CSE 113 – Spring 2010 Exam 4 Review Sheet

What is overloading – specifically method overloading?

Method overloading is having two methods that have the same name in a class. These methods must differ in either number or type of parameters. We saw this with multiple constructors for the same class most often.

Notes: Arrays are fixed size and are indexed by integer numbers. Arrays contains elements of the same type.

Declare a variable that refers to an array that contains Actor objects.

Actor[] actors;

GENERAL FORM: TypeOfThingInArray[] nameOfArrayVariable;

Create the array to hold a maximum of 20 actors and assign it to the variable you created in the previous part.

actors = new Actor[20];

GENERAL FORM: nameOfArrayVariable = new TypeOfThingInArray[MaximumNumberOfElements];

Put a new Actor object into the array at index 10.

actors[10] = new Actor();

GENERAL FORM: nameOfArrayVariable[index] = valueToBeInserted;

Write the code that would retrieve object at index 4 of the array.
actors[4]

GENERAL FORM: nameOfArrayVariable[index]

What is the first index for an array? It is ALWAYS zero (0).

What is the last index for an array? It is the size of the array minus 1.

Notes: While loops are used for repeating (like for-loop), except they are classified as indefinite loops. It loops until the condition on the loop is false.

Write a while loop that continues to execute while the variable x is true, but stops when x is false. Every time the body of the loop is executed, you should create a new Ball object.

SYNTAX:

```
while(condition) {

   //code to execute repeatedly

}
```

ANSWER:

```
while(x == true) {

      new Ball();

}
```

Below is a class for an Actor subclass named Dancer. You should program the Dancer class to keep track of how many steps it has danced. When it has danced 45 steps, the Dancer object should reset its count back to zero and set the Dancer object's rotation to be 45 more to the right than it was previously.

```
public class Dancer extends Actor {

        private int steps = 0;

        public void act() {

                steps = steps + 1;

                if(steps >= 45) {

                        steps = 0;

                        setRotation(getRotation() + 45);

                }

        }

}
```

There may also be some similar questions to the code-writing questions from Exam 3 (using if/else and loops).