Dillon McLaughlin

Professor Kanapala

CSCE-145

9/5/2022

Problem:

To create an algorithm which prints an image to the console of initially one piece of toast and subsequently two pieces of toast which are horizontally adjacent.

Proposed Solution:

In order to achieve the desired output printing the images of toast, one must use a series of print statements [For example: System.out.println("Text inserted here")] combined with text symbols from the keyboard in the correct sequence in order to print the image/s to the console.

Tests and Results:

The body of the algorithm is as shown below. The toast is printed correctly to the console upon demand. The program does not contain values which vary in amount; therefore, the program would not fail to run due to invalid values.

```
System.out.println " _     _ ");
System.out.println "( `^`))";
System.out.println "|    ||";
System.out.println "|    ||";
System.out.println "'----'`";

System.out.println "  _   _                    _   _ ");
System.out.println "( `^`))                 ( `^`))";
System.out.println "|    ||                 |    ||";
System.out.println "|    ||                 |    ||";
System.out.println "'----'`                 '----'`";
```

Problems Encountered:

No significant problems were encountered during the creation of this solution. The process was fairly straightforward and left little room for syntax, run-time, and logical errors as the program consists almost solely of print statements.

Conclusion and Discussion:

The lab in question contained only extremely introductory material with respect to the creation of a program and/or algorithm. The only thing one could imagine would make the creation process more efficient would be to copy and paste the print statements from the original image of toast onto the lines below instead of recreating the statements. Thus, the principle applies: one could build on work previously completed in order to create a more complex program. (In this case two images of toast rather than one).

Additional Questions:

1. What is *bytecode*?

   Bytecode is an intermediate code that cannot be read by any machine. The source code, written and understood by the code writers, is converted into bytecode in order to be understood by any machine with JVM. Then it is further converted into machine code, consisting only of 0's and 1's. Thus, bytecode is essentially the "middle-man" allowing Java programs to be transferred between machines.

2. Expand RAM

   RAM, which stands for Random Access Memory, is the storage contained within the machine that contains the data which the computer is working with currently. RAM can be upgraded and increased depending on the machine's compatibility.