**CS230 Assignment 1**

**Author:** Diana Eastman

**Date Submitted:** February 4, 2013

**Collaborators:** None.

**Notes:**

1. No known bugs.

2. The programs work as indicated in the comments.

3. To demonstrate that the intended functionality has been achieved, I have included a soft copy of testing typescript.

4. **Problem 1** – By design, I have restricted the types of numbers that can come up for the item's cost on the backend to prevent costs from coming up that would not make sense to a user (i.e., a negative cost or a cost of $0). For testing purposes, however, I hard-coded additional values directly into the cost variable; these values have been commented out.

5. **References** - http://www.dr-bill.net/CSC012/class\_summaries/5-6/

http://docs.oracle.com/javase/1.4.2/docs/api/java/io/FileWriter.html

/\*

CorrectChange.java

Purpose: Output the minimum, correct amount of change given a randomly generated cost

CS230 Homework Assignment 1 Problem 1

Written by: Diana Eastman

Modified date: February 1, 2013

\*/

import java.util.\*;

import java.io.\*;

public class CorrectChange {

public static void main (String[] args) {

// Declarations and setup

Scanner sc = new Scanner(System.in);

Random rand = new Random();

int [] currencyArray = {1000, 500, 100, 25, 10, 5, 1}; // Expressed in cents: $10, $5, $1, $0.25, $0.10, $0.01

String [][] currencyNames = { {"Hamilton","Hamiltons"}, {"Lincoln", "Lincolns"}, {"Washington", "Washingtons"},

{"Quarter", "Quarters"}, {"Dime", "Dimes"}, {"Nickel", "Nickels"}, {"Penny", "Pennies"} };

int [] changeArray = new int[7]; // Empty array to store change of each currency type

// Generate a random integer for the item's cost, up to $1000.

int cost = rand.nextInt(100000); // Numbers for testing: -20, 0, 4394832

double paid;

int change;

// Validate the user's input and stop execution if the following criteria are NOT met:

// 1. The input must be an integer

// 2. The input must be greater than or equal to the item cost (this compensates for possible negative change)

System.out.println("Welcome to the Wellesley College vending machine! The cost of your item is $"

+ (double)cost/100 + ".");

do {

System.out.println("Please enter enough money to pay for this item.");

while (!sc.hasNextDouble()) {

System.out.println("Sorry, this doesn't look like a valid number. Please try again.");

sc.next();

}

paid = sc.nextDouble();

} while (paid\*100 < cost);

// If above criteria are met, calculate the change owed and print

change = (int)(paid\*100) - cost;

System.out.println("You paid $" + paid + "." + "Your change is $" + (double)(change)/100);

// If change equals = mod of change divided by the particular currency, the currency is > than the change and 0

// is inserted into the changeArray

for (int i = 0; i < currencyArray.length; i++) {

if (change == change % currencyArray[i]) {

changeArray[i]=0;

}

// Otherwise, we traverse the array and determine the number of each currency denomination needed

else {

changeArray[i]=change/currencyArray[i]; // Put appropriate number in changeArray

change=change-currencyArray[i]\*changeArray[i];// Update number with the appropriate subtraction

System.out.println(changeArray[i]+ " " + ((changeArray[i] == 1) ? currencyNames[i][0] : currencyNames[i][1]));

}

}

// BEGIN EXTRA CREDIT SECTION

// Create a new array to store the five image names

String [] imagePaths = {"ten.jpeg", "five.jpeg", "one.jpeg", "quarter.jpg", "dime.jpg", "nickel.jpg", "penny.jpg"};

// Use StringBuilder class to build out the skeleton of an HTML file

StringBuilder sb = new StringBuilder();

sb.append("<!DOCTYPE html>");

sb.append("<head>");

sb.append("<title>Welcome to the Wellesley College Vending Machine");

sb.append("</title>");

sb.append("</head>");

sb.append("<body>");

sb.append("<img src='http://new.wellesley.edu/sites/default/files/assets/images/wellesleycollege\_logo.jpg' style='width: 150px; display: inline;'>");

sb.append("<h1 style='font-family:Tahoma, san-serif; display: inline;'>Vending Machine</h1>");

sb.append("<div>");

sb.append("<h2 style='font-family:Tahoma, san-serif;'>Here's Your Change:</h2>");

// Use nested for-loop to:

