CMSC131 Lab

Sections 0108 or 0109

Tim Lin

tlin@cs.umd.edu

OH: Tue. 6-7pm

Written Assignment 2

COVERED

Question 1-2

Which Java component creates and which one initializes the object created when executing `new Cat(“Fluffy”)`?

```
public class Cat {
    public Cat() {
        /* initialization */
    }
    public Cat(String name) {
        /* initialization */
    }
}
```

When can an object reside in the stack?

COVERED

Questions 3-5

When drawing a memory map we have frames. Describe what a frame is, its contents, and duration.

When should a constant be defined `static final` vs. just `final`?

Describe the differences between an instance variable and a local variable. When each of the them is created? When each of them is destroyed?

COVERED

Questions 6-8

How can we avoid code duplication when defining constructors for a class?

What is null? What does it represent?

When can the current object reference be null?

COVERED

Question 9

How many constructors are there for the left Dog class?

How many constructors are there for the right Dog class?

```java
public class Dog {
    private String name;
}
```

```java
public class Dog {
    private String name;
    public Dog(String name) {
        this.name = name;
    }
}
```

COVERED

More Memory Maps!

COVERED

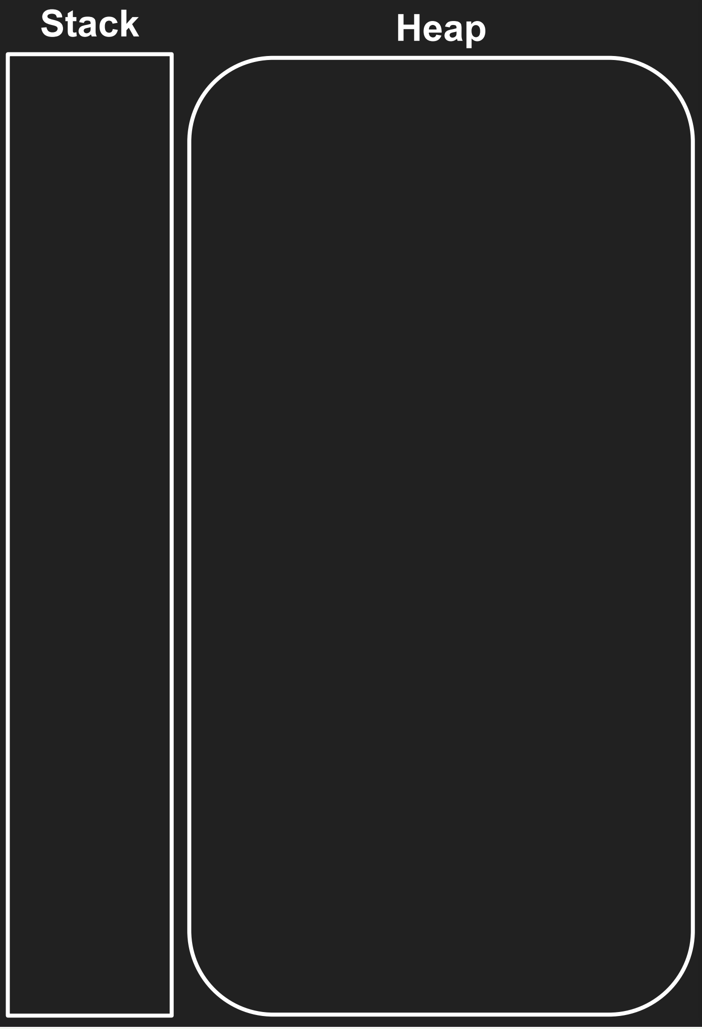
Stack

Static Area

```
public class Driver {
    public static void increaseBy(int[] data, int delta) {
        for (int i = 0; i < data.length; i++) {
            data[i] += delta;
        }
        /* HERE */
    }
    public static void main(String[] args) {
        int[] src = { 10, 30, 40 };
        increaseBy(src, 5);
        for (int i = 0; i < src.length; i++) {
            System.out.println(src[i]);
        }
    }
}
```

Heap

COVERED



Stack

Heap

```
public class Cat {
    private String name;
    private int lives;
    public Cat(String name) {
        this.name = name;
        this.lives = 7;
    }
    public void decreaseLives() {
        lives--;
    }
    public String toString() {
        return “Cat [name=” + name + “, lives=” + lives + “]”;
    }
}
public class Driver {
    public static void decreaseLives(Cat[] all) {
        for (int i = 0; i < all.length; i++) {
            all[i].decreaseLives();
        }
        /* HERE */
    }
    public static void main(String[] args) {
        Cat[] allCats = new Cat[2];
        allCats[0] = new Cat(“Garfield”);
        allCats[1] = new Cat(“Fluffy”);
        decreaseLives(allCats);
        for (int i = 0; i < allCats.length; i++) {
            System.out.println(allCats[i]);
        }
    }
}
```

COVERED

Office Hours