

# COP 3330: Object-Oriented Programming Summer 2011

## Final Exam Review

Instructor : Dr. Mark Llewellyn  
markl@cs.ucf.edu  
HEC 236, 407-823-2790  
<http://www.cs.ucf.edu/courses/cop3330/sum2011>

Department of Electrical Engineering and Computer Science  
Computer Science Division  
University of Central Florida



# Material Covered – Exam 1

- **Introductory Notes (2 sets of notes).**
  - Basic OO concepts.
  - Criteria for elegant software.
  - Advantages of OO.
  - Introduction to variables, classes, and methods.
  - Introduction to UML.
  - Inheritance – Specialization and Generalization
  - Subtypes and Polymorphism
- The notes on the Java Environment are not covered on the exam.



# Material Covered – Exam 2

- Classes in Java – 5 sections of notes plus a prequel set of notes.
- Inheritance and polymorphism.
- Abstract classes and interfaces.



# New Material Covered

- Exception Handling in Java. You should be able to modify a segment of code that does not catch an exception into a segment of code that does catch an exception.
- Event-driven Programming in Java – GUIs. Only very basic stuff from this area will appear on the final exam.
- File I/O – basic concepts only
- Java Networking – not covered on final.



# Test Format

- Very similar to the other two tests.
- Some True/False questions
- Some fill-in-the-blanks questions.
- Some tracing through code and producing the output.
- Some writing of Java console application programs.



# Sample Test Questions

1. Write a complete Java program that will print the average of five integer numbers which are read from the keyboard with the following restrictions.
  - Do not use an array to store the integer values.
  - Prompt the user for each of the five input values.
  - Use a Scanner object for the input.



## Solution – Problem 1

```
public class AvgFive {
    public static void main(String[] args) throws IOException {
        final int LIST_SIZE = 5;
        int currentInput;
        int runningTotal = 0;
        //BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        Scanner stdin = new Scanner(System.in);
        System.out.println("Enter first integer: ");
        //currentInput = Integer.parseInt(stdin.readLine());
        currentInput = stdin.nextInt();
        runningTotal += currentInput;
        System.out.println("Enter second integer: ");
        //currentInput = Integer.parseInt(stdin.readLine());
        currentInput = stdin.nextInt();
        runningTotal += currentInput;
        System.out.println("Enter third integer: ");
        //currentInput = Integer.parseInt(stdin.readLine());
        currentInput = stdin.nextInt();
        runningTotal += currentInput;
        System.out.println("Enter fourth integer: ");
        //currentInput = Integer.parseInt(stdin.readLine());
        currentInput = stdin.nextInt();
        runningTotal += currentInput;
        System.out.println("Enter fifth integer: ");
        //currentInput = Integer.parseInt(stdin.readLine());
        currentInput = stdin.nextInt();
        runningTotal += currentInput;
        int average = runningTotal / LIST_SIZE;
        System.out.println("The input average is: " + average);
    } //end main method
} //end class AvgFive
```



```
class Thing{
    public int value;    public Thing one;    public Thing two;
    Thing() {value = 4;    one = null; two = null; }
    Thing(int inval, Thing x){value = inval; one = x; two = this;}
} //end class Thing

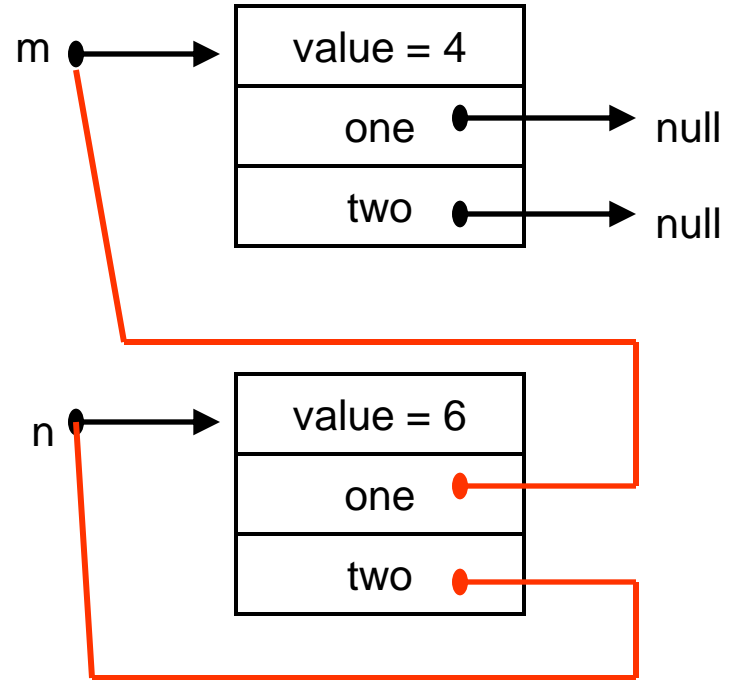
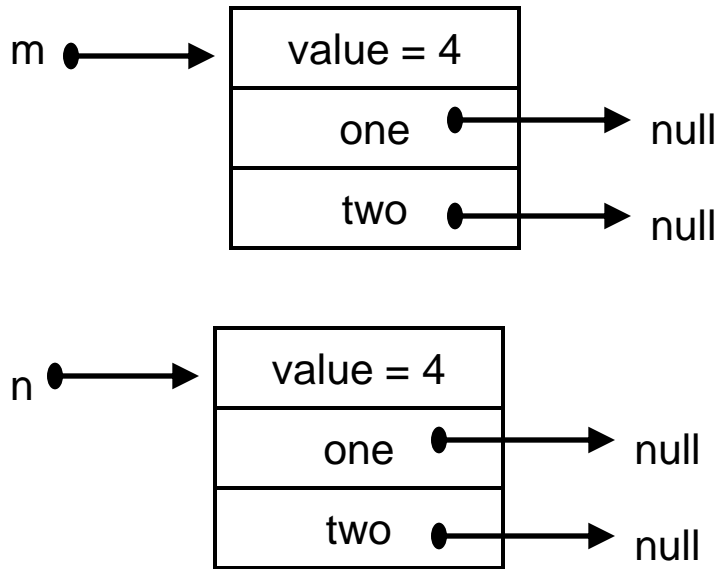
public class ReviewQuestion2 {
    public static void main (String args[]) {
        Thing m = new Thing();
        Thing n = new Thing();
        n.value = 6;    n.one = m;    n.two = n;
        Thing o = new Thing(8, new Thing());
        System.out.println("1: m.value = " + m.value);
        System.out.println("2: n.value = " + n.value);
        System.out.println("3: o.value = " + o.value);
        System.out.println("4: n.one.value = " + n.one.value);
        System.out.println("5: o.one.value = " + o.one.value);
        o.one.one = n.one;
        System.out.println("6: o.one.one.value = " + o.one.one.value);
        n.two = o.one.one;
        m.two = o.two;
        System.out.println("7: n.two.value = " + n.two.value);
        System.out.println("8: m.two.value = " + m.two.value);
        Thing p = new Thing(10, o.one);
        p.one.value = 12;
        System.out.println("9: p.value = " + p.value);
        System.out.println("10: p.one.value = " + p.one.value);
        System.out.println("11: o.one.value = " + o.one.value);
        o.one.two = p.two;
        System.out.println("12: o.one.two.value = " + o.one.two.value);
    } //end main method
}
```

