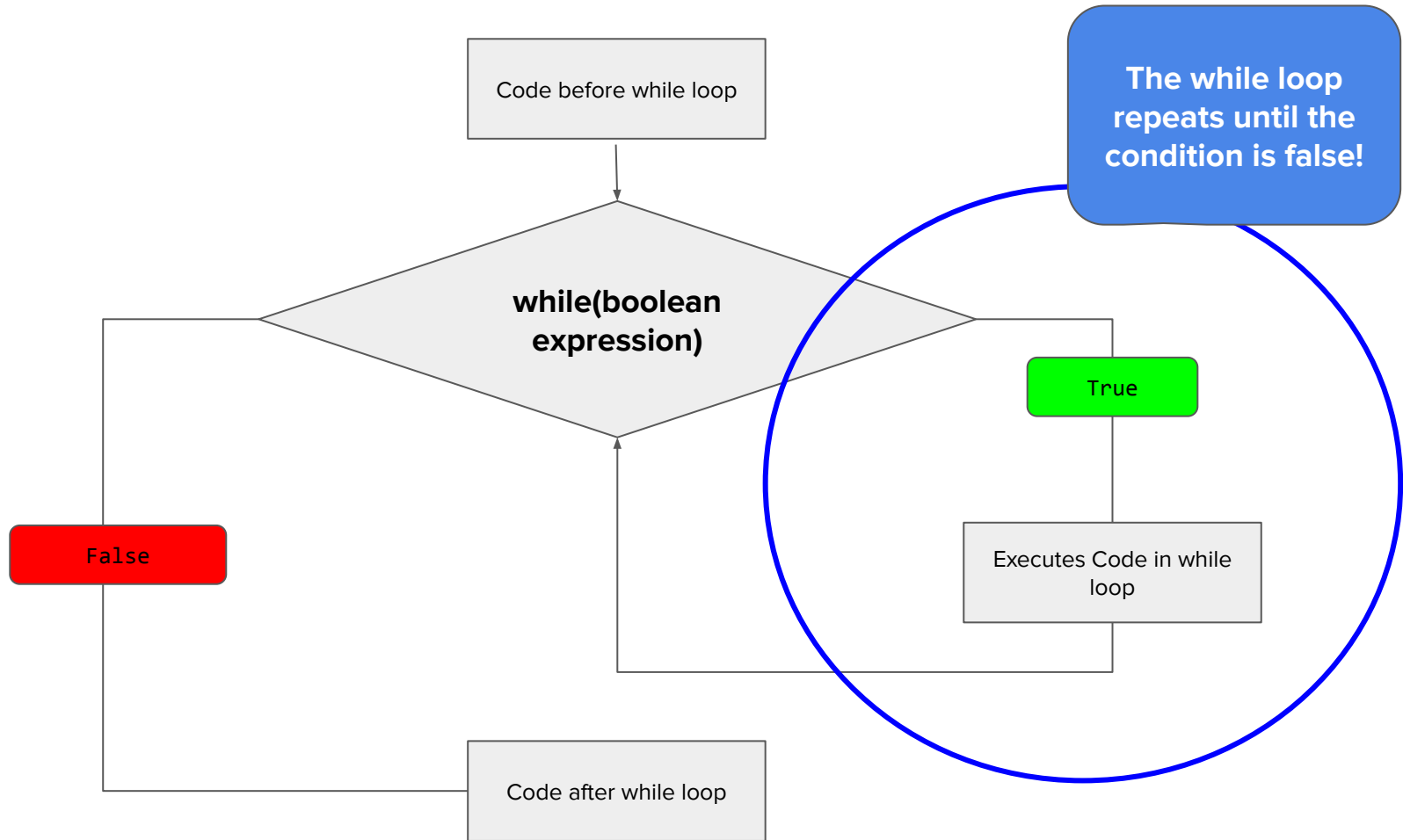


# AP Computer Science A: for *Loops*

# Recap: while Loop Flowchart

---



# for Loops

---

Another tool we can use to repeat code sequences is the **for loop**.

