CS 125 - Lecture 11

Objectives:

Algorithm development; Ternary operator

To do: MP2 (MP1-regrade); Keep up with readings and Turing's Craft; Quiz #2

1. Warm-up. Complete the following using 'true' or 'false' 'anything' means any value as long as it is 'true' or 'false'!

||: _____ OR'd with anything is _____

&&: _____ AND'd with anything is _____

2. Which code snippets increment the value of count?

```
count + 1;
count = count +1;
count += 1;
count ++;
++count;
count = 1 + count;
```

3. Fix and/or simplify the following statements (don't change the context).

```
boolean output = line.indexOf("spoon") == true;

if( score > 80 == true) TextIO.putln("First");

if( score > 70 == false) TextIO.putln("Second");

if( score > 60 == false) TextIO.put("");
```

4. Pre & Post Increment Challenge (aka unreadable code)

Why does the following code print "x=2,yPost=1, yPre=6"?

```
int x = 0;

int yPost = 2 * x++ + x;

int yPre = 2 * ++x + x;

System.out.println("x="+x+",yPost="+yPost+", yPre=“+yPre);
```

5. Fill in the missing the code and fix any errors you notice.
Update the code so it keeps asking for a password until a good password is entered.

```
_____________ done = false;

_____________________________

TextIO. _____________________ ( “Prompt the user: New password? 10
or more characters, mixed case, no spaces" );


____________________ = TextIO. _____________


____________ short = ____________________; // true if too short
____________ noUpperCase= _____________
____________ hasSpaces = _____________
____________ badPass = short || noUpperCase && hasSpaces;


if( _____________) }
    TextIO.putln("Bad password – try again.");
}
TextIO.putln("Password accepted, thanks.");
```

6. **Fix this PALINDROME CHECKER:**

```
public static void main(String[] args) {
   String original = “Bob";
   String s = original.toUpperCase();
   boolean isPalindrome = true;
   // We'll change isPalindrome to false
   // if we find a counter-example
   int lengthToCheck = s.length() / 2;
   int i = 0;
   while (i < lengthToCheck && isPalindrome) {
      if (s.charAt(i) != s.charAt(s.length() - i)) {
         isPalindrome = false;
      }
      i++;
   }
   if (isPalindrome)
      TextIO.putln(original + " is a palindrome”);
}
```

7. **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);
   }
}
```

8. **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.";
```

9. **Two common ways to nest if-statements:**

```
if( inChicago ) {
   if( withFriends ) {
      goDownTown();
   } else {
      callFriends();
      goDownTown();
   }
else {
   if(withFriends {
      watchMovieTogether();
   } else {
      watchTV();
   }
}
```

picture:

```
if(inChicago)
   goDownTown();

else if(inWisconsin)
   goSkiing();

else if(inNYC )
   eatBagel();

else browseFB();
```

picture:

10. **Spot the Mastikes**

Some code starts with the following :

```
String s = TextIO.getln();
boolean ok = _____ see erroneous expressions below
```

We need you to fix the following to be correct and accurate Java expressions. Note, “iff” means “if and only if”.

Evaluates to true iff *s* contains "Jim" or "Fred".
(Ignore upper/lower case e.g. “jiM” should evaluate to true)

```
s.toLowerCase.indexOf('jim') > 0 | s.toLowerCase.indexOf('Fred') == true
```

Should be true iff *s* has at least four characters and starts with "ABCD":

```
s.length = 4 & s.substring(1,4) = “ABCD"
```

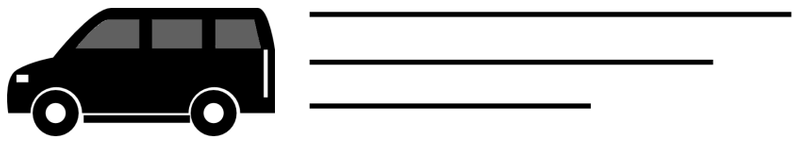
Write an expression that is true iff *s* starts with "ABC" or *s* is an empty string and false otherwise:

11. Code analysis

```
// What happens if it reads "Help"?
// What happens if it reads "Think Secret!"?

  TextIO.readFile(“data.txt”);

   String word = TextIO.getln();
   int posn = word.toLowercase().indexOf("secret");

   if(posn != -1) TextIO.putln( word.substring(0,posn) );
```

5. **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);
   }
}
```



7. **What do code 'blocks' and minivans with tinted windows have in common?**

Scope: the 'region' of code in which a variable is valid (i.e., can be read or written).

```
int y = 8;  <--- y declared outside block
if (y < 13)
   {
      int x = 3; <--- x declared inside block
      System.out.print(x); <-- x can be used here
      y = y + 7;   <--- y can be used here
   }
System.out.println(y);  <--- output: _______________
System.out.println(x);  <--- output: _______________
```

```
if ( x > y ) {
   int temp;  // A temporary variable for use in this block.
   temp = x;  // Save a copy of the value of x in temp.
   x = y;  // Copy the value of y into x.
   y = temp;  // Copy the value of temp into y.
}
```

6. **Does the following method works 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;
}
```

8. **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
/** Prints encrypted string. a->b, b->c,c->d...,z->a but leave other
characters unchanged  */
void encrypt() {
   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';

      } else count --;

   TextIO.put(c);
   TextIO.putln(count + " chars modified")
```

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

6. **What do code 'blocks' and minivans with tinted windows have in common?**

Scope: the 'region' of code in which a variable is valid (i.e., can be read or written).

```
int y = 8;  <--- y declared outside block
if (y < 13)
   {
      int x = 3; <--- x declared inside block
      System.out.print(x); <-- x can be used here
      y = y + 7;   <--- y can be used here
   }
System.out.println(y);  <--- output: _____
System.out.println(x);  <--- output: _____
```

```
if ( x > y ) {
   int temp;  // A temporary variable for use in this block.
   temp = x;  // Save a copy of the value of x in temp.
   x = y;  // Copy the value of y into x.
   y = temp;  // Copy the value of temp into y.
}
```

7. **Sample exam question:** Complete the following program:

```
public class PostMaster {

/**
* Print "RAIL" , "UPS", "DHL" or "FEDEX" *
* Domestic Non-urgent packages under 10 lbs are shipped
     UPS
* Domestic Urgent packages 150 lbs or greater are
     shipped by RAIL
* International packages are always shipped using FEDEX
* All other packages are shipped using DHL */
public static void main(String[] args) {
   TextIO.putln("Package Weight?");
   int weight = TextIO.getlnInt();

   TextIO.putln("Urgent?");
   boolean urgent = TextIO.getlnBoolean();

   TextIO.putln("International?");
   boolean international = TextIO.getlnBoolean();
```

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: