**CSC142, Computer Science II, Project 4 assignment**

Answer all the questions and submit both document and java files to D2L. Later submission is not accepted.

# Tracing Code with Strings

(Show what is stored in memory at the end of each of these programs)

public class String-CharAt {

public static void main(String [] args) {

String s = “hello”;

char c = s.charAt(2);

c++; //char +, +1, ++

s = “he” + c + c + “o”; //String +

}

}

c, s?

c

s

?

?

public class String-Substring {

public static void main(String [] args) {

String t = “hee dee hee dee hee dee hee”;

int x = t.indexOf(“dee”); // indexOf(str)

String u = t.substring(0,x); // substring(x, y)

u = u + “haw”;

x = t.indexOf(“dee”); //second call of indexOf(str)

}

}

u, x?

public class String-IndexI {

public static void main(String [] args) {

String s = “hi dee hi dee hi”; // find four blank space first!

String t = “”;

int x = s.indexOf(“hi dee”, 0); // indexOf(str, x)

while(x>=0) {

t = t + “ho dee”;

x = s.indexOf(“hi dee”, x+1); // indexOf(str, x)

}

t = t + “ho”;

}

}

t, x?

public class String-Equals {

public static void main(String [] args) {

String s = “hibbity” +

“hibbity”;

int i=0;

int count = 0;

while(i<s.length()-3) { // length()

if(s.substring(i,i+3).equals(“bit”)) { //substring(x,y), equals(str)

count++;

}

i++;

}

}

}

count, i?

public class String-Assignments {

public static void main(String [] args) {

String s;

String t = null;

String u = "you";

String v = new String("me"); // compare u and v!

String w = u + v;

}

}

s, t, u, v, w?

public class String-Commands {

public static void main(String [] args) {

String s = "Call me Ishmael!";

int len = s.length(); //length()

int ishPos = s.indexOf("Ish"); //indexOf(str)

int jackPos = s.indexOf("Jack"); // another indexOf(str)

String ishSub = s.substring(ishPos); //substring(x)!!!

char c = s.charAt(ishPos); // charAt(x)

}

}

ishPos, jacPos, ishSub, c, len?

// Here is an example that removes a portion of a String,

// and inserts a replacement

public class String-Insert-Delete {

public static void main(String [] args) {

String s = "It was a cold day!";

int start = s.indexOf("cold"); //indexOf(str)

int end = start + "cold".length(); //length()

s = s.substring(0, start) //substring(x,y)

+ "hot"

+ s.substring(end); //substring(x)!

}

}

s, start, end?

// Here is a typical example of a loop used to

// process a String.

// In this example, the loop visits each character

// in the String once.

public class String-Processing {

public static void main(String [] args) {

String s = "Call me Ishmael!";

int aCount = 0, i = 0;

char c = 0;

for( ; i<s.length(); i++) { //length()

c = s.charAt(i); //charAt(int)

if(c == 'a') {

aCount++;

}

}

}

}

aCount, c?

// Here is an example that repeatedly loops through the String,

// processing one word at a time.

public class String-Processing2 {

public static void main(String [] args) {

String s = "Ships at a distance have every man's wish.";

int spacePos1 = 0;

int spacePos2 = s.indexOf(" "); //indexOf(“<space>”)

String hyphenated = ""; //empty string

while(spacePos2>=0) {

String word = s.substring(spacePos1,spacePos2); //substring(x, y)

hyphenated = hyphenated + word + "-";

spacePos1 = spacePos2 + 1;

spacePos2 = s.indexOf(" ", spacePos1); //indexOf(str, x)

}

if(spacePos1<s.length()) {

hyphenated = hyphenated + s.substring(spacePos1); //substring(x)!

}

}

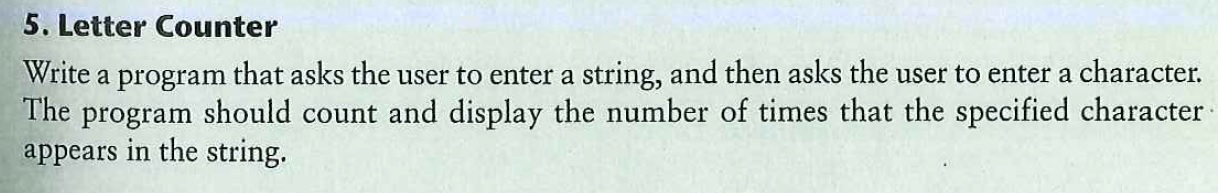
}

spacePos1, spacePos2, hyphenated?

