Mr. Fahrenbacher
AP Computer Science
Unit 3B - Definite and Indefinite Loops

List of Java Concepts and Vocabulary From This Unit

- For Loops
- Math Sequences
- Accumulation
- Scope
- Finding Max Value
- While Loops
- For-While Loop Equivalence
- Infinite Loops

Definite Loops

Read the program below and compare the code with the output. Try to deduce how the new code affects how the program runs.

```
public class CountUp {

	public static void main(String[] args) {

		int i;
		for(i = 1; i <= 4; i++) {			//for loop
			System.out.print(i + ", ");
		}
		System.out.println(i);
	}
}
```

Program Output: **1, 2, 3, 4, 5**

The Life Cycle of the For Loop

1. code before the loop begins (non-repeated)

```
                                        ||=============
                                        v                ||
for( 2. variable initialization; 3. keep going condition; 5. some variable change that gets us closer to stopping) {


	4. body of the loop - this IS repeated

}
```

6. code after the loop ends (not-repeated)

Loops and Drawing (http://semmy.me/ide/)

Below is code for another language called Processing, which is very similar to Java. Copy this code into this online ide for Processing. Can you modify the code to use a loop to draw the 8 ellipses?

```
void setup()
{
  //sets the size of the drawing area
  size(400,400);
}


void draw(){

  //filles the background with the color white
  background(255, 255, 255);

  //sets the fill color to green (r, g, b) - each between 0 and 255
  fill(100, 200, 50);

  //draws 8 ellipses
  ellipse(100, 100, 20, 20);
  ellipse(100, 120, 20, 20);
  ellipse(100, 140, 20, 20);
  ellipse(100, 160, 20, 20);
  ellipse(100, 180, 20, 20);
  ellipse(100, 200, 20, 20);
  ellipse(100, 220, 20, 20);
  ellipse(100, 240, 20, 20);
}
```

Fun Extras

- Try using the variable mouseX, mouseY, or frameCount to replace any of the numbers above
- Initialize an int variable at the top of the program to 0, increment that variable by 1, modded by 400, and use that variable as the x-coordinate of the ellipses… what do you notice?
- Check out the Processing language documentation to see other commands you can add!

When Does The Loop Actually End? (http://semmy.me/ide/)

Below is code for another language called Processing, which is very similar to Java. Copy this code into this online ide for Processing. How many ellipses do you think the code below will draw?

```
void setup()
{
   //sets the size of the drawing area
   size(400,400);
}


void draw(){

   //filles the background with the color white
   background(255, 255, 255);

   //sets the fill color to green (r, g, b) - each between 0 and 255
   fill(100, 200, 50);

   for(int i = 1; i <= 5; i++) {
            i++;
            ellipse(100, 100*i, 20, 20);
    }
}
```

Math Sequences

Below is a program that demonstrates two different strategies for a for loop to generate the first **11** perfect squares. Compare and contrast the two strategies - pay close attention to the use of the variable i.

```
public class PerfectSquares {

	public static void main(String[] args) {

		//Approach #1
		for(int i = 0; i <= 10; i++) {
			System.out.print(i * i + " ");
		}

		System.out.println();

		//Approach #2
		int perfectSquare = 0;
		int diff = 1;

		for(int i = 0; i <= 10; i++) {
			System.out.print(perfectSquare + " ");

			perfectSquare += diff;
			diff += 2;
		}
	}
}
```

Program Output: **0 1 4 9 16 25 36 49 64 81 100**
0 1 4 9 16 25 36 49 64 81 100

Accumulating a Sequence (Series)

Consider the program below - what is the code trying to do? How does it accomplish its intended task?

```
public class TriangleNumber {

	public static void main(String[] args) {

		int sum = 0;
		for(int i = 0; i <= 100; i++) {
			sum += i;
		}

		System.out.println("The sum of integers from 0 to 100 is " + sum);
	}
}
```

Accumulating Values From a User

Consider the program below - what is the code trying to do? How does it accomplish its intended task?

```
import java.util.Scanner;

public class TotalBill {

	public static void main(String[] args) {

		Scanner input = new Scanner(System.in);

		int sum = 0;
		for(int i = 0; i < 5; i++) {
			System.out.print("Enter the cost of the next item: ");
			int next = input.nextInt();
			sum += next;
		}

		System.out.println("Your total bill is  " + sum);
	}

}
```

Scope

The previous program is modified below... but this modification creates an error on the print statement. Why do you think it creates an error?

```
import java.util.Scanner;

public class TotalBill {

	public static void main(String[] args) {

		Scanner input = new Scanner(System.in);

		for(int i = 0; i < 5; i++) {
			int sum = 0;		//Dory problem - can't use variables declared in loop
						//after the loop
			System.out.print("Enter the cost of the next item: ");
			int next = input.nextInt();
			sum += next;
		}

		System.out.println("Your total bill is  " + sum);
	}

}
```

Counting Strategy

The program below uses a common strategy to determine if a set of values has a certain property. Identify the strategy.

```
import java.util.Scanner;

public class AllTall {

	public static void main(String[] args) {

		Scanner input = new Scanner(System.in);
		int count = 0;  //accumulator

		for(int i = 0; i < 24; i++) {  //loop runs 24 times
			System.out.print("Enter the next student's height in inches: ");
			int height = input.nextInt();  //reads in a new number from the user

			//check if entered value is 'short', if so we increase the count
			if(height < 66) {
				count++;
			}
		}

		if(count > 0) {
			System.out.println("We have " + count + " many short students");
		}
		else { //count = 0
			System.out.println("Everyone is tall!");
		}
	}
}
```

