CSCI 1101 – Winter 2017
Laboratory No. 3

SOLUTIONS (Note: Only solution outlines for relevant exercises are given here)

Exercise 1(b):

```
public class TurnTakerDemo3b
{
      public static void main(String[] args)
      {


            for(int i = 1; i<= 5; i++)
            {

                  TurnTaker person1 = new TurnTaker("Romeo", i);
                  TurnTaker person2 = new TurnTaker("Juliet", i);
                  System.out.println("Turn = " + TurnTaker.getTurn());
                  if (person1.isMyTurn())
                        System.out.println("Love from " + person1.getName());
                  if (person2.isMyTurn())
                        System.out.println("Love from " + person2.getName());
            }
      }
}
```

Exercise 1(c): Modify the above appropriately.

Exercise 2:

```
//PizzaStand.java
public class PizzaStand
{
      private int ID;
      private int num;
      private static int total=0;
      private static double cost;

      public PizzaStand(int n)
      {
            ID = n;
            num = 0;
      }
      public static void setCost(double c)
      {
            cost = c;
      }
      public void justSold()
      {
            num++;
            total++;
      }
      public int getNumSold()
      {
            return num;
      }
      public static int getTotal()
      {
            return total;
      }
      public static double getSales()
      {
            return (total*cost);
      }
      public String toString()
      {
            return (ID + "\t" + num + "\n");
      }
}

//PizzaDemo.java
public class PizzaDemo
{
      public static void main(String[] args)
      {
            PizzaStand.setCost(5.00);
            PizzaStand One = new PizzaStand(1);
            PizzaStand Two = new PizzaStand(2);
            PizzaStand Three = new PizzaStand(3);
            PizzaStand Four = new PizzaStand(4);
            PizzaStand Five = new PizzaStand(5);
            One.justSold();
            Two.justSold();
            Three.justSold();
            Four.justSold();
            Five.justSold();
            One.justSold();
            System.out.println("Pizza Sales: ");
            System.out.println(One +"\n" +  Two + "\n" + Three +"\n" +
Four+"\n" + Five);
            System.out.println("Total pizzas sold: " + PizzaStand.getTotal());
            System.out.println("Total sales: " + PizzaStand.getSales());
      }
}
```

Exercise 3:

```
   public class MyInteger
   {
      private int value;

      public MyInteger(int value)
      {
         this.value = value;
      }

      public int getValue()
      {
         return value;
      }

      public static boolean isEven(int n)
```

```
 {
    return (n%2==0);
 }
 public static boolean isOdd(int n)
 {
    return (n%2!=0);
 }
 public static boolean isPrime(int n)
 {
    for (int i=2; i<n/2;i++)
       if (n%i==0)
          return false;
    return true;
}
 public boolean isEven()
 {
    return isEven(this.getValue());
 }
 public boolean isOdd()
 {
    return isOdd(this.getValue());
 }
 public boolean isPrime()
 {
    return isPrime(this.getValue());
 }
 public static boolean isEven(MyInteger m)
 {
    return m.isEven();
 }
 public static boolean isOdd(MyInteger m)
 {
    return m.isOdd();
 }
 public static boolean isPrime(MyInteger m)
 {
    return m.isPrime();
 }
 public boolean equals(int n)
 {
   return (n==this.value);
 }
 public boolean equals(MyInteger m)
 {
   return (m.equals(this.value));
 }
 public static int parseInt(char[] s)
 {
    int result=0;
    for(int i=0;i<s.length;i++)
       result = result*10+((int)s[i]-48);

   return result;
 }
 public static int parseInt(String s)
 {
    int result = 0;
```

```
      for(int i=0; i<s.length();i++)
         result = result*10+((int)s.charAt(i)-48);
      return result;
   }
}
```

Exercise 4:

```
//Algorithm: Find the distance between the two centers. Add the radii.If
//the result is the same,
   //then the circles touch externally.
   //The code uses distance squared and sum of radii squared
   public boolean touchesExternally(Circle other)
   {
      double x = (cx - other.getCX());
      double y = (cx - other.getCY());
      double distsqrd = (x*x)+(y*y);
      if (distsqrd ==
(radius+other.getRadius())*(radius+other.getRadius()))
         return true;
      else
         return false;
   }

   //Method to check if this Circle object touches another Circle object
//internally
   //Algorithm: Find the distance between the two centers. Subtract the
//radii.If the result is the same,
   //then the circles touch internally.
   //The code uses distance squared and difference of radii squared
   public boolean touchesInternally(Circle other)
   {
      double x = (cx - other.getCX());
      double y = (cx - other.getCY());
      double distsqrd = (x*x)+(y*y);
      if (distsqrd == (radius-other.getRadius())*(radius-
other.getRadius()))
         return true;
      else
         return false;
   }
```