

CSCI 1103: Finale

Chris Kauffman

*Last Updated:
Wed Dec 13 11:03:52 CST 2017*

Logistics

Date	Lecture	Outside
Mon 12/11	Recursion	Lab 14: Review
Wed 12/13	Review	P5 Due
Wed 12/20	Final Exam	1:30pm-3:30pm KELLER HALL 3-210

Goals: Review

- ▶ Wrap-up project
- ▶ Move on to the final exam review

Project 5

- ▶ Due Tonight
- ▶ Accepted as late until Friday
- ▶ Note on academic integrity
- ▶ Questions?

Final Exam

Logistics

- ▶ Wed 12/20 1:30-3:30pm
- ▶ Keller Hall 3-210 (EAST BANK)
- ▶ Comprehensive: combination of coding, analysis, short answer, full semester material may be examined
- ▶ Length: 5-6 sides of paper
 - ▶ Midterms were 3 sides of paper
- ▶ No bluebook or bubble sheet required
- ▶ Open Resource as were the midterm exams

Topics Request

Any particular topics folks would like to discuss prior to review questions?

Review Question 1: Pet Adoption

A pet adoption agency which needs to keep records of dogs ready for adoption. The information needed is as follows:

- ▶ Name of the dog
- ▶ Age
- ▶ Whether it like cats or not

Describe a simple Dog class which has a constructor and private fields for this task.

Review Question 2: Dog Methods

Given the below Dog class, fill in bodies for the methods that need definitions.

```
public class Dog{
    private String name;
    private int age;
    private boolean likesCats;

    public Dog(String name, int age,
               boolean likesCats)
    {
        this.name=name;
        this.age=age;
        this.likesCats=likesCats;
    }

    // Retrieve the name of the dog
    public String getName(){
        ???
    }

    // Change the name of the dog to
    // the new name
    public void rename(String name){
        ???
    }

    // Increase the age of the dog by 1
    public void birthday(){
        ???
    }

    // Change the internal state of the
    // dog so that it likes cats
    public void trainForCats(){
        ???
    }
}
```

Review Question 3: Dog Creation

Add the following method to the dog class which creates a dog based on a string of information.

```
public class Dog{
    private String name; private int age; private boolean likesCats;

    public Dog(String name, int age, boolean likesCats) {
        this.name=name;
        this.age=age;
        this.likesCats=likesCats;
    }

    // Create a dog from a string of information. Use a Scanner to parse
    // the string argument. Valid input strings contain the Dog's name,
    // age, and the word yes or no depending on whether the dog likes
    // cats. Examples:
    //   Dog d = Dog.fromString("Val 6 yes");
    //   Dog e = Dog.fromString("Stout 3 no");
    public static Dog fromString(String s){
        ??
    }
}
```

Review Question 4: Dogs in Files

Add the following method to the dog class which creates an array of dogs from the named file.

```
public class Dog{
    public static Dog [] readDogsFromFile(String filename) throws Exception
    // Read dogs from the given
    // file. Count lines in the file
    // using a Scanner (do not assume a
    // countLines method). Allocate an
    // array of that size, reset the
    // scanner to beginning of the
    // file, then read lines and create
    // dogs in the array. Make use of
    // the Dog.fromString() method.
```

SAMPLE FILE:

Val	6	yes
Stout	3	no
Ein	4	yes
Kudjo	9	no
Balto	7	yes
Amaterasu	100	yes
Bo	9	no