Objectives: For Loops. Scope Errors. SCOPE Lament
Deadlines: MP1 graded tonight 8pm (99%) & Mon 8pm (98%)
MP2 due Mon 8pm
MP3 "Top Secret" will be out soon…
This lecture will make most sense if you did the readings....

1. Spot the mistake in this code? *output* is a boolean; inspired by MP2

```
if (output = true);
   TextIO.putln("Hello"); // This example is
```

2. For the following code,

```
for (int i = 100 ; i > 0 ; i = i / 10) {  TextIO.put(i); }
```

a. What does it print?
b. How many times is is i=i/10 evaluated?
c. How many times is i>0 evaluated?
d. Convert the above code into an equivalent while loop.

3. What is the final value of i?

```
int i = 4; for (i-- ; i < 15; i++) { i = i * 2; }
```

4. Convert the following code to use a for loop

```
int count = 0;
int x = 7;
while (x < 50) {
  x = x * 2;
  count ++;
}
TextIO.putln("Final value:" + x);
```

5. Which examples will have the same behavior?

```
// Read an integer value from the user:
int b = TextIO.getln();
// Followed by one of the following:
```

A) `int  i; for (i = b ; i < 10 ; i ++) { i = i * 2; }`

B) `int i = b; for ( ; i < 10; i++) i = i * 2;`

C) `for (int i = b ; i < 10 ; ) { i = i * 2; i++; }`

D) `for (int i = b ; i < 10 ; ); { i = i * 2; i++; }`

E) `int i = b; while (i < 10) { i = i * 2; i ++; }`

F) `int i = b; while (i < 10); {i = i * 2; i ++; }`

G) `int i = b; do { i = i * 2; i ++ ; } while (i < 10);`

Be mindful of the short lifetime of temporary variables

A lament for lost variables by L. Angrave 9/18/2009

A temporary variable, known by some as a "local"
burns brightly but not for long:
You close your brace "}"
or in haste leave your for-loop -
They're done, spent-up, lost. Gone.
But don't give up hope.
Your problem is *scope*.
Declare them before
and they will live for a little more.
But soon your function will return.
And its temporary variables you'll no longer need.
Their precious memory locations, to be repurposed, are freed.

Professor Jack Good, cryptanalyst working at the time with Turing at Bletchley Park, later said: "Turing's most important contribution, I think, was of part of the design of the bombe, the cryptanalytic machine. He had the idea that you could use, in effect, a theorem in logic which sounds to the untrained ear rather absurd; namely that from a contradiction, you can deduce everything." (Source: Wikipedia)

The bombe searched for possibly correct settings used for an Enigma message (i.e., rotor order, rotor settings, etc.), and used a suitable "crib": a fragment of probable plaintext. For each possible setting of the rotors (which had of the order of 10^{19} states, or 10^{22} for the U-boat Enigmas which eventually had four rotors, compared with the usual Enigma variant's three), the bombe performed a chain of logical deductions based on the crib, implemented electrically. The bombe detected when a contradiction had occurred, and ruled out that setting, moving onto the next. Most of the possible settings would cause contradictions and be discarded, leaving only a few to be investigated in detail. Turing's bombe was first installed on 18 March 1940.

Solving "Knight and Knaves" Logic Problems Computer Science Style!

- Person 1 says "Person 2 is lying"
- Person 2 says "There are two liars here"

```
// 0 = liar, 1 = tells the truth
for (int person1 = 0 ; person1 < 2 ; person1++)
  for (int person2 = 0 ; person2 < 2 ; person2++){
      // Person 1 says "Person 2 is lying"
      boolean testimony1IsTruthful = (person2 == 0);
      // Person 2: "There are two liars here"
      boolean testimony2IsTruthful = ((person1 + person2) == 0);

      boolean assertion1 = ((person1 == 1) && testimony1IsTruthful)
          || ((person1 == 0) && !testimony1IsTruthful);
      boolean assertion2 = ((person2 == 1) && testimony2IsTruthful)
          || ((person2 == 0) && !testimony2IsTruthful);

      TextIO.put("Person 1 is "
          + ((person1 == 0) ? "a liar" : "truthful")
          + ". Person 2 is "
          + ((person2 == 0) ? "a liar" : "truthful") );
      TextIO.putln(": Fits assertion 1 and 2 ?" + assertion1 + "," +
assertion2);
   }
}
```

6. When do *i* and *j* go out of scope?
   What does the following code snippet print?

```
public static void main(String[]) {
        int i = 4;
        while (i < 6) {
            int j =1;
            while (j < 3) {
                TextIO.put("(" + i + "," + j + ")");
                if (j > 1) TextIO.put(",");
                j ++;
            }
            TextIO.putln();
        }
}
```

7. Write a program to print out all possible 2 letter words aa to zz:
Hint use 2 for loops.

8. Complete the following program to print a triangle of stars:

```
*
**
***
****
public static void main(String[] ){
TextIO.putln("Number of rows?")
int n = TextIO.getlnInt();
```