MIDTERM EXAM #2 SOLUTIONS

November 21, 2006

Closed Book
10 Questions

NAME: ________________________________________________________

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>POINTS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td>100/100</td>
</tr>
</tbody>
</table>
1. In the space below write a Java program method that computes the wages of an hourly employee. The method should take the number of hours worked and the pay rate of the employee. The method should return the dollar amount. The computation of wages should include the fact that employees who work more than 40 hours get time and a half for overtime pay. You only need to write the method, not a full class or a calling method.

// an “if” statement question
//Modification of listing 5.2 in book:

public double wages(double hours, double rate) {

    final double STANDARD = 40.0;
    double wages;
    if (hours > STANDARD)
        wages = STANDARD * rate
            + (hours - STANDARD) * (rate * 1.5);
    else
        wages = hours * rate;

    return wages;
}

1 point for correct method structure
   (don't worry about “public” or “static”)
2 points for correct parameters
2 points for correct if condition
2 points for “then” clause
2 points for correct “else” clause
1 point for correct return statement
(can lose at most one point for bad punctuation)
2. Write a JAVA method that takes three integers parameters and returns the minimum of the three integers. Use nested if statements to determine the minimum value.

//modification of Listing 5.6 in book

public int minOfThree(int x, int y, int z) {
    int min;
    if (x < y)
        if (x < z)
            min = x;
        else
            min = z;
    else
        if (y < z)
            min = y;
        else
            min = z;

    return min;
}

1 point for correct method structure
(don't worry about "public" or "static")
3 points for correct parameters
3 points for correct if condition
2 points for nesting structure
1 point for correct return statement
(can loose at most one point for bad punctuation)
3. Write a JAVA method that takes a single character representing a letter grade as a parameter and returns a String with the appropriate feedback. Use a switch statement to assign the correct response. If the grade is an 'A', the response should be “Great Job!”. If the grade is a 'B', the response should be “Above Average”. If the grade is a 'C', the response should be “Average”. If the grade is an 'D', the response should be “Study Harder”. If the grade is a “F”, the response should be “Get Help”. For any other character, the response should be “Unexpected Grade - see instructor”.

public String feedback(char c) {
    String response;
    switch (c) {
        case 'A' : response = “Great Job!”; break;
        case 'B' : response = “Above Average”; break;
        case 'C' : response = “Average”; break;
        case 'D' : response = “Study Harder”; break;
        case 'F' : response = “Get Help.”; break;
        default  : response = “Unexpected Grade - see instructor”;
    }
    return response;
}

6 points for correct case statements (1 point each)
1 point for correct “switch” clause
1 point for proper “break;” statements
    (only one point at most for forgetting breaks)
1 point method structure
1 point correct return
(can loose at most one point for bad punctuation)
4. Write a JAVA program consisting of a class containing a single main method. The method should compute the average of a set of values entered by the user. The method should keep prompting the user to “Enter an integer (0 to quit)” until the user enters a sentinel value of zero. It is not known how many values the user will enter. After the user enters zero, the method should print the average of the values entered (not including the final zero). The method should print “No values were entered” when appropriate. The user enters integers, but the final average should be a double. It is not necessary to do any decimal formatting. Include appropriate import statements.

```java
import java.util.Scanner;
public class Average {
    public static void main(String[] args) {
        int sum = 0, value, count = 0;
        double average;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter an integer (0 to quit) ");
        value = scan.nextInt();
        while (value != 0) {
            count ++ ;
            sum += value;
            System.out.print("Enter an integer (0 to quit) ");
            value = scan.nextInt();
        }
        if (count == 0)
            System.out.println("No values were entered");
        else {
            average = (double) sum / count;
            System.out.println("Average == "+ average);
        }
    }
}
```

1 point for import statement
1 point for class definition
1 point main method declaration
1 point variable declaration
1 point for scanner construction.
1 point for loop control (while condition)
1 point for correct calculation in loop
1 point for ending properly
   (without including zero)
1 point for if statement check on count of zero
1 point for proper calculation of average (with type conversion)
5. Write a JAVA method that takes an array of integers as a parameter and uses a for loop to find and return the minimum value integer in the array. Just write the method, not a whole class or program.

```java
public int minfinder(int x[]){
    min = x[0];
    for (int i = 0 ; i < x.length ; i++)
        if (x[i] < min)
            min = x[i];
    return min;
}
```

1 point for method declaration
1 point for initialization of min
3 points for “for” (1 point per clause)
2 points for if statement
1 point for proper update in if statement
1 point for return value
1 point for punctuation
public int[][] mtable(int a, int b) {
    int min = (a < b) ? a : b;
    int max = (a < b) ? b : a;

    int size = max - min + 1;
    int [][] table = new int[size][size];

    for (int i = 0; i < size; i++) {
        int row = min ;
        for(j = 0; j < size; j++) {
            int col = min ;
            table[i][j] = col * row ;
            col++;
        }
        row++
    }
    return table;
}