# Repeat-X and Sum Algorithms with Strings

1. Write a counter-controlled loop to solve the following problems. Each one will involve a String.

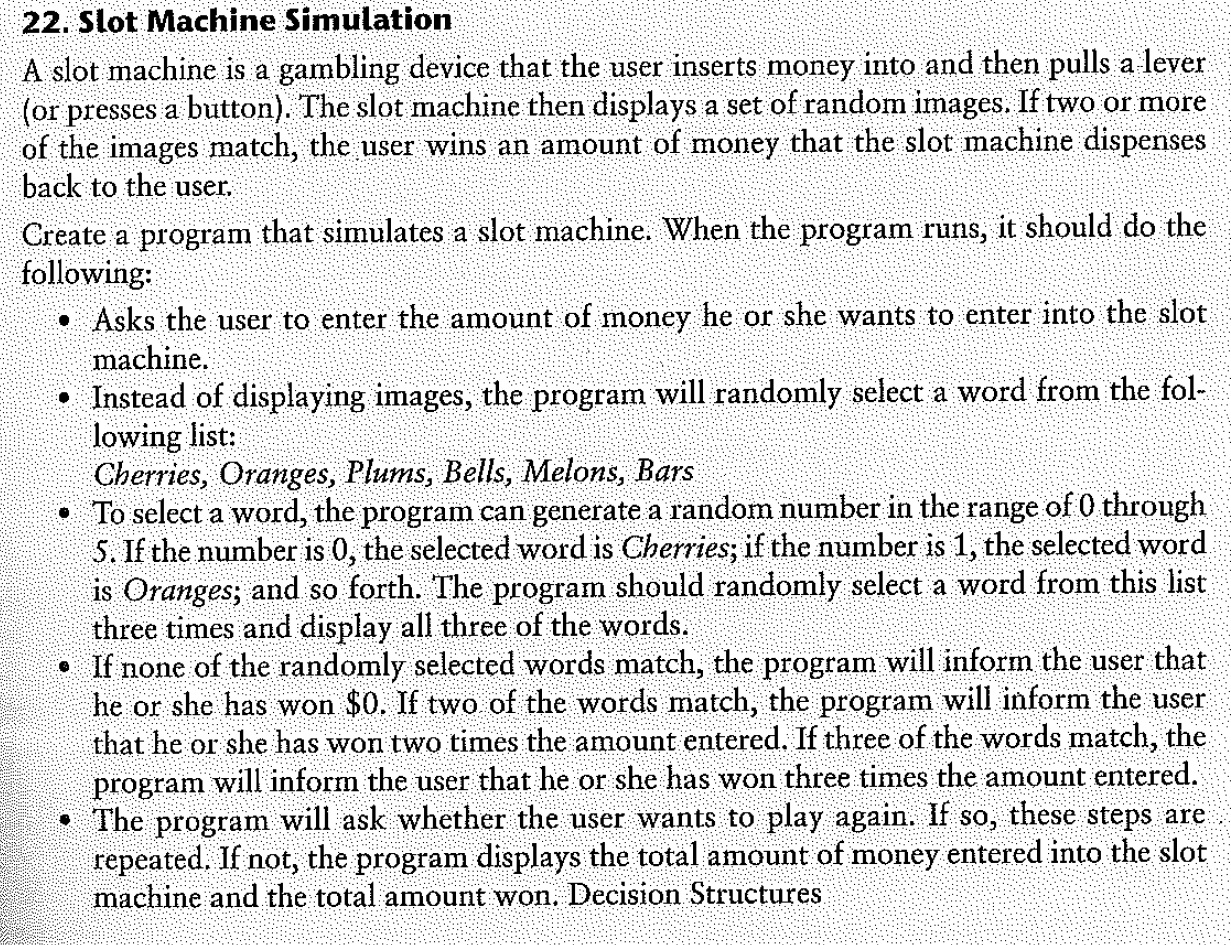
* (Reverse.java) Read a String from the keyboard. Declare another String variable called reverseStr, initialized to the empty String “”. Write a loop so that by the end of the loop, reverseStr contains the reverse of the String value stored.
* (Counting.java) Read a String from the keyboard, and count how many letter 's' or 'S' are in the String that the user enters.
* (Counter.java) challenges 5, p263. Note it is different from the above and needs an exact matching (i.e., case-sensitive).



* (TotalLengh.java) Read 10 Strings from the keyboard, and compute their total length.

1. Write an event-controlled loop to solve the following problem, based on the fence problem discussed in loop chapter.

* (SlotM.java), Challenge 22, p267. Note that the user can give an answer such as “Y,” “y,” “yes,” “YES,” “Yes,” etc., to continue the game.



* (StringConcat.java) This program asks the user to repeatedly enter a String. It should concatenate those Strings together, but insert spaces and the word “not” between every pair of words the user enters. Stop when the user enters the String “stop”. Display the final String. For instance, the program output might look like:

Please enter some Strings:

Such

eyes

you

have

stop

Such not eyes not you not have

Source code?