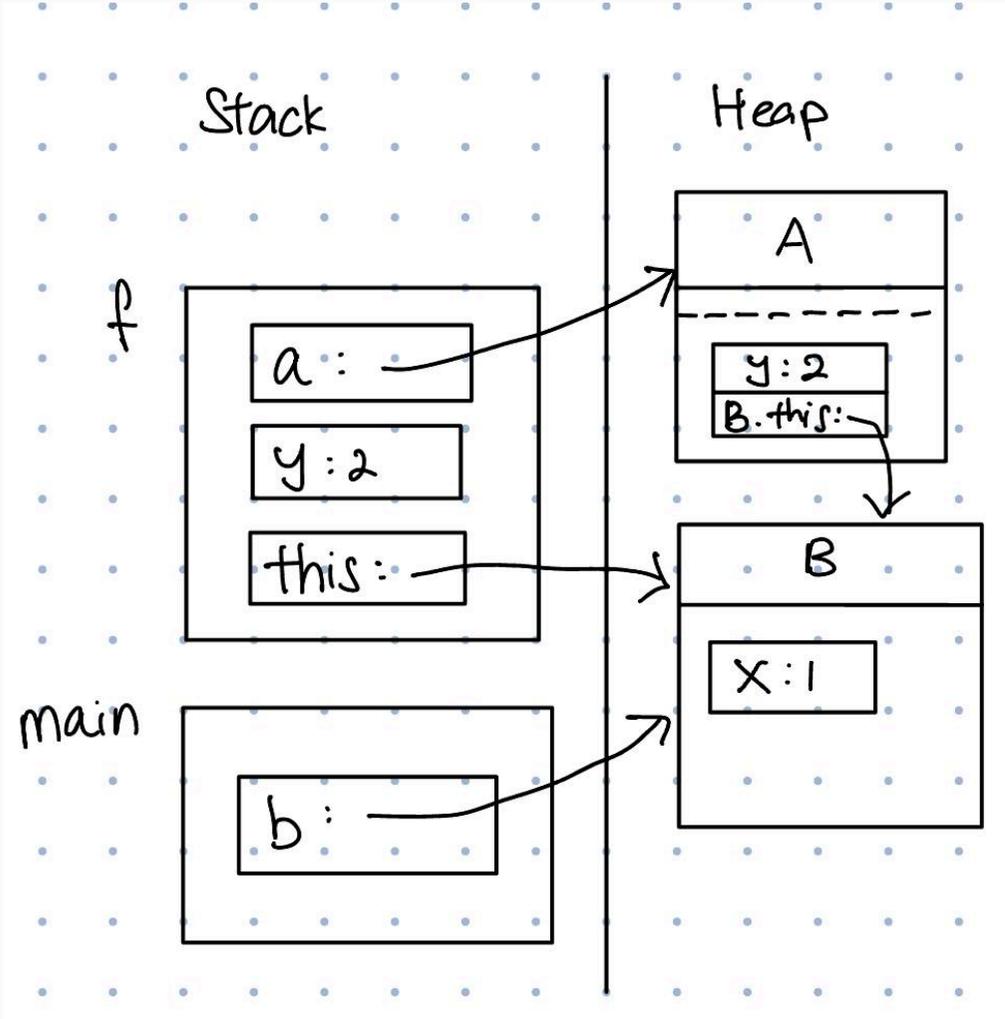
Q1c: Qualify and draw stack and heap

```
1 class B {
2     int x = 1;
3     void f() {
4         int y = 2;
5         class A {
6             void g() {
7                 x = y;
8             }
9         }
10        A a = new A();
11        // Line A
12        a.g();
13    }
14 }
```

Q1c: Qualify and draw stack and heap

```
1 class B {
2     int x = 1;
3     void f() {
4         int y = 2;
5         class A {
6             void g() {
7                 B.this.x = y;
8             }
9         }
10        A a = new A();
11        // Line A
12        a.g();
13    }
14 }
```

Q1c: Qualify and draw stack and heap



- General steps to make a class immutable
 - Make class final
 - Make fields final
 - any mutator methods should create a new instance instead

The End

bye!