SI@UCF Java GUI Test #1 Solutions
6/17/2016

1) (15 pts) A geometric sequence is one where each subsequent term is created by multiplying the previous term by a common ratio. For example, the geometric sequence of 5 terms, starting with 3 with a common ratio of 2 is 3, 6, 12, 24 and 48. Complete the program below so that it calculates the ***sum*** of the geometric sequence specified by the user.

```
public class sum {

    public static void main(String[] args) {

        Scanner stdin = new Scanner(System.in);
        int sum = 0;
        System.out.println("How many terms are in the sequence?");

        int numTerms = stdin.nextInt() ; // 0 pt
        System.out.println("What is the first term of the sequence?")

        int firstTerm = stdin.nextInt() ; // 0 pt
        System.out.println("What is the common ratio between terms?")

        int ratio = stdin.nextInt() ; // 0 pt

        for (int i=0; i<numTerms; i++) {    // 5 pts
            sum += firstTerm;               // 5 pts
            firstTerm *= ratio;             // 5 pts
        }

        System.out.println("The sum of the sequence is "+sum+".");
    }
}
```

2) (10 pts) Complete the segment of code below so that it reads in the length of the hypotenuse of a right triangle and one of its legs and prints out the length of the other leg. (If the input is 5 and 3, respectively, the output should be 4.)

```
Scanner stdin = new Scanner(System.in);
System.out.println("Enter the hypotenuse of the right triangle.");
double hyp = stdin.nextDouble();
System.out.println("Enter one side of the right triangle.")
double side1 = stdin.nextDouble();
double side2;

double side2 = Math.sqrt(Math.pow(hyp,2)-Math.pow(side1,2));

System.out.println("The other side has length "+side2+".");

// Grading: 3 pts for each pow, 1 pt subtract, 2 pts sqrt,
// 1 pt assign
```

3) (10 pts) What is the output of the following program if the user enters the integer 82346231.

```
import java.util.*;

public class Q3 {

    public static void main(String[] args) {

        Scanner stdin = new Scanner(System.in);
        System.out.println("What value of n?");
        int n = stdin.nextInt();
        int res = 0;

        while (n > 0) {
            res = 10*res + n%10;
            n = n/10;
        }
        System.out.println(res);
    }
}
```

13264328 (Grading: 1 pt per correct digit, 2 pts bonus for getting all)

4) (10 pts) What is the output of the following program?

```
public class Q4 {
    public static void main(String[] args) {
        for (int i=0; i<5; i++) {
            for (int j=0; j<5; j++) {
                 if ((i+j)%2 == 0)
                      System.out.print("X");
                 else
                      System.out.print("O");
            }
            System.out.println();
        }
    }
}
```

XOXOX
OXOXO
XOXOX
OXOXO
XOXOX

Grading:2 pts for having 5 rows, 2 pts for having 5 cols, 3 pts for placement of X's, 3 pts for placement of Os.

Consider creating a class to store a food object. Each food object has a weight (int in ounces) and a number of calories (double). Complete the methods described below for this class.

```
public class Food {
  private int weight;
  private double calories;

  // Precondition: w > 0
  // Postcondition: Creates a food object with weight w and number of
  // calories set to 100w.
  public Food(int w) {
  }

  // Precondition: w > 0, c > 0
  // Postcondition: Creates a food object with weight w and number of
  // calories set to c.
  public Food(int w, int c) {
  }

  // Precondition: amount <= this.weight
  // Postcondition: Executes eating amount ounces of this food object,
  // adjusting the food object appropriately and returns the number
  // of calories consumed.
  public double eat(int amount) {
  }

  // Postcondition: returns a new food object created by combining
  // this food object with other.
  public Food combine(Food other) {
  }
}
```

5) (5 pts) Write a constructor for the Food class that creates a food object with weight w and 100w number of calories.

```
public Food(int w) {

    weight = w;                 // 2 pts
    calories = 100*w;           // 3 pts
}
```

6) (5 pts) Write a constructor for the Food class that creates a food object with weight w and 100w number of calories.

```
public Food(int w, int c) {

    weight = w;                 // 2 pts
    calories = c;               // 3 pts
}
```

7) (10 pts) Add a method eats to the Food class that adjust this food object so amount ounces of food were eaten from it (so both the object's weight and calories must be adjusted) and returns the number of calories that were consumed. Assume that the calories in the food are evenly distributed, so if half the food is eaten, so are half of its calories.

```
public double eat(int amount) {

     double ratio = (double)amount/weight;   // 2 pts
     double calLoss = ratio*calories;        // 2 pts
     weight -= amount;                       // 2 pts
     calories -= calLoss;                    // 2 pts
     return calLoss;                         // 2 pts

}
```

Grading: Note - only take off 1 pt total for int division.

