Tommy McDermott
10/5/2020
C212 Week 3 Lecture Qs

R3.10:
The balance in this case will simply go into the negatives.

E3.6 & 3.7:

```
package com.company;

public class BankAccount {
   private double balance;

   public BankAccount() {
       balance = 0;
   }

   public BankAccount(double initialBalance) {
       balance= initialBalance;
   }

   public void addInterest(double rate)
   {
      balance = balance + (balance*(rate*.01));
   }

   public void deposit(double amount)
   {
       balance = balance + amount;
   }

   public void withdraw(double amount)
   {
       balance = balance - amount;
   }

   public double getBalance()
   {
       return balance;
   }
}

public class Main extends BankAccount {

   public static void main(String[] args) {
  // write your code here
       BankAccount myAcc = new BankAccount();
       myAcc.deposit(1000);
       myAcc.withdraw(500);
       myAcc.withdraw(400);
       myAcc.addInterest(10);
       System.out.println(myAcc.getBalance());
   }
}
```

E3.13:

```
package com.company;

public class ProductPrinter{
   public static void main(String[] args)
   {
       Product toyCar = new Product("Toy Car",32.69);
       Product toyBoat = new Product("Toy Boat", 66.66);
       System.out.println(toyCar.getName()+", "+ toyCar.getPrice());
       System.out.println(toyBoat.getName()+", "+ toyBoat.getPrice());
       toyBoat.reducePrice(5.00);
       toyCar.reducePrice(5.00);
       System.out.println(toyCar.getName()+", "+ toyCar.getPrice());
       System.out.println(toyBoat.getName()+", "+ toyBoat.getPrice());
   }
}
public class Product extends ProductPrinter{
   private String name;
   private double price;

   public Product(String name, double price)
   {
       this.name = name;
       this.price = price;
   }

   public String getName()
   {
       return name;
   }

   public double getPrice()
   {
       return price;
   }

   public double reducePrice(double amt)
   {
       price = price - amt;
       return price;
   }
}
```

R3.15:
Since we can calculate *area* from the value *sideLength*, it would be pointless to store this data instead of writing the calculation. It would be like storing somebody's age when you already have their birthdate.

R3.16:
The error in this code exists in the fact that *area* never changes value, rather it is declared in the constructor and never altered again. If I were to fix this, in *getArea()* I would simply return *sideLength*sideLength* instead of *area*.

R3.11:
this refers to a certain variable in a class if it has the same title as that in its constructor. It can be used in a situation such as if there was a variable *y* in a constructor and a *private int y* in the class. Inside the constructor the programmer would write *this.y* = *y* to set that object's *y* equal to that under which it was declared.

R3.13:
The *mystery* method subtracts the balance of the bankAccount being dealt with, and then does the same to whatever bank account is called in the arguments.

E3.24:
```
package com.company;

import javax.swing.*;
import java.awt.*;
import java.awt.geom.Ellipse2D;

public class Olympic extends JComponent {
   public void paintComponent(Graphics g)
   {

       Graphics2D g2 = (Graphics2D) g;
       g2.setStroke(new BasicStroke(2));
       g2.setColor(Color.BLUE);
       g2.drawOval(40,40,75,75);
       g2.setColor(Color.YELLOW);
       g2.drawOval(80, 80, 75,75);
       g2.setColor(Color.BLACK);
       g2.drawOval(120,40,75,75);
       g2.setColor(Color.GREEN);
       g2.drawOval(160, 80, 75,75);
       g2.setColor(Color.RED);
       g2.drawOval(200, 40, 75,75);
   }
}
import javax.swing.*;

public class OlympTest extends JComponent{
   public static void main(String[] args)
   {
       JFrame fr = new JFrame();
       fr.setSize(500,500);
       fr.setTitle("Olympia");
       fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        Olympic olym = new Olympic();
        fr.add(olym);

        fr.setVisible(true);
    }
}
```