Dwayne Fraser COP 4331 003
Homework # 1
Problem # 1

```
/*
DWAYNE FRASER
HOMEWORK 1.1
 */

package q1;

public class Fib{

int f0;
int f1;


  // constructor
   public Fib(int f0, int f1){

       //'THIS' Keyword used for differentiating instance variable and local variable.
       this.f0=f0;
       this.f1=f1;
   }

   public int f(int n){
   /**
   ** computes F(n) using an ***iterative*** algorithm, where F(n) = F(n-1) + F(n-2) is the
recursive definition.
   ** use instance variables that store F(0) and F(1).
   ** check parameter and throw exception if n < 0. Don't worry about arithmetic overflow.
   */
       int temp1 = f0;
       int temp2;
           int sum = f1;

           if (n < 0) {
               throw new IllegalArgumentException();
           }
           if (n == 0)
               return f0;
           if (n == 1)
               return f1;
```

```
            for (int i=2; i<=n; i++) {
                temp2 = temp1;
                temp1 = sum;
                sum = temp1 + temp2;
            }

            return sum;
   }

   // computes F(n) using the ***recursive*** algorithm, where F(n) = F(n-1) + F(n-2) is the recursive definition.
   // use instance variables that store F(0) and F(1).
   // check parameter and throw exception if n < 0. Don't worry about arithmetic overflow.

    public int fRec(int n)
    {
            if (n < 0) {
                throw new IllegalArgumentException("wrong parameter n=" + n);
            }
            if (n == 0)
                return f0;
            if (n == 1)
                return f1;

            return fRec(n-1) + fRec(n-2);
    }
}

/*
DWAYNE FRASER
HOMEWORK 1.1
 */

package q1;

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

        try {
            // numbers F(0) and F(1) from args[0] and args[1].

            int f0=Integer.parseInt(args[0]);

            int f1=Integer.parseInt(args[1]);
```

```
         // n from args[2]:
         int n = Integer.parseInt(args[2]);

         // a Fib object with params F(0) and F(1)
         Fib fibobj = new Fib(f0,f1);

         // calculating F(0), ..., F(n) and displaying them using the iterative method f(i)
         System.out.println("iterative below");
            for (int i=0; i<=n; i++) {
            int fib = fibobj.f(i);
            System.out.println(fib);
            }

         // calculating F(0), ..., F(n) and displaying them using the recursive method fRec(i)
         System.out.println("recursive below");
            for (int i=0; i<=n; i++) {
            int fib = fibobj.fRec(i);
            System.out.println(fib);
            }

      }catch(NumberFormatException e){
         System.out.println("The argument must be an integer.");
         System.exit(1);
      }

   };
}
```

Problem #2

```
/*
DWAYNE FRASER
HOMEWORK 1.2
 */

package q2;

public class Greeter {

// constructor
  public Greeter(String aName)
  {
     name = aName;
  }
  /**
       Greet with a "Hello" message.
       @return a message containing "Hello" and the name of
       the greeted person or entity.
  */
  public String sayHello()
  {
     return "Hello, " + name + "!";
  }

   /**
    * swaps the names of this greeter and another instance
    * @param other greeter object
   */
  public void swapNames(Greeter other)
  {
     String temp;

     //Swapping names using temporary variable
     temp = this.name;
     this.name = other.name;
     other.name = temp;
  }

   /**
    * returns a new Greeter object with its name being the qualifier string
    * followed by" " and the executing greeter's name
    * @param qualifier object
    * @return new greeter object
   */
  public Greeter createQualifiedGreeter(String qualifier)
  {
      String temp;

      temp = qualifier + " " + this.name;

      return new Greeter(temp);
  }

  private String name;
}
```

```
/*
DWAYNE FRASER
HOMEWORK 1.2
 */

package q2;
/**
  Greeter Test Class
*/
public class GreeterTester {

  public static void main(String[] args)
  {

    Greeter GreetObj1 = new Greeter("HELLO");
    Greeter GreetObj2 = new Greeter("WORLD");


    String greet1 = GreetObj1.sayHello();
    String greet2 = GreetObj2.sayHello();


    System.out.println(greet1 + greet2);


    GreetObj1.swapNames(GreetObj2);

    greet1 = GreetObj1.sayHello();
    greet2 = GreetObj2.sayHello();


    System.out.println(greet1 + greet2);

    Greeter GreetObj3 = new Greeter("WORLD");

    Greeter GreetObj4 = GreetObj3.createQualifiedGreeter("BEAUTIFUL");

    String greet3 = GreetObj4.sayHello();

    System.out.println(greet3);
  }
}
```