8) (10 pts) Add a method combine to the Food class that takes the contents of this food object and the food object other and creates a new Food object that is essentially the "sum" of the two - the weight of the new object will be the sum of the weights of this and other and the number of calories will be the sum of the calories of this and other. Then, return this new object. Neither this nor other should be altered in any way.

```
public Food combine(Food other) {

     int totalW = this.weight + other.weight;
     int totalC = this.calories + other.calories;
     return new Food(totalW, totalC);


}
```

8) (12 pts) Complete the draw method shown below so that as long as it's properly invoked with all the appropriate support code, it will draw a sign similar to the red cross logo:

Make your creation two rectangles, each with dimensions 300 x 100, symmetrically placed on top of each other. The top left corner of each rectangle should be (100, 200) and (200, 100).

```
public void draw(Graphics g) {

    g.setColor(Color.WHITE);          // 0 pts
    g.fillRect(0, 0, 500, 500);       // 0 pts

    g.setColor(Color.RED);            // 2 pts
    g.fillRect(200, 100, 100, 300);   // 5 pts
    g.fillRect(100, 200, 300, 100);   // 5 pts


}
```

9) (4 pts) In the posted solution for the moving box, one of the methods has the following lines of code:

```
    int mult = r.nextInt(5) - 2;
    while (mult == 0)
        mult = r.nextInt(5) - 2;
```

What are the possible values of mult after this segment of code completes?

-2, -1, 1, 2 (Grading: 1 pt per correct item, -1 per incorrect item, cap at 0.)

10) (4 pts) In the posted solution for the moving box, the moveLeft() method is as follows

```
public void moveLeft() {
    positionX = (positionX - STEP + MAXX)%MAXX;
}
```

What is the purpose of the first occurrence of MAXX (the one inside the parentheses)? What behavior does this method enable, if the appropriate key is pressed?

It prevents the value in the parentheses from becoming negative, which, then prevents positionX from becoming negative. The behavior enabled is a wrap around from the left end of the screen to the right, or the right end of the screen to the left. (Grading: 2 pts for negative, 2 pts for wrap-around)

11) (4 pts) On the directions for the moving box assignment, the comments for the framework said, "Do not edit this method", referring to the draw method. Many of you edited this method to get your desired behavior. But, for the original version of the assignment, why was this note attached to the framework?

The scope of the original assignment was to have a single box on a white background, whose movement was completely specified by the key presses. In terms of the draw method, for this particular task, what should always be drawn is white on the whole screen and then the box at its current location. Since the draw method had access to information about the current location of the box, the draw method could easily be written without any knowledge of the key controls. Thus, to simplify the assignment for students, we wrote the draw code in its entirety, because we could and the full assignment could be completed by appropriately manipulating the instance variables of the class StudentWorkSolution in other places in the code.

Grading: 2 pts for recognizing that draw had all the info it needed, 2 pts for rest

12) (1 pt) What sound does Cap't Crunch cereal make when one eats it? **Crunch (Give to all)**

Graphics Class Methods

```
// Sets the color of this graphics object to color c.
setColor(Color c);

// Fills a rectangle with the current fill color with the top
// left corner at (topLeftX, topLeftY), with width width and
// a height of height. The x axis is horizontal and the y axis
// is vertical, with y coordinates increasing going down the
// screen. The top left corner of the display is (0, 0)
fillRect(int topLeftX, int topLeftY, int width, int height);
```

Color Constants

```
// Constant for the color red.
Color.RED
```

Math Class Methods

```
// Returns the square root of x (if x >= 0).
// Returns NaN otherwise.
sqrt(double x);

// Returns base raised to the power exp.
pow(double base, double exp)
```