// 1. Traverse change array

// 2. Print an image source tag for the number of each currency

for (int i = 0; i <changeArray.length; i++){

for (int j = 0; j<changeArray[i]; j++) {

sb.append("<img src='http://cs.wellesley.edu/~cs230/Other/"+imagePaths[i]+"'>");

}

}

sb.append("</div>");

sb.append("</body>");

sb.append("</html>");

// Because we are using an I/O object, we must check for possible I/O exceptions

try {

// Use FileWriter class to open new html file

FileWriter fstream = new FileWriter("change.html");

BufferedWriter out = new BufferedWriter(fstream);

out.write(sb.toString());

out.close();

}

catch (IOException e){

}

} //End main

}//End class CorrectChange

/\* Username.java

Purpose: A program that generates Wellesley-like (first letter of first name,

followed by no more than 7 letters of last name, all in lowercase) user names,

given a user's first name and last name.

CS230 Homework Assignment 1 Problem 2

Modified by: Diana Eastman

Modified date: February 3, 2013

\*/

import java.util.\*;

public class Username {

private String firstname;

private String lastname;

private String fcname; // the firstclass-like username

final int MAX = 8;

//-----------------------------------------------------------------

// Constructor. Sets the f and l as firstname and lastname,

// and computes the fcname with the first character of f

// and up to the first 7 characters of l

//-----------------------------------------------------------------

// Validate our input to make sure a string is actually entered for the first and last name

// to avoid error. This just checks that the string is not null, an empty space, or a string

// with numbers mixed in (especially important in the later part of the assignment when we generate

// usernames that will have numbers to avoid dups); the check could be more robust to prohibit other special chars.

public Username(String f, String l) {

if (f == "" || f == " " || f.matches(".\*\\d.\*")) {

throw new IllegalArgumentException("F can't be null, an empty string, or contain numeric values.");

}

if (l == "" || l == " " || l.matches(".\*\\d.\*")){

throw new IllegalArgumentException("L can't be null, an empty string, or contain numeric values.");

}

//Strip trailing and leading whitespaces, and remove dashes--if any--before taking substrings and concatenating

firstname = f.trim();

lastname = l.trim().replaceAll("-","");

String toSevenCharacters = lastname.substring(0, Math.min(lastname.length(), 7));

fcname = (firstname.substring(0,1)).toLowerCase()+toSevenCharacters.toLowerCase();

}

//-----------------------------------------------------------------

// Returns a string description of the Username.

//-----------------------------------------------------------------

public String toString()

{

return (firstname + " " + lastname + " " + fcname);

}

//-----------------------------------------------------------------

// Checks for equality between this and another Username and

// returns boolean

//-----------------------------------------------------------------

public boolean equals (Username another){

if (fcname.equals(another.fcname)) {

return true;

}

else {

return false;

}

}

public static void main (String[] args) {

Username u1 = new Username ("Kathy", "Lee"); System.out.println(u1);

Username u2 = new Username ("Jennifer", "Anniston"); System.out.println(u2);

if (!u2.equals(u1)) System.out.println("u2 and u1 do not have the same fcname");

Username u3 = new Username ("Kate", "Lee"); System.out.println(u3);

if (u3.equals(u1)) System.out.println("u3 and u1 have the same fcname");

Username u4 = new Username ("Jane", "Anniston"); System.out.println(u4);

if (u4.equals(u2)) System.out.println("u4 and u2 have the same fcname");

// Additional tests

Username u5 = new Username ("JOANNE", "KANG"); System.out.println(u5);

Username u6 = new Username ("joanne", "kang"); System.out.println(u6);

if (u5.equals(u6)) System.out.println("u5 and u6 have the same fcname");

Username u7 = new Username ("Kathy", "Lee "); System.out.println(u7); //Add a trailing blank

Username u8 = new Username (" Kathy", "Lee "); System.out.println(u8); //Add leading blank to fname and trailing to lname

if (u7.equals(u1)) System.out.println("u1 and u7 have the same fcname");

if (u8.equals(u1)) System.out.println("u1 and u8 have the same fcname");

Username u9 = new Username ("Andrea", "Java-Bean"); System.out.println(u9);

// Tests to throw exception (nothing prints in console -- can see exception in interaction pane)

Username u10 = new Username ("", "Lee"); System.out.println(u10);

Username u11 = new Username (" ", "Lee"); System.out.println(u11);

Username u12 = new Username ("Kathy", "Lee8"); System.out.println(u12);

}

}