

**COMP 110/L**  
**Fall 2022**

**Lecture 19 Handout**

1.) Consider the following class, which defines a rectangle:

```
public class Rectangle {
    private int length;
    private int width;
    public Rectangle(int length, int width) {
        this.length = length;
        this.width = width;
    }
    public int getArea() {
        return length * width;
    }
}
```

**Define a Square class, which will represent a square, but based on Rectangle. Square's constructor takes the length of a side. You should not need to define the getArea method or any instance variables on Square. Your Square class should work correctly with the following code.**

```
Rectangle shape = new Square(3);
int area = shape.getArea();
System.out.println(area); // prints 9
```

2.) Consider the `Animal` class below, which defines a `speak` method:

```
public class Animal {
    public void speak() {
        System.out.println("This animal speaks");
    }
}
```

Define classes below to make the following code compile and produce the correct output. As a hint, you'll need to extend the `Animal` class for every kind of animal you have, and each subclass of `Animal` will need to override the `speak` method.

```
Animal cat = new Cat();
cat.speak(); // prints "meow"
Animal dog = new Dog();
dog.speak(); // prints "woof"
```

3.) Consider the following code:

```
public class Example {  
    public static void main(String[] args) {  
        BaseClass object = new ChildClass(true);  
        String result = object.someMethod(7);  
    }  
}
```

Without modifying any existing code, define any classes and methods necessary to get this to compile. You do not need to worry about what the code should do, only what you need to get this to compile.

4.) The code below has some issues that prevent it from compiling. Circle all the lines in Altogether's main method that will cause a compile-time error.

```
public class SomeClass {
    public void method1() {
        System.out.println("method1");
    }
}

public class SomeOtherClass extends SomeClass {
    public void method2() {
        System.out.println("method2");
    }
}

public class Altogether {
    public static void main(String[] args) {

        SomeClass c1 = new SomeClass();

        c1.method1();

        c1.method2();

        SomeClass c2 = new SomeOtherClass();

        c2.method1();

        c2.method2();

        SomeOtherClass c3 = new SomeClass();

        c3.method1();

        c3.method2();

        SomeOtherClass c4 = new SomeOtherClass();

        c4.method1();

        c4.method2();
    }
}
```