Name: Bartu Ōnel 121A Date: ____________

AP Computer Science A
Iteration. Notes

While Loops. A while loop is a control structure that allows you to write code that is executed repeatedly as long as some condition is true.

Example 1.

```
int n = 3;
while( n <= 5 ) {                    // if the expression is true, execute the block of code.
        System.out.print(n + "x");
        n++;
}
```

The above prints 3x4x5x

Every pass through the body of a loop is called an ____________________.

Example 2.

```
int n = 0;                     (1- 7)
while (n < 10) {
        int k = (int) (7 * Math.random()) + 1;
        System.out.print(~~k +~~ "  ");
                         5
        n += k;
}
System.out.print(n);
```

k	n
5	5
6	11

If the first numbers printed are 5 and 6, what is the last number printed? 11

If a total of four numbers are printed, and the last number is 12, which one(s) of the following can be the first three numbers.

[✓ a)] 5, 1, 6
~~b)~~ 5, 6, 1
~~c)~~ 3, 2, 8
[✓ d)] 4, 4, 4

k	n
5	5
1	6
6	12

k	n
5	5
6	11

k	n
3	3
2	5
8	

Example 3.

```
Scanner sc = new Scanner( System.in );
System.out.println("Password?");
String pass = sc.nextLine();
while ( !pass.equals( "asdf") ){
        System.out.println("Password?");
        pass = sc.nextLine();                  // line X
}
System.out.print("Success");
```

What must happen for Success to be printed? if pass.equals("asdf")

If line X was deleted, how would it effect the code? then the code does not ask a new input after it prints "Password?"

The Break Statement. The break statement causes a loop to end immediately.
For example:

```
        int x = 0;
        while ( x >=0)  {
                x = (int)( 100 * Math.random() );          // Creates a rondom  int value
                if (x % 10 == 0 )                              between  0 and 99 inclusive.
                        break;
        }
        System.out.println( x );  // what may be displayed? It prints the rondom values one
```

under the other until the value becomes the multiple of 10.

For Loops. A for-loop is typically used when you need to repeat a process a specific number of times.

The first statement in a for-loop contains three statements separated by semicolons.

The three parts of a for loop:

1. The first statement usually declares and initializes a variable known as the loop counter.
2. The middle statement is a condition that usually involves the counter.
3. The last statement indicates how to Change the counter after each iteration.

Example 1. What does this display? 3x4x 5x

```
        for ( int n = 3; n <= 5; n++) {

                System.out.print(n + "x");

        }
```

Example 2. What does this loop display? 5 8 11

```
        for ( int k = 5; k < 14; k += 3 ){

                System.out.print( k + "  " );

        }
```

Example 3. What does this loop display? 25 18 11

```
        for ( int j = 25; j >= 10; j = j- 7 ){

                System.out.print( j + "  " );

        }
```

Warning. If the counter is declared in the for statement (which is usually the case), then it has

____

While Loops:

Three elements must be present with any while loop. They are

1) Initialization
2) Condition
3) Update

for(int i=0; i<= 5; i++)

For Loop:

The for loop contains the three elements of while loop in a single line.
The initialization is a statement that is always executed before the first pass through the loop, condition is tested before each pass through the loop and update is a statement executed at the end of each pass through the loop.

What is the output of the given code segment:

```
    int sum = 0;
    for (int k = 0; k < 10; k++)
        sum += k * k;
    System.out.println(k);
```

0
1
2
3
4
.
.
.
10

→ K values will be printed

K	sum
0	0
1	1
2	5
3	14
4	30
5	55
6	91
7	140
8	204
9	285

Syntax for while and for loop:

```
for(initilazation; condition; update)
{
    ....
}
```

```
initilazation;
while(condition)
{
      update
}
```

Compare while and for loop:

For loop typically used when the number of iteration is known.
While loop generally used when the number of iteration does not known.

Nested Iteration.

A loop within a loop is called a nested loop. The inner loop must complete all its iterations before the outer loop can continue.

k = 2

1. What is displayed?
1 2
1 2
1 2
	`for ( int k = 1; k < 4; k++ ){`
	`    for ( int n = 1; n < 3; n++ ){`
	`        System.out.print( n + "  " );`
	`    }`
	`    System.out.println();`
	`}`

2. What is displayed?
1 1
2 2
3 3
	`for ( int k = 1; k < 4; k++ ){`
	`    for ( int n = 1; n < 3; n++ ){`
	`        System.out.print( k + "  " );`
	`    }`
	`    System.out.println();`
	`}`

3. What is displayed?
5 6
6 7
7 8
 	
a | b | num
0 | 5 | 5
0 | 6 | 6
1 | 5 | 6
1 | 6 | 7
2 | 5 | 7
2 | 6 | 8
	`for ( int a = 0; a < 3; a++ ){`
	`    for ( int b = 5; b < 7; b++ ){`
	`    {   int num = a + b;`
	`        System.out.print( num + "  " );`
	`    }`
	`    System.out.println();`
	`}`

4. What is displayed?
1 2 3 4
2 3 4
3 4
 	
a | b
1 | 1 2 3 4
2 | 2 3 4
3 |
	`for ( int a = 1; a < 4; a++ ){`
	`    int b = a;`
	`    while ( b < 5 ){`
	`        System.out.print( b + "  " );`
	`        b++;`
	`    }`
	`    System.out.println();`
	`}`

Informal Code Analysis.

We can estimate algorithm efficiency by counting the number of times statements are executed.

Both code snippets do the same thing; they print the even numbers from 2 to 100. Let's compare them.

`for ( int n = 1; n <=100; n++ ){`
`    if ( n % 2 == 0 )`
`        System.out.println( n );`
`}`
	`for ( int k = 2; k <= 100; k += 2 )`
	`    System.out.println( k );`

int n = 1 is executed 1 time(s) n <= 100 is evaluated 100 time(s) n++ is executed 99 time(s) n%2 == 0 is evaluated 50 time(s) SOP is executed 50 time(s) Total 300	int k = 2 is executed 1 time(s) k <= 100 is evaluated 50 time(s) k += 2 is executed 49 time(s) SOP is executed 50 time(s) Total 150

What do you observe from comparing both the code segments above?

The second code segment is more effient as it directly increments by 2, eliminating the need for an if condition to check for even numbers.