CS 120 Project 4

Due 11:59 PM Friday March 10

Project 4

- In this project you will complete the program shown on the following slides. The program takes line items for an invoice, checks the format of each line item and if the line items are correctly formatted, calculates the total cost for each line item.

- To complete the program you must implement the methods checkIdNum, checkName, checkCount and checkCost. In the methods that you must complete you must only use the function isDigit (which I gave you) and the string functions length and charAt. You must not use any other string functions or any other classes such as Scanner to look at the string. You can create other helper methods but these must also only use the functions isDigit, length and charAt.

- You must not change any other part of the program.

Project 4: line item format

- idNumber:name:count:unitCost
- An id number begins with a # followed by 1 or more digits
- A name begins with a letter (upper or lower case) followed by 0 or more letters (upper or lower case) or spaces
- A count is a sequence of 1 or more digits
- A unit cost is a sequence of 1 or more digits followed by a decimal point followed by exactly 2 digits

Project 4: Example Input

The first line indicates the number of line items

```
8
#12345:Drill:1:50.89
#900327:Hammer:3:10.50
#8124:Large Bucket:2:27.99
#12a7:Nails:100:8.49
#769347:Drafting Table:1:235.88
#532::3:10.37
#22713:Crowbar:1:
#59176:Filing Cabinet:2:50.49
```

Project 4: Example Output (Input shown in bold)

```
8
#12345:Drill:1:50.89
#900327:Hammer:3:10.50
#8124:Large Bucket:2:27.99
#12a7:Nails:100:8.49
#769347:Drafting Table:1:235.88
#532::3:10.37
#22713:Crowbar:1:
#59176:Filing Cabinet:2:50.49


      Id Number                Name     Count  Total Cost
         #12345               Drill         1       50.89
        #900327              Hammer         3       31.50
          #8124        Large Bucket         2       55.98
        #769347      Drafting Table         1      235.88
         #59176      Filing Cabinet         2      100.98


The following line items contain errors
#12a7:Nails:100:8.49
#532::3:10.37
#22713:Crowbar:1:
```

Project 4

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

public class MakeInvoice {

    public static boolean isDigit(char c) {
    //returns true if c is a digit otherwise return false
        return c >= '0' && c <= '9';
    }

    public static int checkIdNum(String s) {
    //if the id number is correctly formatted return the position of the colon marking the end of the id number field otherwise return -1

     }

    public static int checkName(String s, int i) {
    //i is the first position in s of the name field
     //if the name is correctly formatted return the position of the colon marking the end of the name field otherwise return -1
    }
```

Project 4

```
public static int checkCount(String s, int i) {
//i is the first position in s of the count field
//if the count is correctly formatted return the position of the colon marking the end of the count field otherwise return -1

}

public static boolean checkCost(String s, int i) {
//i is the first position in s of the cost field
//if the cost is correctly formatted return true otherwise return false

}
```

Project 4

```
public static boolean formatCheck(String s) {
//returns true if s is a correctly formatted line item otherwise return false
//DO NOT CHANGE THIS METHOD
    int i;
    if ((i = checkIdNum(s)) == -1) return false;
    if ((i = checkName(s, i+1)) == -1) return false;
    if ((i = checkCount(s,i+1)) == -1) return false;
    if (!checkCost(s,i+1)) return false;
    return true;
}
```

Project 4

```
public static String formatItem(String lineItem){
//formats a syntactically correct line item for output
//DO NOT CHANGE THIS METHOD
    Scanner s = new Scanner(lineItem);
    s.useDelimiter(":");
    String idNum = s.next();
    String name = s.next();
    int count = s.nextInt();
    double cost = s.nextDouble();
    return String.format("%15s%20s%10d%12.2f\n", idNum,name, count, count*cost);
}
```

Project 4

```
public static void main(String args[]) {
//the main method for the program
//this method reads the input, checks the format of each line item (using helper methods)
//and produces two output strings. One string for correctly formatted line items and
//one string for line items with errors.
//DO NOT CHANGE THIS METHOD
    int count;
    String lineItem;
    String lineItems = "";
    String errorItems = "";

    Scanner s = new Scanner(System.in);
    count = s.nextInt();s.nextLine();
    while (count > 0 ) {
        lineItem = s.nextLine();
        if (formatCheck(lineItem)) {
            lineItems = lineItems + formatItem(lineItem);
        } else {
            errorItems = errorItems+lineItem+'\n';
        }
        count = count - 1;
    }
```

Project 4

```
//slide two of the main method
//this part of the main method prints the results
//DO NOT CHANGE THIS CODE
        if (!lineItems.equals("")) {
            System.out.printf("\n\n%15s%20s%10s%12s\n%s", "Id Number",
            "Name", "Count", "Total Cost", lineItems);
        }
        if (!errorItems.equals("")) {
            System.out.printf("\n\nThe following line items contain errors\n%s",
                errorItems);
        }
    }
}    }
```

Project 4 Requirements

- You must write the code yourself

- The name of the class must be **MakeInvoice** and the file containing your program must be called **MakeInvoice.java.** Remember java is case sensitive.

- Include a comment describing the purpose of the program. The comment should include your name.

- You can ask questions of your classmates but you **MUST NOT** share code. If you have a question it is usually better to ask me.

- Use meaningful **variable names and proper indentation**.

Project 4 Submission

- Upload one zip file to Canvas. The zip file must contain only one file called MakeInvoice.java.

- The project is worth 25 points. The project must be submitted on time.

-