**Problem 2 – Show the output from this program**



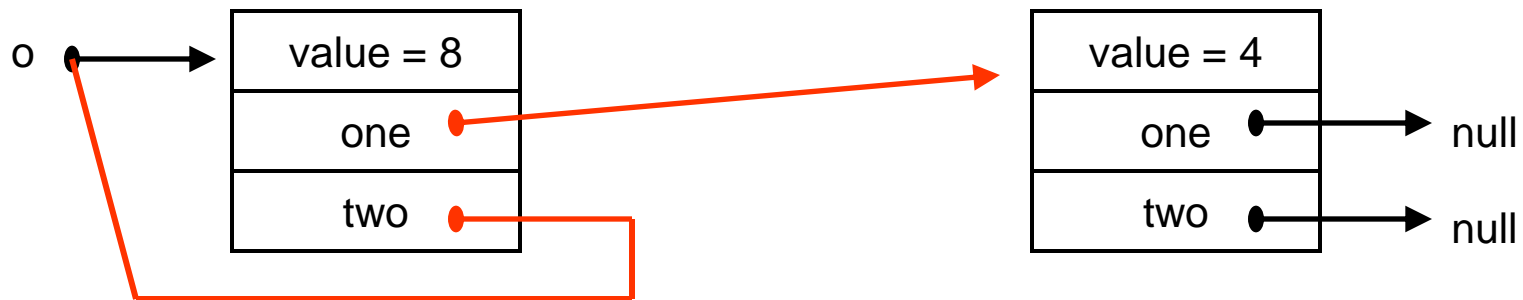
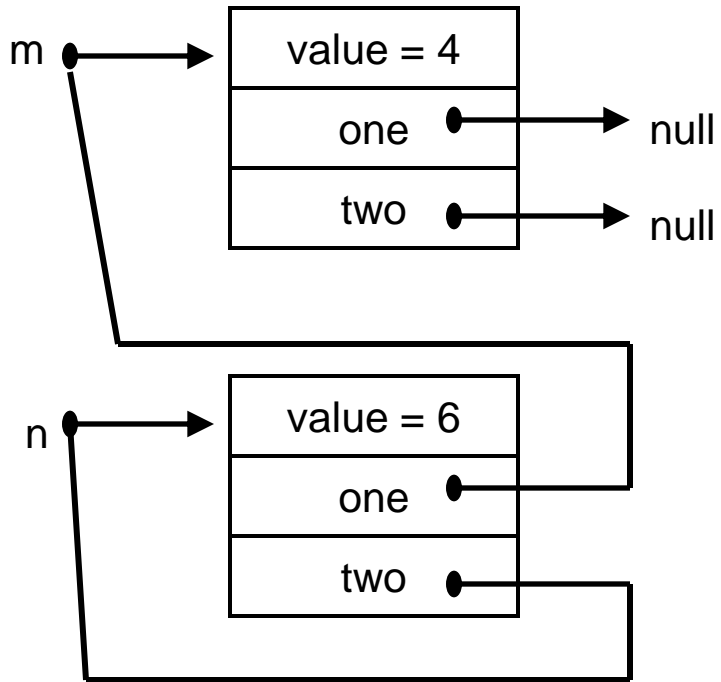


