nested loops with extras

Lab 11

The nested loop

Nested Loop Definition

A nested loop is a loop with another loop inside. Each time the outer loop runs one time, the inner loop runs to completion.

Nested for loop

```
for(int outer=1; outer<=6; outer++)
{
   for(int inner=1; inner<=6; inner++)
      System.out.print("*");
   System.out.println( );
}
```

OUTPUT

```
******
******
******
******
******
******
******
```

Open nestedforbox.java

Nested for loop

```
int outer=1;
      //start      //stop      //increment
for(outer=1;  outer<=2;     outer++)
{       //start      //stop      //increment
  for(int inner=1;  inner<=2;  inner++)
    out.println(outer + " " + inner);
  out.println();

}
```

OUTPUT

```
1 1
1 2

2 1
2 2
```

Open nestedfor.java

Tracing Nested Loops

```
for(int r=1; r<=3; r++)
{
  for(int c=1; c<=r; c++)
    System.out.print("*");
  System.out.println();
}
```

r	c	output
1	1	*
1	2	
2	1	
2	2	**
2	3	
3	1	
3	2	
3	3	***
3	4	
4		

Nested for loop

```
int stop=5;
for(int r=1; r<=stop; r++)  //rows
{
  for(int c=1; c<=r; c++)   //columns
    System.out.print("*");
  System.out.println();
}
```

OUTPUT

```
*
**
***
****
*****
```

Nested for loop

```
int stop=3;
String output="";
for(int r=1; r<=stop; r++)  //rows
{
  for(int c=1; c<=r; c++)   //columns
    output+="<";
  output+="\n";
}
System.out.println(output);
```

OUTPUT

```
<
<<
<<<
```

Open nestedfortri.java nestedfortristring.java

Nested while loop

```
int outer=1;
while(outer<=2)
{
  int inner=1;
  while(inner<=3)
  {
      out.println(outer + " " + inner);
      inner++;
  }
  System.out.println( );
  outer= outer+1;
}
```

OUTPUT

```
1 1
1 2
1 3

2 1
2 2
2 3
```

Open nestedwhile.java nesteddowhile.java

Start work on Lab 11

break

break

continue

break and continue are very popular on UIL tests.

break

break is a reserved word that allows you to shut down the loop.

```
int run;
for(run=1; run<=20; run++)
{
   if (run%3==0)
        break;
}
System.out.println(run);
```

OUTPUT

3

Open break.java

continue

continue is a reserved word that allows you to skip statements.

```
int cnt=0;
for(int run=1; run<=20; run++)
{
    if(run%3==0)
      continue;
    cnt++;
}
System.out.println(cnt);
```

OUTPUT

14

Open continue.java

Character

StringBuffer

Character and StringBuffer are very popular on UIL tests.

Character

frequently used methods

Name	Use
isUpperCase(c)	checks if c is upper case – returns true/false
isLowerCase(c)	checks if c is lower case – returns true/false
isDigit(c)	checks if c is a digit – returns true/false
toUpperCase(c)	returns uppercase version of c
toLowerCase(c)	returns lowercase version of c

Character

```
char c ='A';
out.println(isUpperCase(c));
out.println(isLowerCase(c));
out.println(isDigit(c));
out.println(toUpperCase(c));
out.println(toLowerCase(c));
```

OUTPUT

```
true
false
false
A
a
```

Open charone.java chartwo.java

StringBuffer

frequently used methods

Name	Use
All of the String methods plus more.	
setCharAt(x, c)	set char at x to value c
setLength(c)	change the length to x
reverse()	reverse the order of all chars

StringBuffer

```
StringBuffer s = new StringBuffer("abc");
out.println(s);
s.setCharAt(0,'X');
out.println(s);
s.setLength(10);
s.setCharAt(9,'0');
out.println(s);
s.reverse();
out.println(s);
```

OUTPUT

```
abc
Xbc
Xbc      0
0      cbX
```

Open sbufferone.java

Continue work on Lab 11