ECU178 Computer Science: 210CT - Programming, Algorithms and Data Structures Portfolio

Due on Monday, December 15th, 2014

Dr James Shuttleworth

Robert Rigler : 4939377

Contents

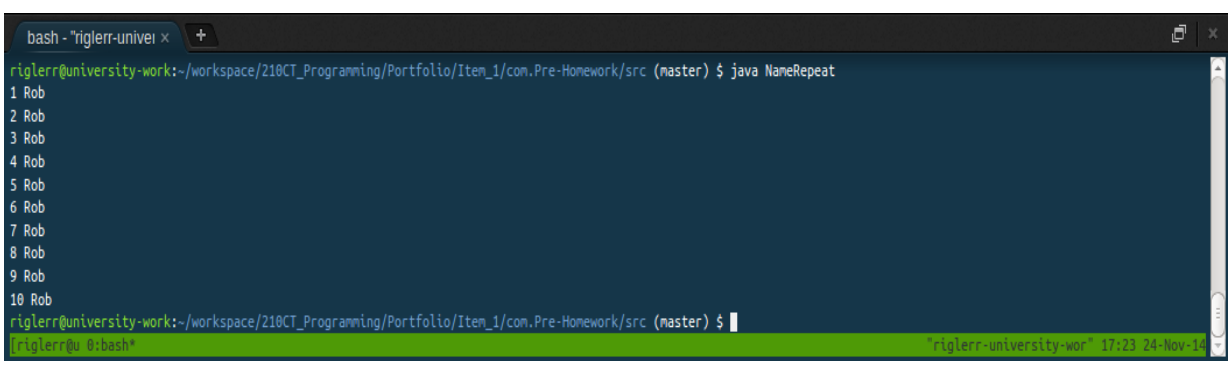
Item 1: Week 3 - Linear Search and Duplicate Finder

Pre-Homework 1: Write a Program that displays your name 10 times

Listing 1: NameReapeat class JAVA code

```java
/**
 * Created by Rob on 23/10/2014.
 */
public class NameRepeat {

    public static void main(String[] args){

        NameRepeat myObject = new NameRepeat();  /*Create Object*/
        myObject.PrintName("Rob");      /*Use object to call PrintName() metho
    }

    public void PrintName(String _name){

        for (int i = 0; i<10;i++){  /* Loop 10 times*/
            System.out.println((i+1) + " " + _name);
          /*, print the number && _name parameter each time.*/

        }


    }
}
```

Evidence

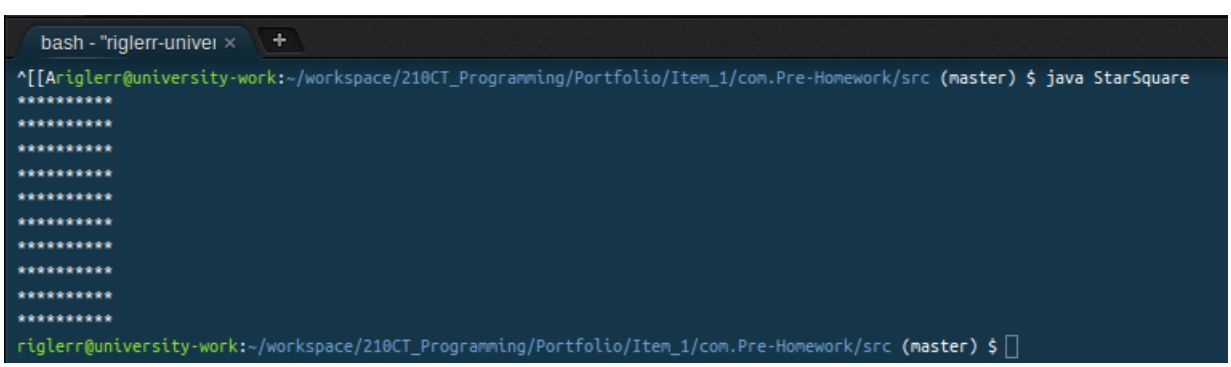


Pre-Homework 2: Write a function that draws a square of stars given as a parameter

Listing 2: StarSquare class JAVA code

```java
/**
 * Created by Rob on 23/10/2014.
 */
public class StarSquare {

    public char ast = '*';  //Create variable to hold asterisk character.
    public static void main(String[] args){

        StarSquare sSquare = new StarSquare(); /*Create Object*/
        sSquare.writeSquare(10); /*Use object to call writeSquare() method*/

    }


    public void writeSquare(int size){

        for(int i =0; i<size;i++){ /*OuterLoop 'size' times*/
            for(int j = 0;j<size;j++){ /*InnerLoop 'size' times*/

                System.out.print(ast); /*Print line of asterisks*/

            }
            System.out.println(); /* Start new line when inner loop finishes*/
        }
    }
}
```