**Solution – Problem 2**



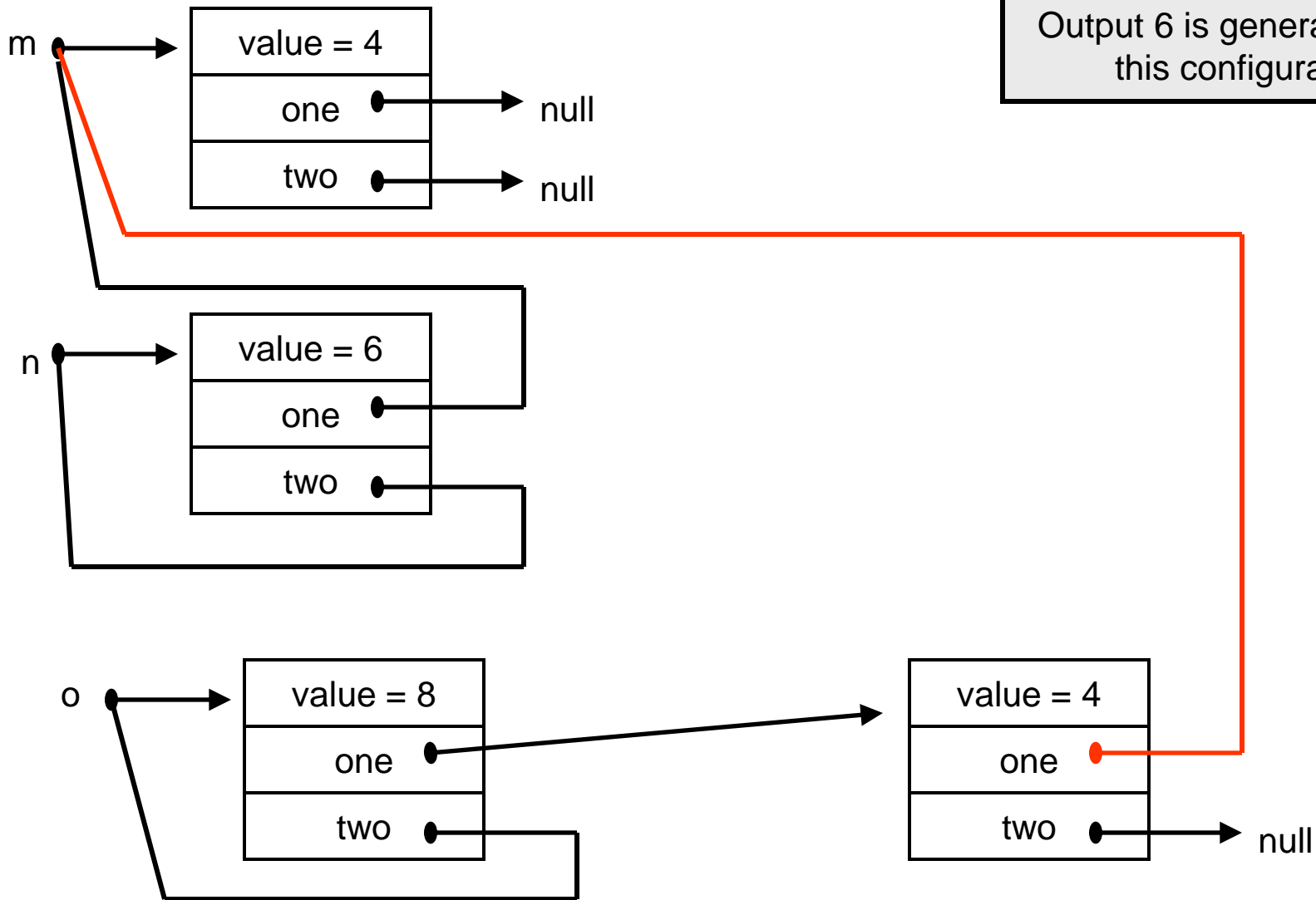
## Solution – Problem 2

First five output lines are generated from this configuration



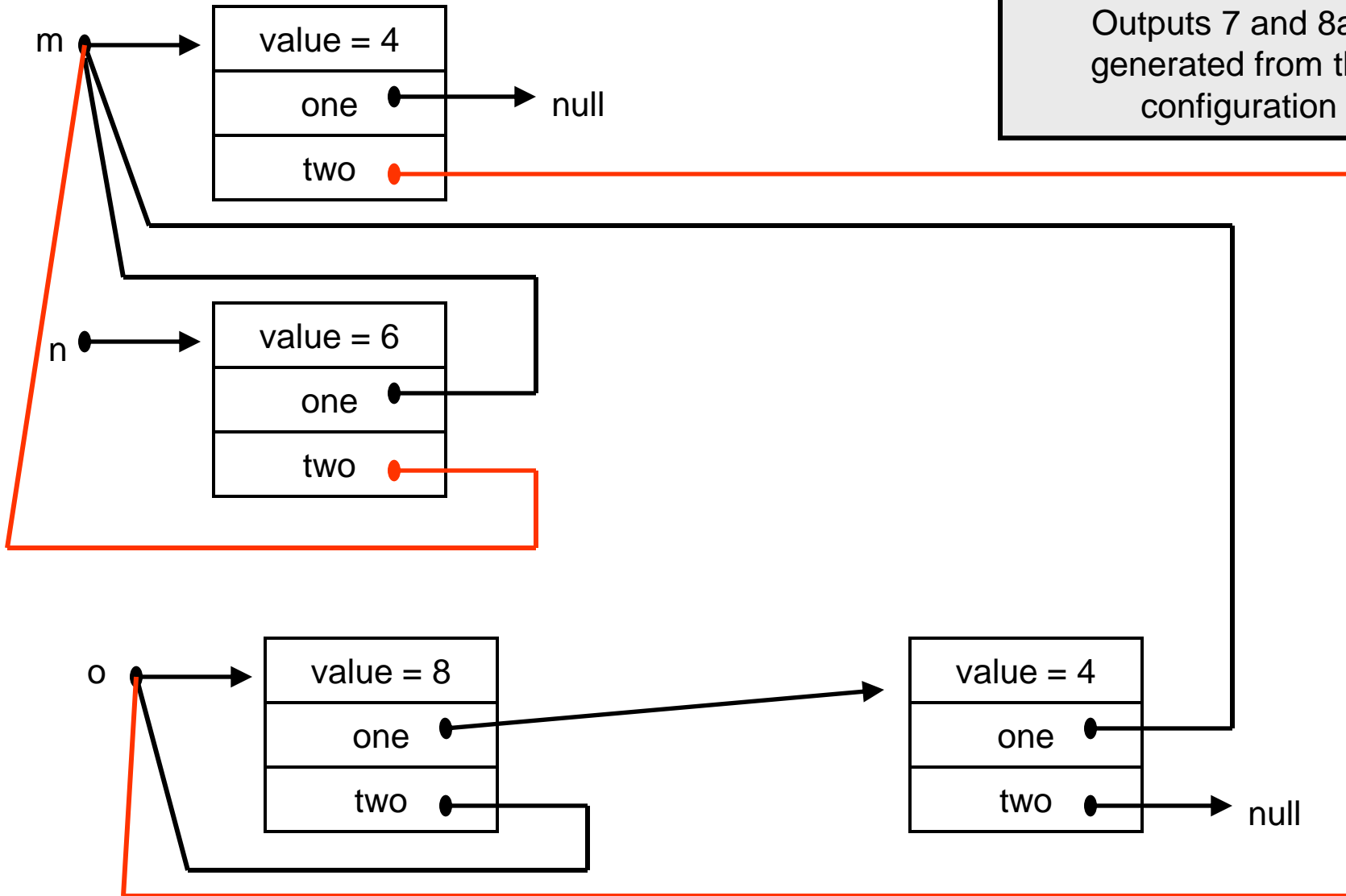
## Solution – Problem 2

Output 6 is generated from this configuration



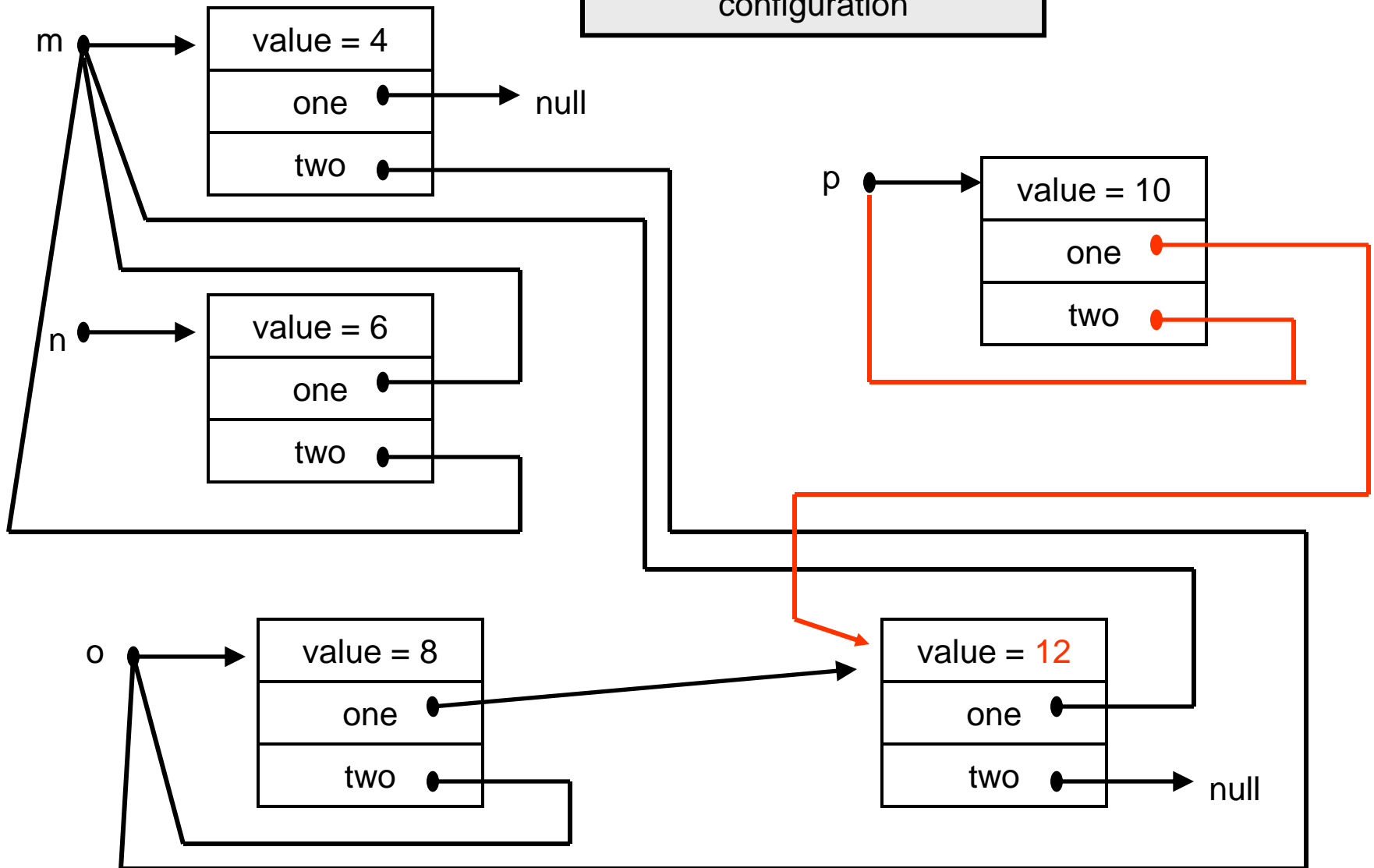
## Solution – Problem 2

Outputs 7 and 8 are generated from this configuration



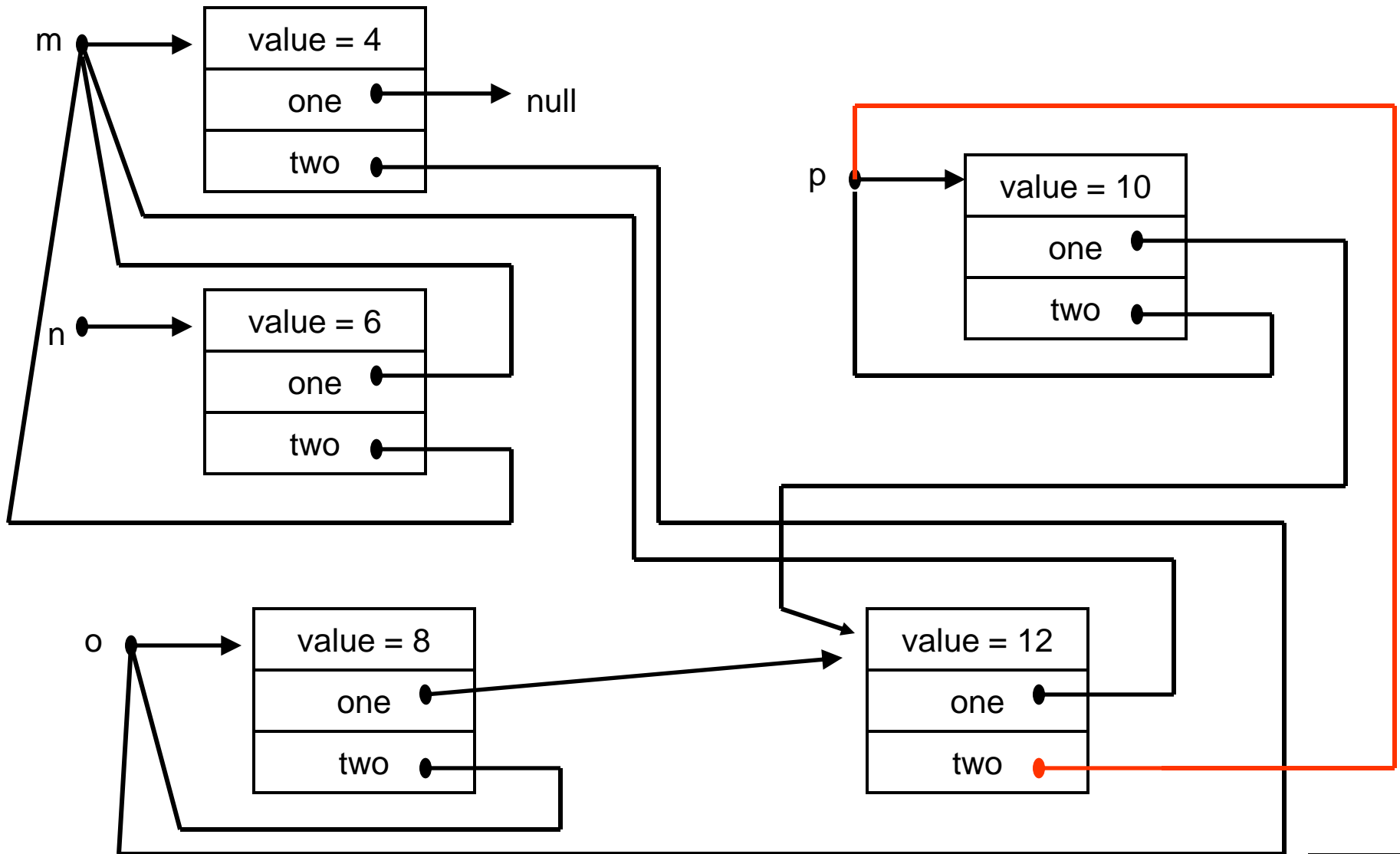
## Solution – Problem 2

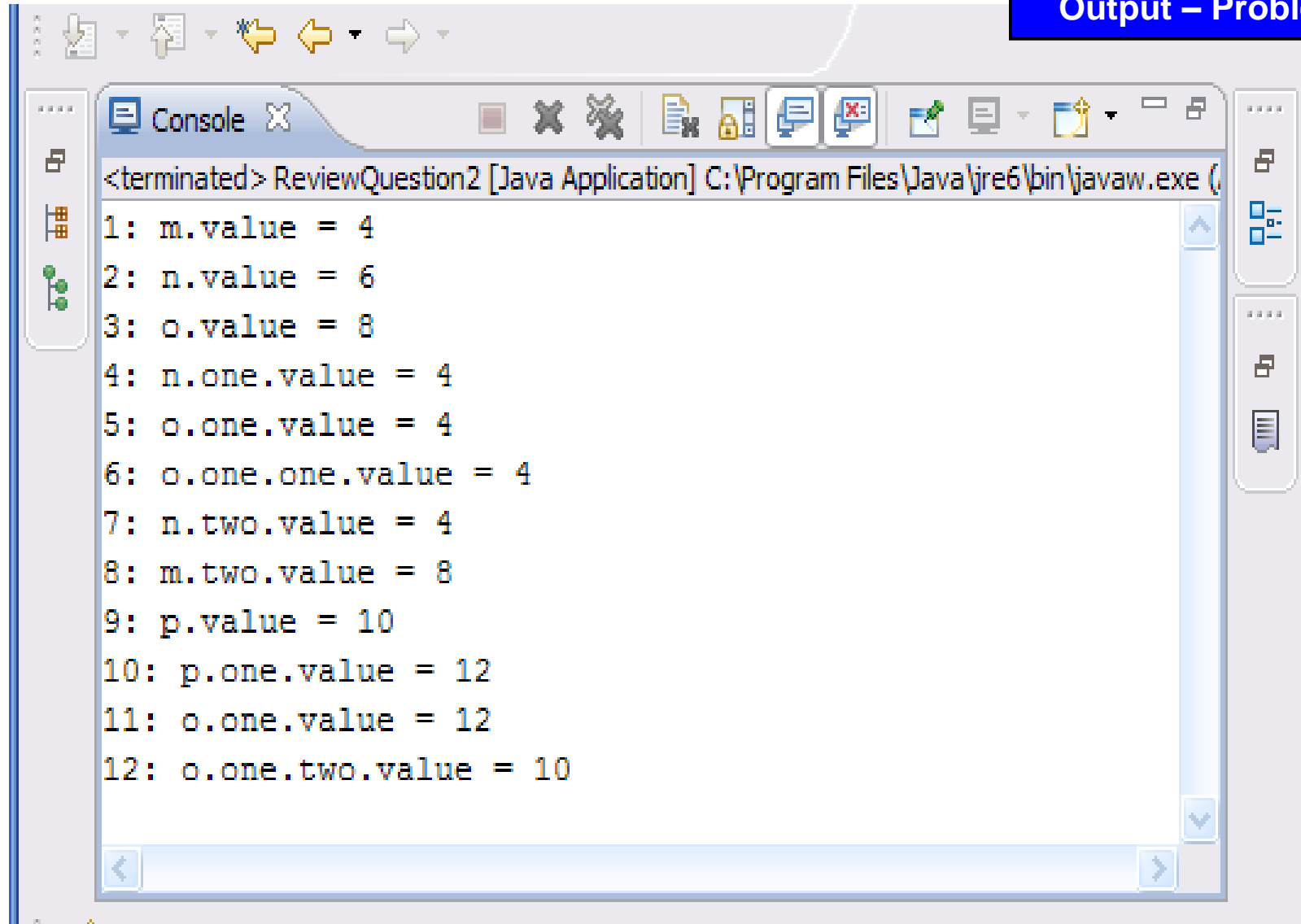
Outputs 9, 10, and 11 are generated from this configuration



## Solution – Problem 2

Final output 12 is generated from this configuration





```
<terminated> ReviewQuestion2 [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (
1: m.value = 4
2: n.value = 6
3: o.value = 8
4: n.one.value = 4
5: o.one.value = 4
6: o.one.one.value = 4
7: n.two.value = 4
8: m.two.value = 8
9: p.value = 10
10: p.one.value = 12
11: o.one.value = 12
12: o.one.two.value = 10
```



### 3. Draw the UML diagram for the following scenario.

We have three different “things” to keep track of:

**Customers** who have a name and an address.

Customers place **Orders** with our company. Each order has a date the order was placed, the current status of the order, and the amount in \$ of the order. For each order we need to calculate the tax to add to the order based on the amount of the order, and we'd like to be able to print all the details of an order.

Finally, **Payments** are made by customers to pay for the orders that they place with our company. Each payment consists of an specified amount in dollars.

Payments can be made by:

**Credit Card** in which case its number, type, and expiration date are to be maintained, and whether or not the charge was authorized;

