MEGA Assignment - practice

These problems won’t be graded, but they will help you prepare for the final. They incorporate material that we’ve covered since the last midterm.

Programming problems

If it walks like a duck…

Taking our Animal.java and Fish.java and Bird.java classes (you can find them on the course site in the Code folder for Inheritance and Polymorphism part II), extend these classes by creating a Duck class that inherits from the Bird class and a Trout class that inherits from fish. Override their getSound() methods to say something appropriate for each.

Then, in a test class, create an Arraylist of 10 random selections of Ducks and Trout and print out what each says.

Cloud to Butt
Inspired by this amazing chrome extension here:

(https://chrome.google.com/webstore/detail/cloud-to-butt-plus/apmlngnhgbnjpajelfkmabhfapgnoai?hl=en)

that replaces occurrence of the words 'the cloud' with 'my butt' and 'cloud' with 'butt', write a program that reads in each line of text from this link: http://cs.nyu.edu/courses/spring16/CSCI-UA.0101-008/assignments/text/cloud_article.txt

and replace every occurrence 'cloud' with 'butt' (or any other word you’d like)

Write each line back out to a text file named butt_article.txt (or whatever word you choose)

---

**Out of Bounds!**

Write a program that:

- Creates an array of 100 randomly assigned integers (between 0-9)
- Asks the user for an index, then, in a method:
  - If the index is valid, display the value of the index to the user
  - If the index is invalid, throw an ArrayIndexOutOfBoundsException and tell the user what happened.
Pong Part 2 - ball

Continuing our Pong game, create a class PongBall that has the following properties:

Data fields:

- size
- x and y position
- Speed x and speed y of the ball
- direction for x and y, (1 or -1)
- references to the game and the paddle

Methods:

- A constructor where you pass a reference to the game PApplet and the paddle
- A draw method
  - that moves the ball to a new x and y position, taking into account the speed and direction
  - draws an ellipse for the ball
  - checks to see if it should bounce (method below)
- A method to check for bounce
  - if the ball hits the left or right side of the screen, the x direction should change
  - if the ball hits the top of the screen, the y direction should change
  - if the ball is down by the paddle, it should check to see if it’s touching the paddle and bounce, otherwise, it should fall off the screen

Create an instance of the PongBall class and run it in your game!

Triangle class

Given the class GeometricObject below, write a subclass Triangle with the following properties:

- Three doubles - side1, side2, and side3 with default values of 1.0 for the lengths of the sides
- A no-arg constructor that creates a default triangle
- A constructor that takes in 3 doubles for sides to create a triangle. Note: In triangles, the sum of any 2 sides is greater than the other side. the constructor should check to make sure that is the case and throw an IllegalArgumentException if you try to make a triangle that doesn’t follow the rule.
- Getters for all 3 sides
- A getArea() method that returns the area of the triangle
- A getPerimeter() method that returns the perimeter
- A toString() method that returns a string description of the triangle like: “Triangle: side 1 = 1.0, side 2 = 1.0, side 3 = 1.0"
Then write a test program that creates an ArrayList of 5 triangle objects. Set each of the following properties to **something other than the default:**

- three sides of the triangle
- the color
- the Boolean value to indicate whether the triangle is filled

You can enter these by hand, no need to ask the user. Wrap them in a try-catch block and make sure that the constructor exception is handled and tells the user if the rule is broken.

The program should then loop through the ArrayList and print out the sides, color and whether each Triangle is filled

```java
public class GeometricObject {
    private String color = "white";
    private boolean filled;
    private java.util.Date dateCreated;

    /** Construct a default geometric object */
    public GeometricObject(){
        dateCreated = new java.util.Date();
    }

    /** Construct a geometric object with the specified color
     * and filled value */
    public GeometricObject(String color, boolean filled){
        dateCreated = new java.util.Date();
        this.color = color;
        this.filled = filled;
    }

    /** Return color */
    public String getColor(){
        return color;
    }

    /** Set color */
    public void setColor(String color) {
        this.color = color;
    }
```
/** Return filled. Since filled is boolean, its getter method is named isFilled */
public boolean isFilled() {
    return filled;
}

/** Set a new filled */
public void setFilled(boolean filled) {
    this.filled = filled;
}

/** Get dateCreated */
public java.util.Date getDateCreated() {
    return dateCreated;
}

/** Return a string representation of this object */
public String toString() {
    return "created on " + dateCreated + "\ncolor: " + color + " and filled: " + filled;
}

Create a new class and copy that in.

Pong Part 3 - High Score! (This one can be pretty challenging)

Keeping our Pong example going, write a ScoreManager class that does the following:

- A method that will create (if it doesn’t exist) a score.txt file that will keep track the highest number of times a player has bounced the pong ball off the paddle. (just the number, saved to a file)
- A method that adds one to the current score (can be called from the pong game each time the ball bounces off the paddle)
- A method that reads the file (if it exists) and holds the current high score in a variable
- A method that checks if the current player’s score is higher than the high score and saves the new high score to the file
Then, within the Pong game class:

- Create an instance of the `ScoreManager` class
- Read or create the high score file (using the method above)
- Show (maybe in the corners) the current player’s score and the high score.
- Each time the ball bounces off the paddle, add one to the current score
- If the ball falls past the paddle, call the method in the `score` object that checks to see if the score is higher than the current high score, and update the file accordingly (using the method above)

**Note:**

- You’ll have to create and use PFont to display text.
- Don’t worry about the corners of the paddle adding too many points. You’ll see in the example gif that it ends up adding like 4 when it hits a corner. You can fix it if you’d like :)
- This problem will be challenging, but if you do it, you’ll have made your first complete game!