

EXAM #1 ANSWERS

Ted Pawlicki

November 23, 2004

Closed Book
10 Questions

NAME: _____

<i>QUESTION</i>	<i>POINTS</i>	<i>SCORE</i>
1	8	
2	10	
3	10	
4	10	
5	12	
6	8	
7	10	
8	12	
9	10	
10	10	
TOTAL	100	/100

1. In the space below write a simple, but complete Java program that prints your name to the console. This should be done by defining a class with a single main method.

```
public class SimpleTest {  
    public static void main(String [] args) {  
        System.out.println("Ted Pawlicki");  
    }  
}
```

2. Write a JAVA class **Circle** that has methods `getArea()` and `getPerimeter()`. A circle is defined by an (x,y) location of the center and a radius. All instance variables should be private. You should supply a single constructor that takes three parameters, accessor and mutator methods for all instance variables.

```
public class Circle {  
  
    private double x, y, radius; // int ok  
  
    public Circle(double x, double y, double radius){  
  
        this.x = x;  
        this.y = y;  
        this.radius = radius;  
    }  
  
    public void setx(double new_x) { x = new_x ;}  
    public double getx() { return x;}  
  
    public void sety(double new_y) { y = new_y ;}  
    public double gety() { return y;}  
  
    public void setradius(double new_radius) {  
        radius = new_radius ;  
    }  
    public double getradius() { return radius;}  
}
```

3. Supply values for the following (one point, each)

1. Convert from Decimal to Binary:

1. 17_____10001_____

2. 35_____100011_____

3. 72_____1001000_____

2. Convert from Binary to Decimal:

1. 11010_____26_____

2. 10111_____23_____

3. Evaluate the following

1. $(\text{true} \ \&\& \ (5 < 12) \ || \ (7 == 3))$ _____true_____

2. $((3 < 7) ? (2 > 3) : ((7-9) <= 0))$ _____false_____

4. $(\text{new String}(\text{"Hello 171"})).\text{equals}(\text{"Hello " + 171})$ __true__

5. $(\text{new String}(\text{"Ted"})).\text{compareTo}(\text{"Thad"}) > 0$ _false_____

4. In the space below, write a static JAVA method called "feedback". The method should take a decimal value parameter called "grade" and return a String. When the grade is below 65 the method should return, "Not So Good", when the grade is above 90 the string should be "Keep up the good work", all other values should return the string "so, so".

```
public static String feedback(double grade) {  
    if (grade < 65) return "Not So Good";  
    if (grade > 90) return "Keep up the good work";  
    return "so, so";  
}
```

5. Write Java method that takes in a double representing an amount of money (1.15 represents one dollar and fifteen cents) and returns a String representing the "change" that can be made from the money in terms of the least number of Quarters, Dimes, Nickles, and Pennies. (e.g. "\$1.19 can be made with 4 Quarters, 1 Dimes, 1 nickles, and 4 pennies", or "\$0.11 can be made with 0 Quarters, 1 dimes, 0 nickels, and 1 pennies"). Don't worry about the number of digits in the decimal value. Don't worry about pluralization.

```
public static String change(double amount) {
    int quarters, dimes, nickles, pennies;
    int qval, dval, nval, pval, rval;
    qval = 25; dval = 10 ;
    nval = 5; pval = 1;
    quarters = dimes = nickles = pennies = 0;
    rval = (int) Math.round(amount * 100);
    while (rval >= pval) {
        if (rval >= qval) {
            quarters++;
            rval -= qval;
        } else if (rval >= dval) {
            dimes++;
            rval -= dval;
        } else if (rval >= nval) {
            nickles++ ;
            rval -= nval ;
        } else if (rval >= pval) {
            pennies++;
            rval -= pval;
        }
    }
    return "$" + amount + " Can be made with "
        + quarters + " Quarters, "
        + dimes + " Dimes, "
        + nickles + " Nickles, and "
        + pennies + " Pennies. ";
}
```

6. Write a Java method using nested iteration that takes a positive integer x as a parameter and prints a multiplication table from 1 times 1 up to x times x to System.out.

```
public static void mtable (int x) {  
    for (int i = 1 ; i <= x ; i++) {  
        for (int j = 1; j <= x; j++)  
            System.out.print( " " + (i * j));  
        System.out.println();  
    }  
}
```

