

CS125 Section 5 "Insert name here"

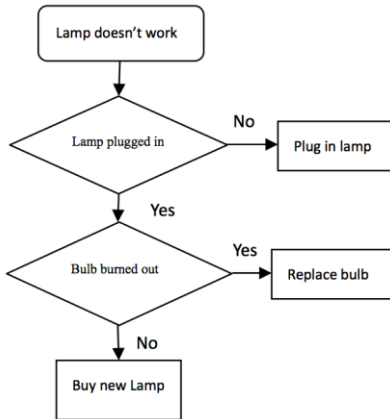
#1 Roles of Variables. The purpose of most variables can be described using a small number of roles. Recognizing these roles will help you learn how to read, write and understand more complex code. For each example read the whole example and then determine the role of each variable.

<p>Code Example 1</p> <pre>int num; // most-recent holder int sum = 0; // gatherer boolean done = false; // one-way flag while(!done){ num = TextIO.getlnInt(); if(num==0) done = true; sum += num; }</pre>	<p>Most-recent holder: The value of a most-recent holder is the latest one of a certain group or simply the latest input value.</p> <p>Gatherer: The value of a gatherer accumulates all the values gone through so far.</p> <p>One-way flag: A one-way flag has two possible values but cannot get its original value anymore after it has been once changed.</p>
<p>Code Example 2</p> <pre>int TOTAL = 100; // fixed-value int i=0; // stepper int num = 0; // most-recent holder int oldNum; // follower while(i < TOTAL){ oldNum = num; num = TextIO.getlnInt(); boolean isBigger = (num > oldNum); // temporary if(isBigger) TextIO.putln("New number is larger"); i = i + 1; }</pre>	<p>Fixed value: Its value is not changed after initialization.</p> <p>Stepper: Stepper goes through a succession of values in some systematic way, predictable succession of values.</p> <p>Follower: A follower always gets the old value of another known variable (usually most-recent holder) as its new value.</p> <p>Temporary: The value of a temporary is only needed for a very short period.</p>

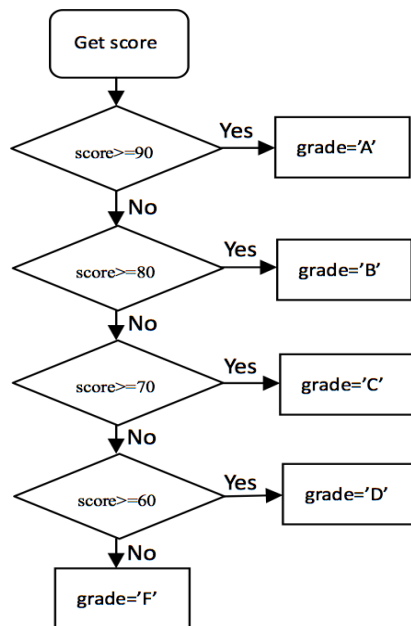
<p>Code Example 3</p> <pre>String line = TextIO.getln(); String newString = ""; // gatherer int i = 0; // stepper char lastChar = ' '; // follower boolean changedSomething = false; // one-way flag while(i < line.length()) { char c = line.charAt(i); // most-recent holder boolean toUC = Character.isWhitespace(lastChar) && Character.isLetter(c); // temporary if (toUC) { c = Character.toUpperCase(c); changedSomething = true; } newString += c; lastChar = c; i ++; } if (changedSomething) newString+="!"; TextIO.putln(newString);</pre>	<p>Code Example 4</p> <pre>int num; // most-recent follower int expected = 24; // fixed-value boolean found = false; // one-way flag while(!found){ num = TextIO.getlnInt(); if(num == expected){ TextIO.putln("Found number!!"); found = true; } }</pre>
---	---

#2: Flow charts and if-else structures

Flow charts are a common way to show program flow. Can you convert a flowchart specification into an if-else structure? Study the given the given example then try the scores problem. Carefully Check your neighbor's code with scores 90, 80, 70, 60 & 50.



```
// pseudo code
if(lamp plugged in) {
    if(bulb burned out) {
        replace bulb
    } else {
        buy new lamp
    }
} else {
    plug in lamp
}
```

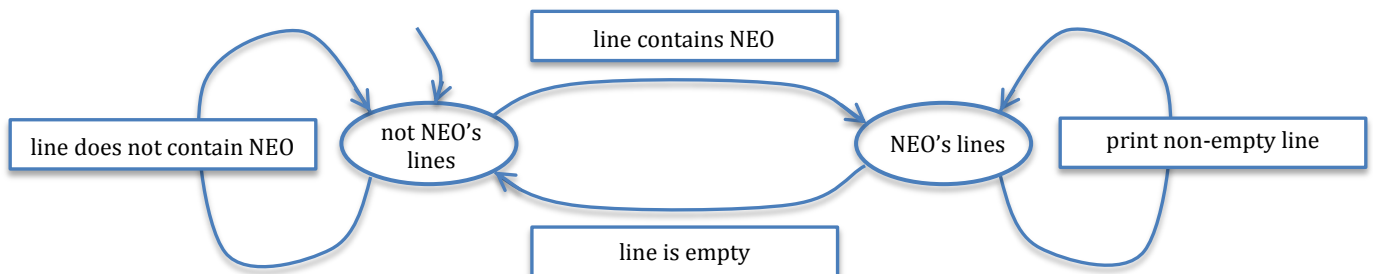


```
public class convertScore {
    public static void main (String args[]) {
        int score = TextIO.getInt();
        char grade='?'; // Add code in the space below.
```

```
if score >= 90
    grade = 'A';
else if score >= 80
    grade = 'B';
else if score >= 70
    grade = 'C';
else if score >= 60
    grade = 'D';
else
    grade = 'F';
```

```
System.out.println("Score: " + score + " Grade: " +
    grade);
} // end of main method
} // end of class
```