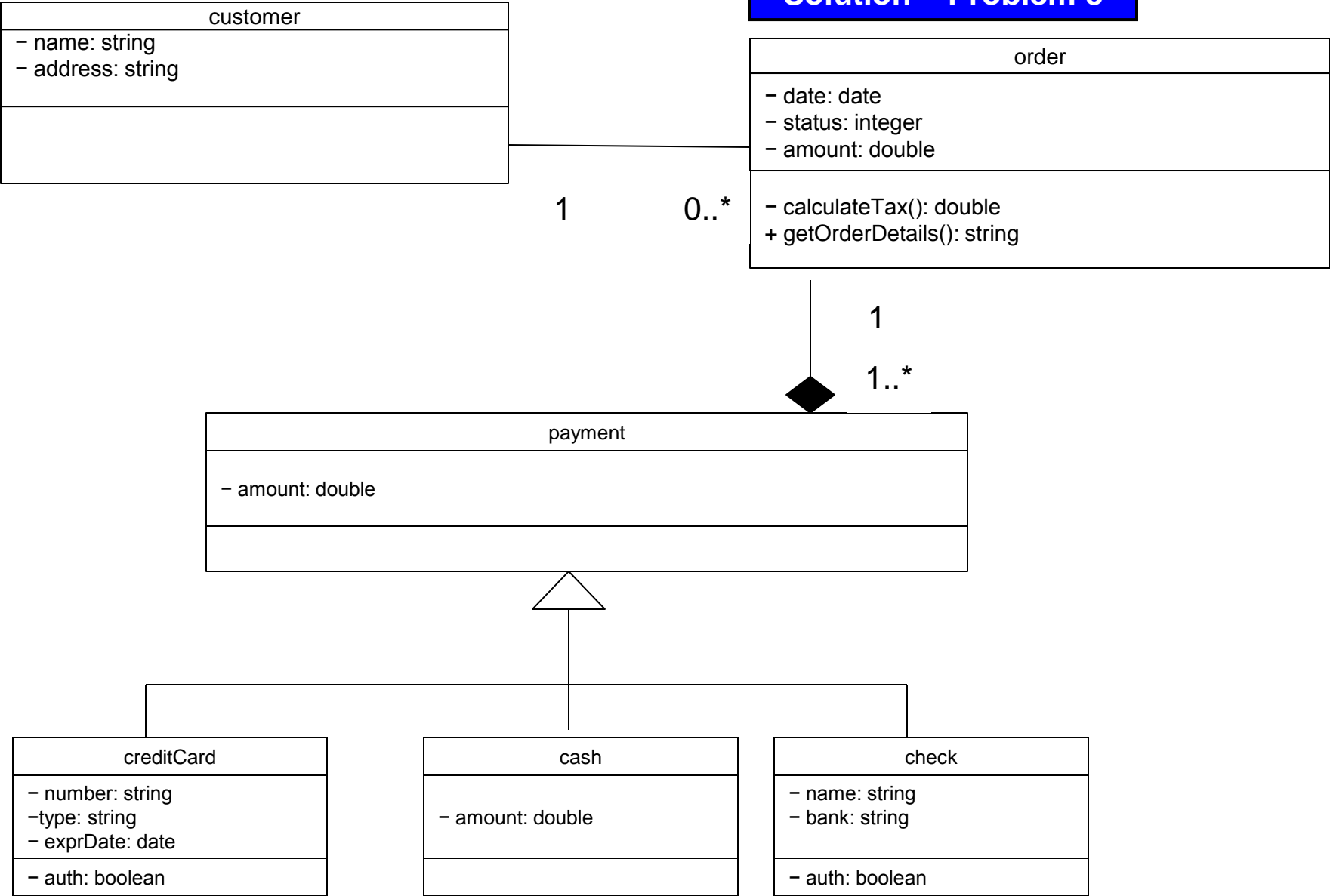
**Cash** in which case only the dollar amount is maintained.

**Check** in which case the name on the check and the name of the bank are recorded, and whether or not the charge was authorized.





# Solution – Problem 3



#### 4. What is the exact output from the following program?

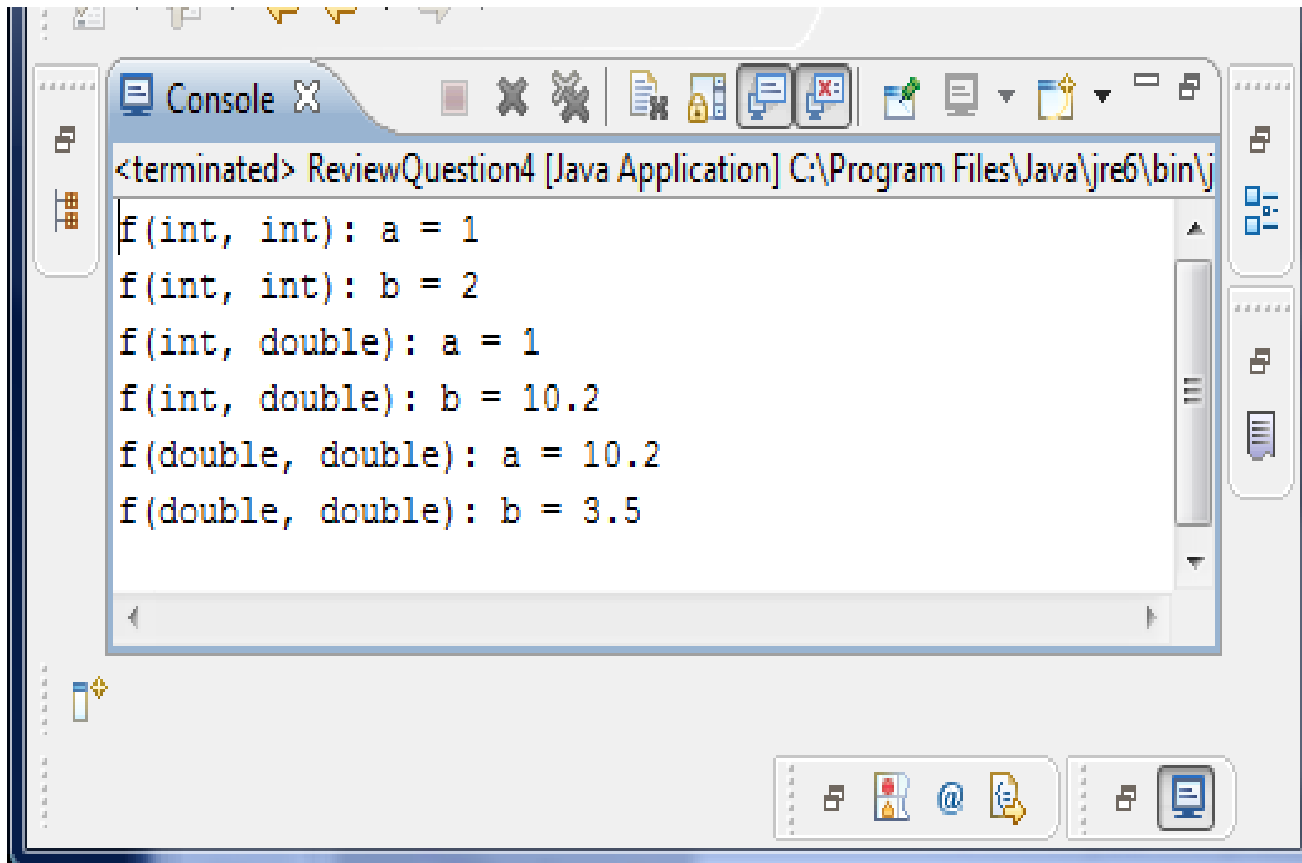
```
Question4.java AvgFive.java ReviewQuestion4.java >>80
//Class: ReviewQuestion4
//Final Exam Review - COP 3330 - Summer 2011

class Overload{
    public static void f(int a, double b) {
        System.out.println("f(int, double): a = " + a);
        System.out.println("f(int, double): b = " + b);
    }
    public static void f(int a, int b) {
        System.out.println("f(int, int): a = " + a);
        System.out.println("f(int, int): b = " + b);
    }
    public static void f(double a, double b) {
        System.out.println("f(double, double): a = " + a);
        System.out.println("f(double, double): b = " + b);
    }
}

public class ReviewQuestion4{
    public static void main(String[] args) {
        int i = 1;
        int j = 2;
        double x = 3.5;
        double y = 10.2;
        Overload.f(i,j);
        Overload.f(i,y);
        Overload.f(y,x);
    }
}
```