7. What is the output of the following program?

```
public class Advice {  
  
    public final static int LITTLE_ADVICE = 0;  
    public final static int MORE_ADVICE = 1;  
    public final static int LOTS_OF_ADVICE = 2;  
  
    public static void main(String[] args) {  
        for (int some_advice_please = 0 ;  
            some_advice_please <= 3;  
            some_advice_please++)  
            dispenseAdvice(some_advice_please);  
    }  
  
    public static void dispenseAdvice(int howMuchAdvice) {  
        switch (howMuchAdvice) {  
            case LOTS_OF_ADVICE:  
                System.out.println("See no evil");  
            case MORE_ADVICE:  
                System.out.println("Hear no evil");  
            case LITTLE_ADVICE:  
                System.out.println("Speak no evil");  
                break;  
            default:  
                System.out.println("No advice");  
        }  
    }  
}
```

Speak no evil
Hear no evil
Speak no evil
See no evil
Hear no evil
Speak no evil
No advice

8. Three parts:

1. Write a JAVA class "Room" - rooms are described by a room number and a building name. You need only supply one constructor. Since the location of a room never changes, make the instance variables private and only supply accessor methods, not mutator methods. (3 methods)
2. Write a JAVA Class "Classroom" that is a subclass of Room - classrooms are rooms with a number of seats in them. Again one constructor, private instance variables, but allow the number of seats to be changed via appropriate methods. (3 methods)
3. Write a JAVA Class "LectureHall" that is a subclass of Classroom - lecture halls are classrooms with a name of the type of sound system. (3 methods)

```
public class Room {
    private String buildingName;
    private int roomNumber ;

    public Room(String buildingName, int roomNumber){
        this.buildingName = buildingName;
        this.roomNumber = roomNumber;
    }

    public String getbuildingName() {
        return buildingName;
    }

    public int getRoomNumber() {
        return roomNumber ;
    }
}
```

Extra sheet for answers

```
public class Classroom extends Room {
    private int seatingCapacity ;

    public Classroom(String buildingName,
                    int roomNumber,
                    int seatingCapacity) {
        super(buildingName,roomNumber);
        setseatingCapacity(seatingCapacity);
    }

    public int getSeatingCapacity() {
        return seatingCapacity;
    }

    public void setseatingCapacity(int seats) {
        seatingCapacity = seats;
    }
}

public class LectureHall extends Classroom {
    private String soundSystem ;

    public LectureHall(String buildingName,
                    int roomNumber,
                    int seatingCapacity
                    String soundSystem) {
        super(buildingName,
            roomNumber,
            seatingCapacity);
        setsoundSystem(soundSystem);
    }

    public String getsoundSystem() {
        return soundSystem;
    }

    public void setsoundSystem(String soundSystem) {
        this.soundSystem = soundSystem;
    }
}
```

9. Write a JAVA method that takes a reference to an array of integers as a parameter and reverses the ordering of the array. (e.g. If the array passed in is {7,2,6,3,5} then it should be changed to {5,3,6,2,7}.)

```
public static void reverse(int [] a) {  
    for (int i = 0 ; i < a.length / 2; i++) {  
        int temp = a[i];  
        a[i] = a[a.length - 1 - i];  
        a[a.length - 1 - i] = temp;  
    }  
}
```

10. What is the output of the following program?

```
public class MethodStuff {
    static int a;

    public static void main(String[] args) {
        a = 4;
        System.out.println("foo("+a+") == " + foo(a));
        System.out.println("a == " + a);
    }

    public static int foo(int x) {
        System.out.println("foo" + x);
        return bar(x--) + bar(x);
    }

    public static int bar(int a) {
        System.out.println("bar" + a);
        a-- ;
        return a + zed(a -1);
    }

    public static int zed (int k) {
        a-- ;
        if (k < 0) return 1;
        System.out.println("zed" + k);
        return zed( --k);
    }
}
```

(Score 1 point per line)

```
foo4
bar4
zed2
zed1
zed0
bar3
zed1
zed0
foo(4) == 7
a ==-3
```