

*Score/70*

*Answer Key*

CMPS 12A Winter 2009

Name: \_\_\_\_\_

QUIZ 4: March 6

Email: \_\_\_\_\_

1) (5 points) The technique of selecting a method at execution time based on the most specific type of the object through which the method was invoked is called (choose one):

- a) Isomorphism
- b) Polyandrisim
- c) Polymorphism
- d) Ethnocentrism
- e) Malapropism

*C*

2) (5 points) In which of the following pairs would it be appropriate to define the second item in the pair as a subclass of the first item? (List all the letters that apply.)

- a) (Vehicle, Truck)
- b) (Animal, Vegetable)
- c) (Person, Lawyer)
- d) (Hand, Finger)
- e) (Car, Engine)

*a, c*

3) (10 points) In the following program, one of the lines labeled "Line 1" through "Line 5" has an error in it. Identify which line contains the error, and explain the error.

```
class Test {
    public static void main(String[] args) {
        Greetings g = new Greetings(); // Line 1
        Greetings.sayHello(); // Line 2
        Greetings.sayGoodbye(); // Line 3
        g.sayHello(); // Line 4
        g.sayGoodbye(); // Line 5
    }
} // end of class Test

class Greetings {
    static void sayHello() {
        System.out.println("Hello");
    }
    void sayGoodbye() {
        System.out.println("Goodbye");
    }
} // end of class Greetings
```

**Line 3:  
Instance method  
sayGoodbye()  
cannot be invoked  
without an object  
reference.**

4) (20 points) Write a definition for a class named Extremes. An instance of this class (an Extremes object) keeps track of the largest and smallest of a sequence of integers that it has "seen" (integers are "seen" by being passed to the methods of the object, as described below.) Your class may have whatever instance variables you like. It must implement the following methods:

a) `Extremes(int n)`

This is a constructor method. Its parameter is the first integer in the sequence to be "seen" by the constructed object.

b) `void see(int n)`

The parameter is another integer in the sequence to be "seen". This method should update the state of the Extremes object as needed.

c) `int largest()`

Returns the largest integer that the Extremes object has "seen" since it was constructed or reset (including the integer provided when the object was constructed or reset).

d) `int smallest()`

Returns the smallest integer that the Extremes object has "seen" since it was constructed or reset (including the integer provided when the object was constructed or reset).

e) `void reset(int n)`

Resets the Extremes object, beginning a new sequence of integers to be "seen", and provides the first integer in the new sequence.

*(Write your definition of the Extremes class on the next page.)*

*(Write your answer to Problem 4 here.)*

```
class Extremes {
    int big;
    int small;
    Extremes(int n) {
        big = n;
        small = n;
    }
    void see(int n){
        if (n > big)
            big = n;
        if (n < small)
            small = n;
    }
    int largest() {
        return big;
    }
    int smallest() {
        return small;
    }
    void reset(int n) {
        big = n;
        small = n;
    }
} // end of class Extremes
```

5) (10 points) What is printed by the following program?

```
class Test {
    public static void main(String[] args) {
        LicensePlate p1 = new LicensePlate("4", "2");
        LicensePlate p2 = new LicensePlate(4, 2);
        System.out.println("Plate p1 is " + p1);
        System.out.println("Plate p2 is " + p2);
    }
} // end of class Test

class LicensePlate {
    String s = "";
    int n = 0;
    LicensePlate(String s1, String s2) {
        this.s = s1 + s2;
    }
    LicensePlate(int n1, int n2) {
        this.n = n1 + n2;
    }
    public String toString() {
        return s + n;
    }
} // end of class LicensePlate
```

**Plate p1 is 420**  
**Plate p2 is 6**

6) (10 points) What is printed by the following program?

```
class Test {
    public static void main(String[] args) {
        Point p1 = new Point(1.0, 1.0);
        Point p2 = p1;
        Point p3 = new Point(p1);
        p1.stretch(2.0);
        System.out.println("p1 = " + p1);
        System.out.println("p2 = " + p2);
        System.out.println("p3 = " + p3);
    }
} // end of class Test

class Point {
    double x;
    double y;
    Point(double x, double y) { // constructor
        this.x = x;
        this.y = y;
    }
    Point(Point p) { // copy constructor
        this.x = p.x;
        this.y = p.y;
    }
    public String toString() {
        return "(" + x + ", " + y + ")";
    }
    void stretch(double factor) { // mutator
        x = x * factor;
        y = y * factor;
    }
} // end of class Point
```

**p1 = (2.0, 2.0)**

**p2 = (2.0, 2.0)**

**p3 = (1.0, 1.0)**

7) (10 points) What is printed by the following program?

```
class Test {
    public static void main(String[] args) {
        Sale s1 = new Sale(5, 10.00);
        Sale s2 = new TaxableSale(20, 5.00);
        Sale[] sales = {s1, s2};
        double total = 0.0;
        for (Sale s:sales) {
            System.out.println("Sale amount = "
                + s.saleAmount());
            total += s.saleAmount();
        }
        System.out.println("Total = " + total);
    }
} // end of class Test

class Sale {
    int quantity;
    double price;
    Sale(int quantity, double price) {
        this.quantity = quantity;
        this.price = price;
    }
    double saleAmount() {
        return quantity * price;
    }
} // end of class Sale

class TaxableSale extends Sale {
    static final double TAX = 0.1; // tax rate
    TaxableSale(int quantity, double price) {
        super(quantity, price);
    }
    double saleAmount() {
        return super.saleAmount() * (1.0 + TAX);
    }
} // end of class TaxableSale
```

**Sale amount = 50.0**

**Sale amount = 110.0**

**Total = 160.0**