Evidence

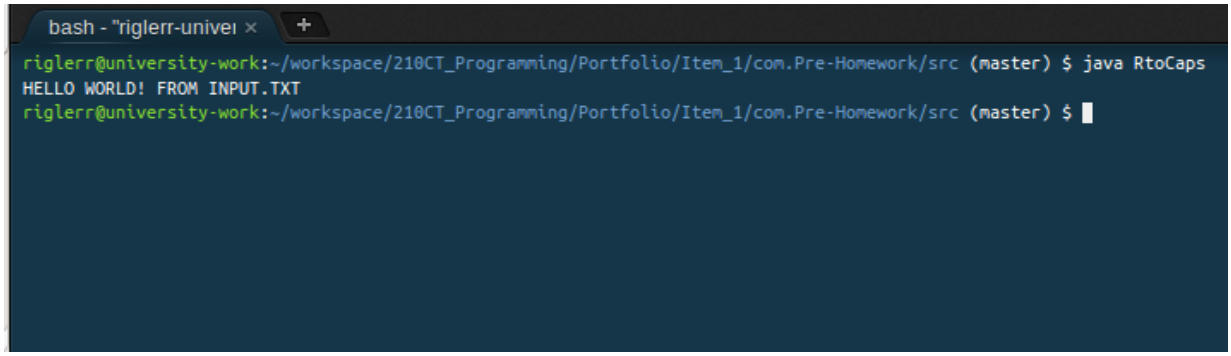


Pre-Homework 3: Write a program to open a file and display it's contents in capitals

Listing 3: RtoCaps class JAVA code

```java
/**
 * Created by Rob on 23/10/2014.
 */
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class RtoCaps {

    public static void main(String[] args)throws FileNotFoundException {
        File inFile = new File("input.txt");
                /*Create a file object */
        RtoCaps obj = new RtoCaps(); /*Create class object*/
        obj.rInput(inFile); /*Use Class object to call rInput() method*/
    }

    public void rInput(File inFile) throws FileNotFoundException{

     /*Create a new scanner to read from the file*/
        Scanner in = new Scanner(inFile);

     /*Loop WHile there is still lines left in the document*/
        while(in.hasNextLine())
        {
          /* Place the next line in a strin varibale*/
            String line = in.nextLine();

           /* Print the line in uppercase*/
            System.out.println(line.toUpperCase());
        }

    }


}
```

Evidence



1. Pseudocode for linear search

Algorithm 1 LinearSearch

```
procedure BOOL LINEARSEARCH(item, list[ ])
    for each element i in list do
        if list[i] = list then
            return true
        end if
    end for
    return false
end procedure
```

2. Pseudocode for finding duplicates in a list

Algorithm 2 Examining for duplicates

```
procedure BOOL EXFORDUPES(list[ ])
    for each element i in list[ ] do
        for each element j in list[ ] do
            if list[i] = list[j] then
                return true
            end if
        end for
    end for
end procedure
```

continued on next page...

Item 2: Week 4 - Time complexities and Big-O notation

1. Describe the runtime bounds of the linear search algorithm

Algorithm 3 LinearSearch

```
procedure BOOL LINEARSEARCH(item, list[ ])

    for each element i in list do        (n)
        if  list[i] = list then t        (n)
            return true                  (n)
        end if
    end for
     return false                        (1)
end procedure
```

The time complexity of the algorithm is O(n)

2. Describe the runtime bounds of the duplicate finder algorithm

Algorithm 4 Examining for duplicates

```
procedure BOOL EXFORDUPES(list[ ])
    for  each element i in list[ ] do        (n)
        for  each element j in list[ ] do    (n*n)
            if  list[i] = list[j] then       (n*n)
                return true                  (n*n)
            end if
        end for
    end for
     return false                            (1)
end procedure
```

The time complexity of the algorithm is O(n^2)

Additional work: Critical values of relative runtimes

Write a function that determines the critical value at which the relative runtime of two linear algorithms swap.

Algorithm 5 Relative runtime comparison algorithm

```
procedure CRITVAL(m1, k1, m2, k2)
    switch ← false
    n ← 0
    if (((m1 * n) + k1) > ((m2 * n) + k2)) then
        while !switch do
            if (((m1 * n) + k1) < ((m2 * n) + k2)) then
                switch ← true
            else
                n++
            end if
        end while
    else
        while !switch do
            if (((m1 * n) + k1) > ((m2 * n) + k2)) then
                switch ← true
            else
                n++
            end if
        end while
    end if
     return n
end procedure
```

Item 3: Week 6 - Harmonic Series or Pivot Selection

Item 4: Week 7 - Heapworksheet or RPN Calculator

Item 5: Week 8 - Linked List Delete function or Linked List Sortings