Midterm 2 Study Guide

CSC 210 Fall 2024

In addition to the weekly practice problems, this guide is to help you prepare for other types of questions.

Code reading and Polymorphism

Given the following code, answer the questions below.

```
public class SimpleTime {
    private int hour;
    private int minute;

    public SimpleTime() {
        System.out.println(toString());
    }

    public SimpleTime(int h, int m) {
        hour = h;
        minute = m;
        System.out.println(toString());
    }

    public void setTime(int h, int m) {
        hour = h;
        minute = m;
    }

    public String toString() {
        return hour + ":" + minute;
    }
}
```

a) Does the class `SimpleTime` make use of method overloading? If so, how?

b) Does the class `SimpleTime` make use of method overriding? If so, how?

c) What is the output for when the code below executes?

```java
public class SetSimpleTime {

    public static void main(String[] args) {
        SimpleTime myTimeOne = new SimpleTime();
        SimpleTime myTimeTwo = new SimpleTime(10, 50);
        myTimeOne.setTime(11, 11);
        System.out.println(myTimeOne.toString());

    }

}
```

Inheritance

The Mayor's Office of Tucson wants a system to keep track of all buildings in the city and to calculate its tax revenue from them. There are residential buildings, retail buildings and industrial buildings. Each building has a monthly tax rate which is the same for all buildings of a specific type. The monthly tax is calculated by multiplying the tax rate by the square foot area of the building. The system needs to keep a list of all buildings and calculate the total tax revenue each month.

a) Draw a UML class diagram for the system. Your system should have an inheritance hierarchy.

b) Write Java code for the super class and one subclass in the inheritance hierarchy. Keep it simple but show the code to calculate the tax.

c) Write Java code to show how the system uses polymorphism to calculate the total tax revenue.

ANSWERS

Code reading and Polymorphism

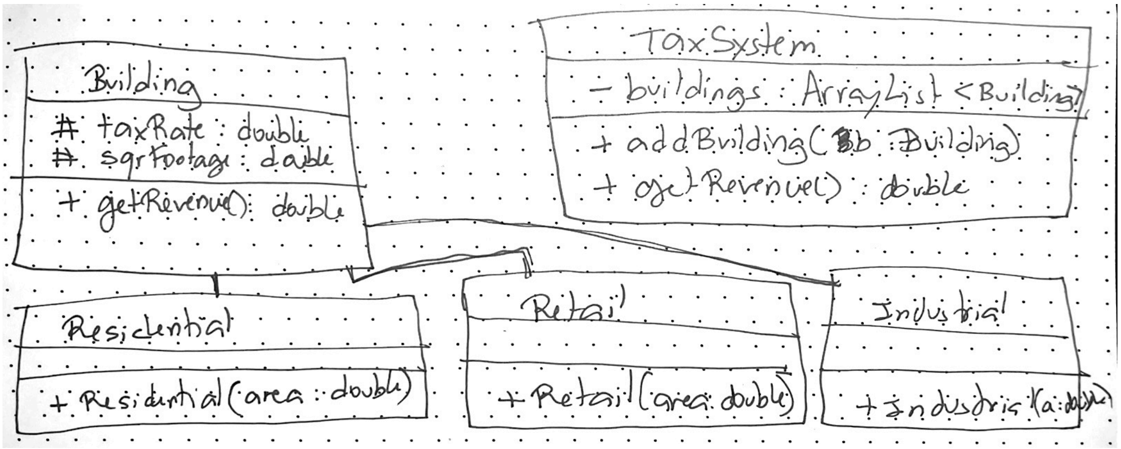
a) Yes, the class `SimpleTime` makes use of method overloading in the two different constructors – one takes no arguments, the other takes two int arguments.

b) Yes, the class `SimpleTime` makes use of method overriding – it overrides the `toString()` method. All classes inherit from the `Object` class, which has a `toString()` method. The `toString()` method in `SimpleTime` overrides that.

c)

```
0:0
10:50
11:11
```

Inheritance

a) 

b)

```
public class Building {
    protected double taxRate;
    protected double squareFootage;

    public double getRevenue() {
        return taxRate * squareFootage;
    }
}
```

```
public class Residential extends Building {

    public Residential (double area) {
        squareFootage = area;
        taxRate = 0.1;
    }
}
```

c) Implement two getRevenue in the class that makes uses of the Building class (with an array of Building). One of the getRevenue gets all revenue, another getRevenue method gets revenue by building type. (no need to show the entire code, here I'm calling three different methods that are not specified in my answer).

```
public double getRevenue() {
        double total = 0;
        for (Building b : buildings) {
            total += b.getRevenue();
        }
        return total;
    }

    public double getRevenue(String type) {
        switch (type) {
        case "residential":
            return getRetRev();
        case "industrial":
            return getIndRev();
        default: // default is residential
            return getResRev();
        }
    }
```