CSE205 Exam#1 Study Guide

Covers chapters ~9, the respective exercises and assignments 1-5. 40% of code writing questions and 60-70% of multiple choices for a total of 110 points, including 10 points bonus. Programming exams are inherently cumulative in nature, so you should be familiar with everything we’ve done to date in the course. **You will need a solid understanding of Decisions, Loops, Arrays, ArrayLists, Inheritance & Interfaces to make a high score on this exam!**

Anything that you’ve done in lab or in an assignment, you should be able to do on the exam...here are some examples

**Comparing numbers, strings**
Type conversions, string concatenations & splits
Boolean type, values, predicate methods
Document the program
Understand if, nested if, if-else
Understand For loop, While loop, Do-While loop, Convert 1 kind of loop to another kind of loop
Trace a loop, Terminate a loop
Syntax of Array, ArrayList
Iterate the collection by using For-Each loop
The unique role of the Object class
Can interfaces be instantiated? Why or why not?
Overriding inherited methods
Software development models & UML
Classes from Java standard library: String, Rectangle, Object

I will give you skeletal code for some classes in an inheritance hierarchy with some interfaces like the following. I will then declare some variables and make some assignments and you will have to understand which statements are legal, and which actual methods are being invoked. I **strongly** recommend you create a new project and implement these classes with each method simply displaying a diagnostic message to the console. In a tester class, experiment with assignments of various object types to several different types of variables and see what is displayed. It should only take a few minutes to do this and it is essential to answering these questions correctly:

```java
public interface Movable{ void move();}

public abstract class Animal implements Movable{
    private String name;
    public abstract eat();
    public void move(){System.out.println("animal moves");}
    public void sleep(){..}
}

public class Mammal extends Animal{
    public void eat(){System.out.println("gimme meat!");}
    public void regulateTemperature(){..}
}

public class Human extends Mammal{
    public void think(){..}
    public void move(){System.out.println("human walks");}
}
```