# Answer Sample Question #4



```
<terminated> ReviewQuestion4 [Java Application] C:\Program Files\Java\jre6\bin\j
f(int, int): a = 1
f(int, int): b = 2
f(int, double): a = 1
f(int, double): b = 10.2
f(double, double): a = 10.2
f(double, double): b = 3.5
```



## 5. What is the exact output from the following program?

```
ReadServerFileTest.j  Question4.java  AvgFive.java  ReviewQuestion4.java  ReviewQu
//Class: ReviewQuestion5
//Final Exam Review - COP 3330 - Summer 2011

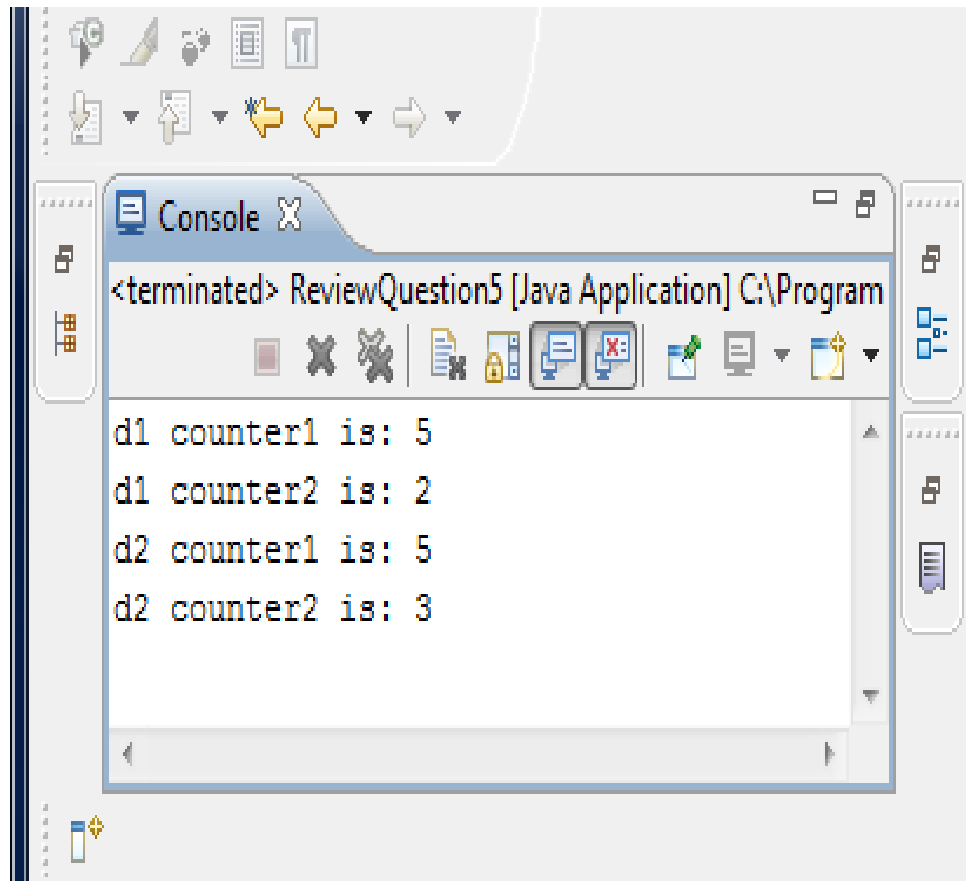
class D {
    private static int counter1 = 0;
    private int counter2 = 0;
    public D(){ }
    public void increment() {
        ++counter1;
        ++counter2;
    }
    public int getCounter1() {
        return counter1;
    }
    public int getCounter2() {
        return counter2;
    }
}

public class ReviewQuestion5 {
    public static void main(String[] args) {
        D d1 = new D();
        D d2 = new D();
        d1.increment();
        d2.increment();
        d1.increment();
        d2.increment();
        d2.increment();
        System.out.println("d1 counter1 is: " + d1.getCounter1());
        System.out.println("d1 counter2 is: " + d1.getCounter2());
        System.out.println("d2 counter1 is: " + d2.getCounter1());
        System.out.println("d2 counter2 is: " + d2.getCounter2());
    }
}

//end class ReviewQuestion5
```



# Answer Sample Question #5



The screenshot shows an IDE console window titled "Console" with a close button. The window displays the output of a Java application named "ReviewQuestion5". The output consists of four lines of text, each on a new line. The first two lines are for "d1" and the last two are for "d2". Each line shows the name of a counter and its value. The console window has a toolbar with various icons for actions like copy, paste, and search. The background of the IDE is light gray.

```
<terminated> ReviewQuestion5 [Java Application] C:\Program  
d1 counter1 is: 5  
d1 counter2 is: 2  
d2 counter1 is: 5  
d2 counter2 is: 3
```