Problem # 3

```
/*
DWAYNE FRASER
HOMEWORK 1.3
 */

package q3;

import java.util.*;

/**
  Data Analyzer Class
  *
  * Computes Min, Max, & Avg
*/
public class DataAnalyzer
{
  /**
     Constructor
     * @param numbers
  */
  public DataAnalyzer(LinkedList<Integer> numbers)
  {
     //Storing numbers
     this.numbers = numbers;
  }

  /**
     @return Max
  */
  public int max()
  {
     int max = numbers.get(0);

     for(int i=0; i < numbers.size(); i++)
     {
        if( numbers.get(i) > max )
           max = numbers.get(i);
     }

     return max;
  }

  /**
     @return Min
```

```
    */
    public int min()
    {
        int minNum = numbers.get(0);

        for(int i=0; i < numbers.size(); i++)
        {
            if( numbers.get(i) < minNum )
                minNum = numbers.get(i);
        }

        return minNum;
    }

    /**
        @return Avg
    */
    public int avg()
    {
        int sum = 0;
        int avg;

        for(int i=0; i < numbers.size(); i++)
        {
            sum += numbers.get(i);
        }

        avg = sum / numbers.size();

        return avg;
    }

    private final LinkedList<Integer> numbers;
}

/*
DWAYNE FRASER
HOMEWORK 1.3
 */

package q3;

import java.util.*;
import java.io.*;
```

```
/**
  Data Analyzer Test Class
*/
class DataAnalyzerTester
{

  public static void main(String args[])
  {
    try
    {

      LinkedList<Integer> numbers_list = new LinkedList<>();
      int number;

      Scanner scan = new Scanner(System.in);

      String fileName;

      System.out.println("\n Enter file name: ");
      fileName = scan.nextLine();

      File file = new File(fileName);

      file.createNewFile();

      try (

            FileWriter writer = new FileWriter(file))
      {
            System.out.println("Enter Integer Value: (0 to exit) \n");

            number = scan.nextInt();


            do{

            writer.write(number + " \n ");

            numbers_list.add(number);

            number = scan.nextInt();
            }while(number != 0);
```

```
                    DataAnalyzer analyzer = new DataAnalyzer(numbers_list);

                    System.out.println("Min: " + analyzer.min());
                    writer.write(analyzer.min());
                    System.out.println("Max: " + analyzer.max());
                    writer.write(analyzer.max());
                    System.out.println("Avg: " + analyzer.avg());
                    writer.write(analyzer.avg());
                    writer.flush();

                }
                catch(IOException ex)
                  {
                      System.out.println(ex);
                  }
       }

       catch(IOException ex)
       {
           System.out.println(ex);
       }
   }
}
```

Problem #4

Without running the code, the value of is will be 11. The greeter objects g1 and g2 will not be equal, therefore g2 is null. The first exception block will be called and x will be 10 and then x will increment leaving x = 11.

Problem # 5

```
/*
DWAYNE FRASER
HOMEWORK 1.5
 */

package q5;

import java.util.ArrayList;
import java.util.List;

public class PrimeFactorizer {
```

```
//Constructor
public PrimeFactorizer(int n){
    this.n = n;

    factors = new ArrayList<>();
    exponents = new ArrayList<>();
}

public int getN(){
    return this.n;
}

public void compute(){

    if(!factors.isEmpty()){
        return;
    }
    int x = n;
    int c = 0;

    while (x%2==0){
        x/=2;
        c++;
    }
    if(c!=0){
        factors.add(2);
        exponents.add(c);
    }

    for(int i=3;i<=Math.sqrt(n);i+=2){
        c = 0;

        while (x%i==0){
            x/=i;
            c++;
        }

        if(c>0){
            factors.add(i);
            exponents.add(c);
        }
        }
        if(x!=1){
            factors.add(x);
            exponents.add(1);
        }
    }

        public void getFactorsAndExponents(int n, ArrayList<Integer> primes,
ArrayList<Integer> exponents){

        primes.clear();
        primes.addAll(factors);
        exponents.clear();
        exponents.addAll(this.exponents);
    }

    @Override
    public String toString(){

        String ans = "";
        for(int i=0;i<factors.size();i++){
            ans = ans+factors.get(i)+"^"+exponents.get(i)+"*";
        }
        return ans.substring(0,ans.length()-1);
    }

    private final int n;
    private final List<Integer> factors;
    private final List<Integer> exponents;

}

/*
DWAYNE FRASER
HOMEWORK 1.5
 */

package q5;
/**
   Prime Factor Test Class
*/
public class PrimeFactorTest {

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

```
        PrimeFactorizer primeFactorizerObj = new PrimeFactorizer(16);

        primeFactorizerObj.compute();

        System.out.println("The primes of 16 is: " + primeFactorizerObj);

        PrimeFactorizer primeFactorizerObj2 = new PrimeFactorizer(81);

        primeFactorizerObj2.compute();

        System.out.println("The primes of 81 is: " + primeFactorizerObj2);
    }
}
```