FSM ideas in MP#2:



Exam Prep

Fix the scoping error:

```
for(int i =0;i<10;i++){  
    TextIO.putln(i);}  
TextIO.putln("i="+i);
```

```
int i;  
for(i =0;i<10;i++) {TextIO.putln(i);}  
TextIO.putln("i="+i);
```

Loop analysis:

What is the output of the following code?

```
int x=0;  
int y = -15;  
while(y>2 || y<-2){  
    x++;  
    y = y / 2 + 1;  
    TextIO.putln("x="+x+",y="+y);  
    x = 2 * y - x;  
}
```

```
x=1,y=-6  
x=-12,y=-2
```

String review:

Modify the following program to print exactly the same output but using TextIO.put instead of TextIO.putln

```
TextIO.putln("\nHello");  
TextIO.putln("How are you?\n");
```

```
TextIO.put("\nHello\nHow are you?\n\n");
```

What is the output of the following code?

```
String str = "CS125 Java Java";  
int pos1 = str.indexOf("CS125");  
int pos2 = str.indexOf("Java");  
for (int i=pos1; i < pos2; i++) {  
    System.out.print(str.charAt(i));  
}
```

```
'CS125 '
```

Complete the following program.

Do not create additional variables or print the same letter in two different places in your code. Complete the program below to print out either "C" "D" "F" or "T" (cat, dog, fish or toy respectively). Use the following rules:

A condo dweller gets a cat provided they have \$13 or greater to spend.

A house owner gets a dog provided they have more than \$17 to spend.

A dorm dweller with \$5 or more, and also house owner with exactly \$5, gets a fish.

All others should get a stuffed toy.

```
public class PurrfectPetsRecommendation {  
    public static void main (String[] args) {  
        TextIO.putln("Do you live in a Dorm, Condo or House?");  
        TextIO.putln("Type D C or H and press return");  
        char type = TextIO.getlnChar(); // assume either D C or H  
        TextIO.putln("Dollars you can spend on your pet each week?");  
        int dollars = TextIO.getlnInt(); // assume a valid integer  
  
        if(type == 'C' && dollars >= 13){  
            TextIO.putln("C");  
        }  
        else{  
            if(type == 'H' && dollars > 17){  
                TextIO.putln("D");  
            }  
            else{  
                if((type == 'D' && dollars >= 5) || (type == 'H' && dollars == 5)){  
                    TextIO.putln("F");  
                }  
                else{  
                    TextIO.putln("T");  
                }  
            }  
        }  
  
    } // end of main method  
}  
// end of class
```

Call Log Retrieval

Following text file was recovered from 007's recent call log when his cell phone crashed. Print out the lines containing "Angrave" with the next 2 call logs (following 2 lines), if any. Printed lines should in the original (unmodified) mixture of upper- and lower-case.

```
"moneyYPenny 650-857-0275, James Bond 100-113-0519, 02/14/2012, 08:12AM, 00:01:16
James bOnd 100-113-0519, Angrave 217-001-1407, 02/14/2012, 06:45Am, 00:10:01
M 212-861-9692, JaMes Bond 100-113-0519, 02/13/2012, 10:43pm, 00:00:03
Q 509-553-1081, James bonD 100-113-0519, 02/12/2012, 11:14AM, 00:03:24
JamEs BoND 100-113-0519, SKYlon REStaurant 207-654-7800, 02/10/2012, 04:20pM, 00:01:09
AnGrAvE 217-001-1407,James BOnd 100-113-0519, 02/10/2012, 07:12AM, 00:02:22 ..."
```

```
public class CallLogRetrieve {
    public static void main(String [] args) {
        TextIO.readFile("007CallLog.txt");

        int count = 0;
        boolean print = false;
        while (!TextIO.eof()){
            String line = TextIO.getLn();
            if (print){
                TextIO.putLn(line);
                count--;
            }
            // Advanced: // if (line.toLowerCase().contains("angrave")){ count = 2;}
            if (count == 0){ print = false;}
            if (!print){
                if (line.toLowerCase().contains("angrave")){
                    TextIO.putLn(line);
                    print = true;
                    count = 2;
                }
            }
        }

        } // end of main method
    } // end of class
```

Advanced: Bond and Angrave may have spoken more than once in any 3 consecutive calls in the call history.