CS 125 - Lecture 12

Objectives:

For loops; While loops; loopy algorithms

To do: MP2 - commit; MP3 is out! Keep up with reading and Turing's Craft

1. **Analyze this: How many dots are printed?**

```
public static void main(String[] args) {
   int a = 0;
   int b;
   while (a < 20) {
      a += 2;
      b = 1;
      while (b < 16) {
         TextIO.put('.');
         b = b * 2;
      }
      TextIO.putln(a);
   }
}
```

2. **Ternary operator**

Composed of : **one item => unary**
two items => binary
three items ==> ternary

Ternary operator examples:

___ ? ___ : ___ is useful if you know how to use it…

```
int value = TextIO.getlnInt();

int bounded = value>10 ? 10 : value;

double average = count >0 ? sum / (double)count : 0;

String mesg=count+“File"+((count!=1) ? "s" : "")+" copied.";
```

3. **Does the following method work as described.** Justify and discuss your answer with another student.

```
/** Rolls two simulated 6 sided dice until both die values are equal
* to one.  Prints out the number of times the dice were rolled.
* @return the dice roll encoded as an integer value. */
public static int rollSnakeEyes() {
   int dice1 = 0, dice2 = 0;
   int count = 1;
   boolean foundSolution = false;
   while (!foundSolution) {
      dice1 = 1 + (int) (Math.random() * 6);
      dice2 = 1 + (int) (Math.random() * 6);
      foundSolution = dice1 + dice2 == 2;
      count++;
   }
   TextIO.putln("That took " + count + " rolls”);
   return dice1 + 10 * dice2;
}
```

4. **Modify the code below to roll three dice.** It should keep rolling until the dice values are unique. You'll need to i) create a new variable (dice3); ii) roll dice 3; iii) change the foundSolution expression and iv) the return expression should represent the number of iterations required.

```
/** Rolls three simulated 6 sided dice until all die values
/*  are unique.
public static int rollThreeUniqueDice() {
   int dice1 = 0, dice2 = 0;

   boolean foundSolution = false;
   while (!foundSolution) {
      dice1 = 1 + (int) (Math.random() * 6.0);
      dice2 = 1 + (int)( Math.random() * 6.0);









   }


}
```

5. **(Sneak Peak at MP3) Complete & fix the bugs in the following code:**

```
public static void encrypt() {
/** Prints encrypted string. a->b, b->c,c->d...,z->a but leave
other characters unchanged  */
   int count = 0;
   int i=0
   String mesg = "Hello World!";
   while( i <
      char c= mesg.charAt( )
      count ++;
      if c>"a" || c<"z" {
         int letter = c – 'a';
         int encrypted = (letter+1) % 26;
         c = (char) ('a' +  encrypted);
      } else count–;


   TextIO.put(c);


   TextIO.putln(count + " chars modified")
```

Why is the last 'else' important? What would happen if it was omitted?

6. **Be a human compiler:**

A. Decompose the following expression into a sequence of three or four simple steps (pseudo code) that the virtual machine might execute. Watch out for the *type* conversions. Math.random() returns a number of type double between 0.0 and 0.99999999...

```
(int) (Math.random() * 6)
```

B. Why are the three pairs of parentheses necessary?

C. List the possible values of the above expression:

7. **What is the final value of i?**

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

8. **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);
```

9. **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.

10. **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);
```

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.

9. **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();
         i++;
     }
}
```

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[] args) {
   TextIO.putln("Number of rows?")
   int n = TextIO.getlnInt();
```