Finding the Max

Read the program below, a modification on an early program we looked at. How does the code go about accomplishing its task. Does it work? Would it work if the array was changed (different number, different length)?

```
public class TheTallest {  //of 25 people!

	public static void main(String[] args) {

		Scanner input = new Scanner(System.in);
		System.out.print("Enter the first height: ");
		int max = input.nextInt();  //first person

		for(int i = 1; i <= 24; i++) {
			System.out.print("Enter the next height: ");
			int next = input.nextInt();

			if(next > max) {
				max = next;
			}
		}

		System.out.println("The tallest height is " + max);
	}
}
```

Arrays And Loops (Not on Quiz)

Accumulating an Array (Before Loops)

Consider the program below - locate the new notation and determine how it enables the program to accomplish its intended task.

```
public class Fundraiser {

    public static void main(String[] args) {

        int[] money = {55, 32, 49, 36, 81, 20, 104, 53, 99, 23, 125, 37, 54, 92};

        int sum = 0;
        sum += money[0];
        sum += money[1];
        sum += money[2];
        sum += money[3];
        sum += money[4];
        sum += money[5];
        sum += money[6];
        sum += money[7];
        sum += money[8];
        sum += money[9];
        sum += money[10];
        sum += money[11];
        sum += money[12];
        sum += money[13];

        System.out.println("The sum of all money raised is " + sum);
    }
}
```

Accumulating an Array (With Loops)

Consider the program below - locate the new notation and determine how it enables the program to accomplish its intended task.

```
public class Fundraiser {

	public static void main(String[] args) {

		int[] money = {55, 32, 49, 36, 81, 20, 104, 53, 99, 23, 125, 37, 54, 92};

		int sum = 0;
		for(int i = 0; i < money.length; i++) {
			sum += money[i];
		}

		System.out.println("The sum of all money raised is " + sum);
	}
}
```

Scope with Arrays

The previous program is modified below... but this modification creates an error on the print statement. Why do you think it creates an error?

```
public class Fundraiser {

	public static void main(String[] args) {

		int[] money = {55, 32, 49, 36, 81, 20, 104, 53, 99, 23, 125, 37, 54, 92};

		for(int i = 0; i < money.length; i++) {
			int sum = 0;
			sum += money[i];
		}

		System.out.println("The sum of all money raised is " + sum);
	}
}
```

Counting Strategy With Arrays

The program below uses a common strategy to determine if an array has a certain property. Identify the strategy. Can the strategy be improved?

```
public class AllPassing {

	public static void main(String[] args) {

		int[] grades = {80, 65, 91, 83, 72, 101, 59, 81, 99, 61};
		int count = 0;

		for(int i = 0; i < grades.length; i++) {
			if(grades[i] < 60) {
				count++;
			}
		}

		if(count > 0) {
			System.out.println("You failed " + count + " class(es)");
		}
		else {
			System.out.println("Congratulations on passing all of your classes!");
		}
	}
}
```

Finding the Max With Arrays

Read the program below, a modification on an early program we looked at. How does the code go about accomplishing its task. Does it work? Would it work if the array was changed (different number, different length)?

```
public class TheGreatest {

	public static void main(String[] args) {

		int[] array = {103, 25, 49, 57, 107, 52, 109, 901, 54, 999, 103, 36, 52, 1};
		int max = array[0];

		for(int i = 1; i < array.length; i++) {
			if(array[i] > max) {
				max = array[i];
			}
		}

		System.out.println("The largest value in the array is " + max);
	}
}
```

Indefinite Loops

The code below introduces a new type of loop. How do you think this code works? How is it similar / different than a for loop?

```
int num = 1;

while(num <= 100) {
        num *= 2;
}

System.out.println("First power of 2 greater than 100 is " + num);


int x = 99;
int div = 2;

while(x % div != 0) {
        div++;
}

System.out.println("The smallest divisor greater than 1 is " + div);
```

The Life Cycle of the While Loop

1) variable initialization that will **control** the loop

while(2) **keep going condition** - usually depends on the variable made before the loop) { ⇐====
||
||
3) various commands ||
||
4) some command that modifies a variable that controls the loop ==========

}

5) post loop code

For-While Equality

Can anything done by a while loop also be done by a for loop (or vice-a-versa)? Try this task below.

While	For

int x = 1;
while(x < 10){
System.out.println("A");
x+=2;
}	

int sum = 0;
int y = 3;
while (y > -4) {
sum += y;
y = y - 1;
}
	int sum = 0;
	for(int y = 3; y > -4; y = y - 1){
	sum += y;
	}

Will Someone Make It Stop!

Why is the code below very appropriate for the song?

```
int x = 1;
while(x < 10) {
        System.out.println("This is the song that never ends");
        System.out.println("It just goes on and on my friend");
        System.out.println("Some people started singing it not knowing what it was,");
        System.out.println("And they'll continue singing it forever just because");
}
```

Accumulating An Indefinite Number of Values From a User

Consider the program below - what is the code trying to do? How does it accomplish its intended task?

```
import java.util.Scanner;

public class TotalBill {

	public static void main(String[] args) {

		Scanner input = new Scanner(System.in);

		int sum = 0;
		System.out.print("Enter the cost of the next item, 0 or less to stop: ");
		int next = input.nextInt();
		int count = 0;

		while(next > 0) {

			sum += next;
			count++;

			System.out.print("Enter the cost of the next item, 0 or less to stop: ");
			next = input.nextInt();
		}

		System.out.println("Your total bill is  " + sum + " , with each item averaging " + sum / count);
	}

}
```