2 points for method declaration
1 point for correct size calculation
2 points for outer loop expression
2 points for inner loop expression
2 points for correct filling in of table
1 point for return value return

2 points EXTRA CREDIT for checking min & max (no points off for assuming b>=a);
7. Write the JAVA definition for an “Address” class. An address consists of a street address, a city, a state (strings) and a zipcode (long). Your class should include two methods: a constructor and a toString method. Accessor and mutator methods are not required.

```java
public class Address {
    private String street;
    private String city;
    private String state;
    private long zipcode;

    public Address(String street,
                    String city,
                    String state,
                    long zipcode) {
        this.street = street;
        this.city = city;
        this.state = state;
        this.zipcode = zipcode;
    }

    public String toString() {
        return street + 
               city + state + zipcode ;
    }
}
```

1 point for proper declarations of class & methods
3 points for proper structure (data, methods)
2 points for constructor
2 points for toString
1 point for spaces (commas, delimiters) in toString return value
1 point for punctuation
8. What is the output of the following (three class) program?

```java
public class Ptest{
    public static void main(String[] args) {
        Pmod modder = new Pmod();
        int x1 = 123;
        Thing x2 = new Thing(456);
        Thing x3 = new Thing(789);

        System.out.println("main Before");
        System.out.println(x1 + " " + x2 + " " + x3);

        modder.cVal(x1,x2,x3);

        System.out.println("main After");
        System.out.println(x1 + " " + x2 + " " + x3);
    }
}

class Pmod{
    public void cVal(int k1, Thing k2, Thing k3) {
        System.out.println("Pmod Before");
        System.out.println(k1 + " " + k2 + " " + k3);
        k1 = 987;
        k2.setVal(654);
        k3 = new Thing(321);
        System.out.println("pmod After");
        System.out.println(k1 + " " + k2 + " " + k3);
    }
}

public class Thing {
    private int y;
    public Thing (int y) { this.y = y;}
    public void setVal (int y) {this.y = y;}
    public String toString() { return y + ""; }
}
```

(1 point per line – 2 points for correct order of lines)
main Before
123 456 789
Pmod Before
123 456 789
pmod After
987 654 321
main After
123 456 789
9. What is the output of the following program?

```java
public class CylinderTest2 {
    public static void main(String [] args) {
        Point pnt1;
        Circle cirl;
        Cylinder cyl1;

        pnt1 = new Point(1,2);
        cirl = new Circle(3,4,5.0);
        cyl1 = new Cylinder(6,7,8.0,9.0);

        Object [] oArray1 = new Object[3];

        oArray1[0] = pnt1;
        oArray1[1] = cirl;
        oArray1[2] = cyl1;

        for (int i = 0 ; i < oArray1.length; i++) {
            Object myObj = oArray1[i];
            System.out.println("oArray["+i+"] : "+myObj.toString());

            if (myObj instanceof Circle)
                System.out.println("\t Diameter = "+((Circle) myObj).getDiameter());
            if (myObj instanceof Cylinder)
                System.out.println("\t Cylinder volume = "+((Cylinder) myObj).getHeight());
        }
    }
} // CONTINUED ON NEXT PAGE
public class Point {
    private int x, y;
    public Point(int xValue, int yValue) {
        x = xValue;
        y = yValue;
    } public String toString() {
        return "["+ getX() + "," + getY() +"]";
    }
}

public class Circle extends Point {
    private double radius;
    public Circle(int xValue, int yValue, double radiusValue) {
        super(xValue, yValue);
        radius = radiusValue;
    } public double getRadius() { return radius; }
    public double getDiameter() { return 2 * getRadius(); }
    public String toString() {
        return "Center = " + super.toString() + "; Radius = " + getRadius();
    }
}

public class Cylinder extends Circle {
    private double height;
    public Cylinder(int xValue, int yValue,
            double radiusValue, double heightValue) {
        super(xValue,yValue,radiusValue);
        height = heightValue;
    }
    public double getHeight() { return height; }
    public String toString() {
        return super.toString() + "; Height = " + height;
    }
}

o Arry1[0] : [1,2]  (1 point)
o Arry1[1] : Center = [3,4], Radius = 5.0  (1 point)
        Diameter = 10.0 ; (1 point)
o Arry1[2] : Center = [6,7], Radius = 8.0, Height = 9.0  (2 points)
        Diameter = 16.0  (2 points)
        Cylinder Volume = 9.0 ; (2 points)

1 point for punctuation

(ANSWER TO Q9)
10. Write a JAVA method called “divide” which takes two integers as parameters, divides the first by the second and returns the result. Use a try/catch structure in your method (no credit for doing it with an “if” statement). Your method should attempt to perform the operation. In the case of a divide by zero error, have your method return -1.

```java
public static int divide (int x, int y) {
    int z = 0;
    try { z = x / y }
    catch (Exception e) {
        z = -1;
    }
    return z;
}
```

any workable variant ok
1 point for punctuation
1 points for method return type declaration
1 point each for parameter declaration
1 point for proper return
3 points for try block
3 points for catch block