# Sample Test Questions

6. What is the exact output of the program shown below?

```
//Class: ReviewQuestion6
//COP 3330 - Final Exam Review - Summer 2011

class F{
    private int value;
    public F() {
        value = 4;
        System.out.println("Created an F object: " + value);
    }
    public F(int x){
        value = x;
        System.out.println("Created an F object: " + value);
    }
    public int getvalue(){
        return value;
    }
}

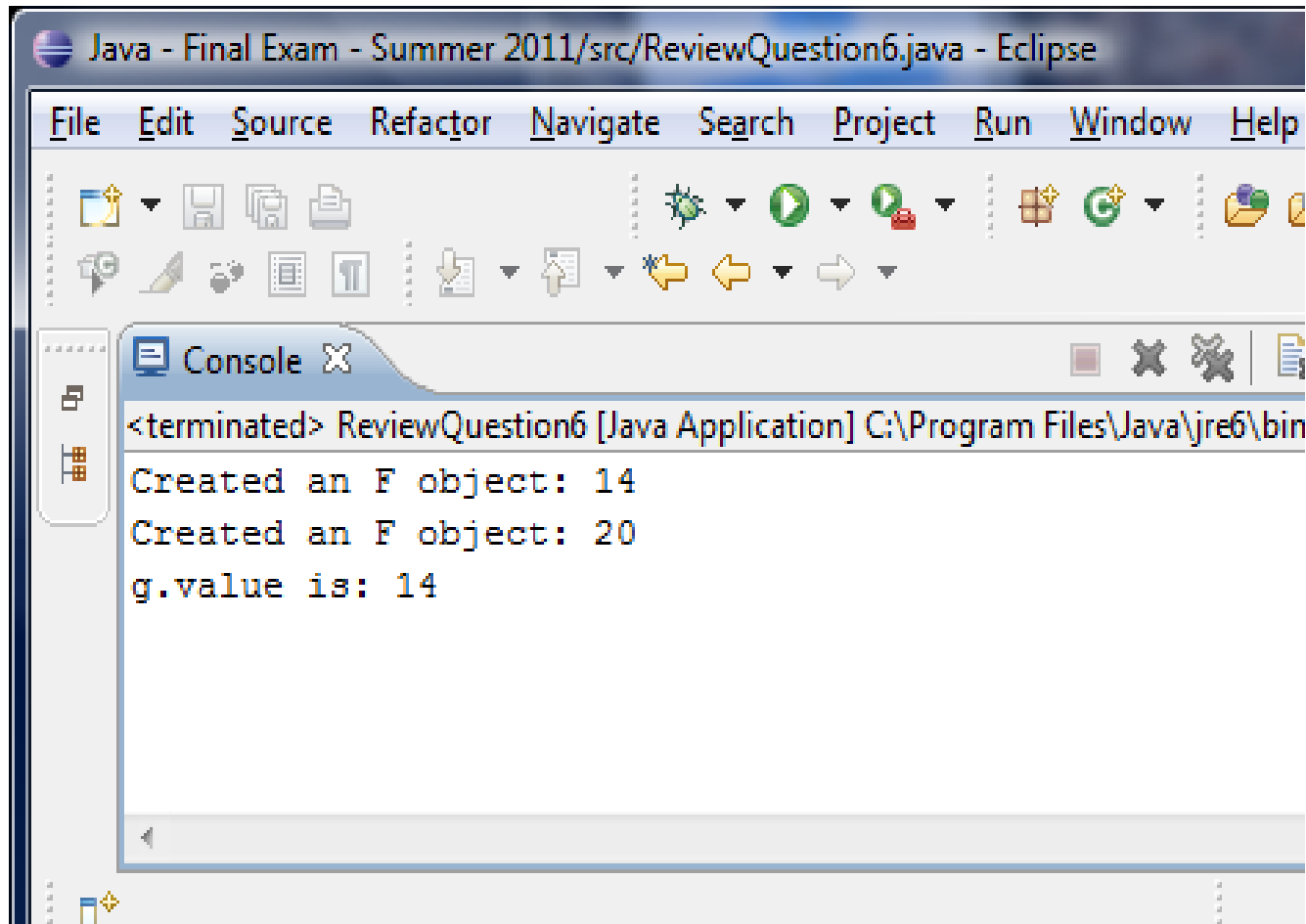
public class ReviewQuestion6 extends F {
    private F value;
    public ReviewQuestion6(int x, int y) {
        super(x);
        value = new F(y);
    }
    public static void main(String[] args) {
        ReviewQuestion6 g = new ReviewQuestion6(14, 20);
        System.out.println("g.value is: " + g.getvalue());
    }
}

}

```



# Answer Sample Question #6



The screenshot shows the Eclipse IDE interface. The title bar reads "Java - Final Exam - Summer 2011/src/ReviewQuestion6.java - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, debugging, and navigation. The Console window is open, displaying the following output:

```
<terminated> ReviewQuestion6 [Java Application] C:\Program Files\Java\jre6\bin
Created an F object: 14
Created an F object: 20
g.value is: 14
```



# Sample Test Questions

7. For each line in the main method in the program shown on the next page, state whether the line will result in a compilation error or not.  
Explain what the error is for those lines that will result in a compilation error and “OK” for lines that will not result in a compilation error.





```

//Class: ReviewQuestion7
//COP 3330 - Summer 2011

class C1 {
    public int r;
    protected double s;
    long t;
    public static int u;
} //end class C1

class C2 {
    public C1 z = new C1();
    public C2 u = new C2();
    protected float v;
    float w;
    private static int x;
} //end class C2

public class ReviewQuestion7 extends C1{
    public static void main(String args[]){
        C1 y = new C1();
        C2 m = new C2();
        y.r = 8;
        y.s = 3.12;
        y.t = y.t + 1;
        C1.u = 99;
        C2.x = 37;
        y.w = 5001;
    } //end main method
} //end class ReviewQuestion7

```



```
class C1 {
    public int r;
    protected double s;
    long t;
    public static int u;
} //end class C1
```

## Solution – Problem 7

```
class C2 {
    public C1 z = new C1();
    public C2 u = new C2();
    protected float v;
    float w;
    private static int x;
} //end class C2
```

```
public class ReviewQuestion7 extends C1{
    public static void main(String args[]){
        C1 y = new C1();
        C2 m = new C2();
        y.r = 8;
        y.s = 3.12;
        y.t = y.t + 1;
        C1.u = 99;
        C2.x = 37;
        y.w = 5001;
    } //end main method
```

OK

**Error** – x is a private variable in class C2 and is not visible here

**Error** – object y is of type C1, class C1 has no instance variable named w.



# Sample Test Questions

8. Modify the code shown on the following page so that it will handle an `InputMismatchException` by informing the user that an integer must be entered at the prompt and will repeatedly do so until the user enters an integer value.



```
//Class: ExceptionDemo
//Final Exam Review - COP 3330 - Summer 2011
//MJL 7/28/2011

import java.util.Scanner;

public class ExceptionDemo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = input.nextInt();

        // Display the result
        System.out.println(
            "The number entered is " + number);
    } //end main method
} //end class ExceptionDemo
```



```
//Class: HandleExceptionDemo
//Final Exam Review - COP 3330 - Summer 2011
//Solution to review problem 8
//MJL 7/28/2011

import java.util.*;

public class HandleExceptionDemo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        boolean continueInput = true;
        do {
            try {
                System.out.print("Enter an integer: ");
                int number = input.nextInt();

                // Display the result
                System.out.println(
                    "The number entered is " + number);
                continueInput = false;
            } //end try block
            catch (InputMismatchException ex) {
                System.out.println("Try again. (" +
                    "Incorrect input: an integer is required)");
                input.nextLine();
            } //end catch block
        } while (continueInput);
    } //end main method
} //end class HandleExceptionDemo
```

**Solution – Problem 8**