# for Loops

---

**for loops** allow us to repeat a set of statements a specific number of times!

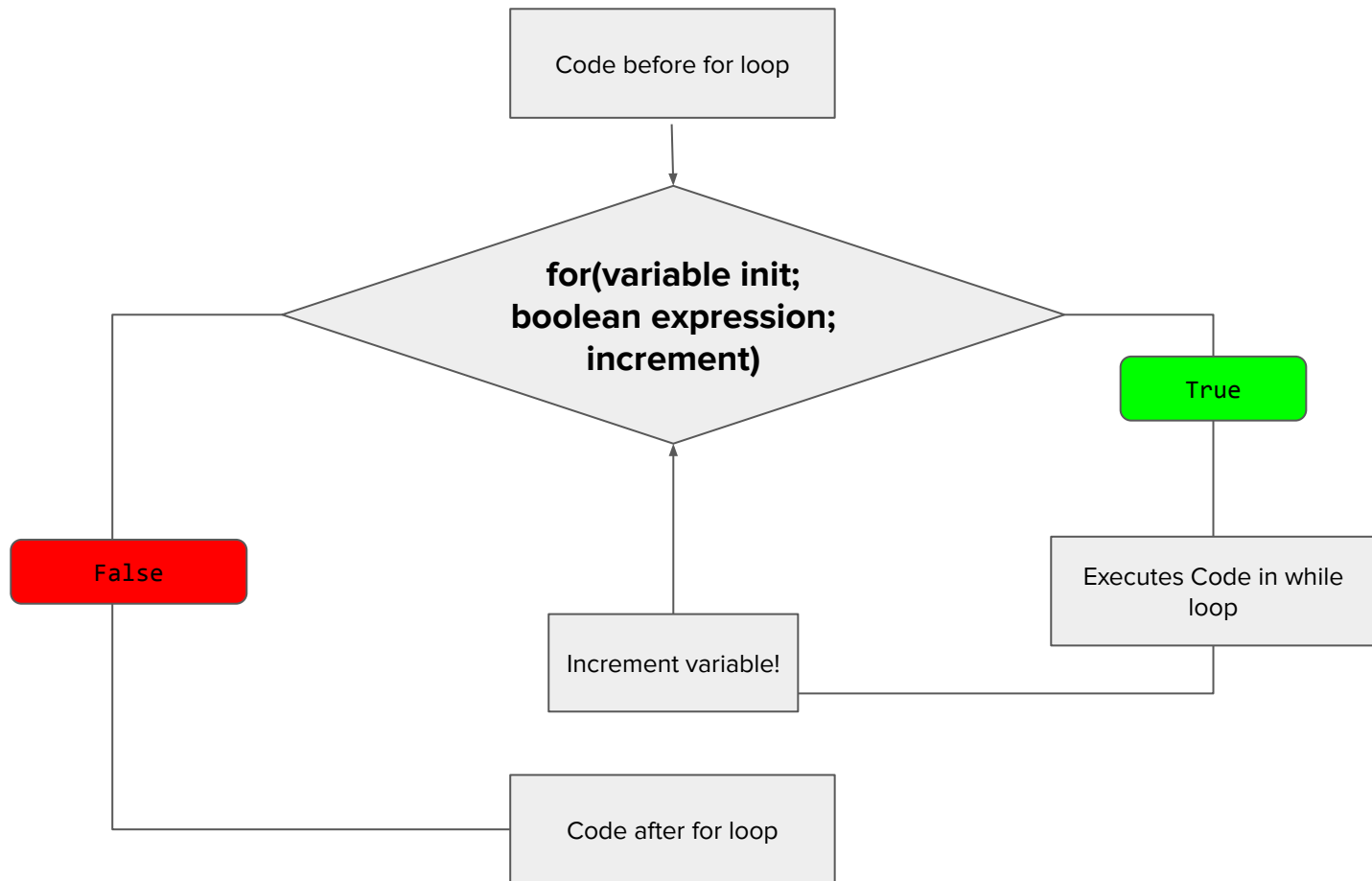
# for Loops

---

```
for(variable initialization; boolean expression; increment)
{
    //will execute if boolean is true, and until the boolean
    expression is false
}
```

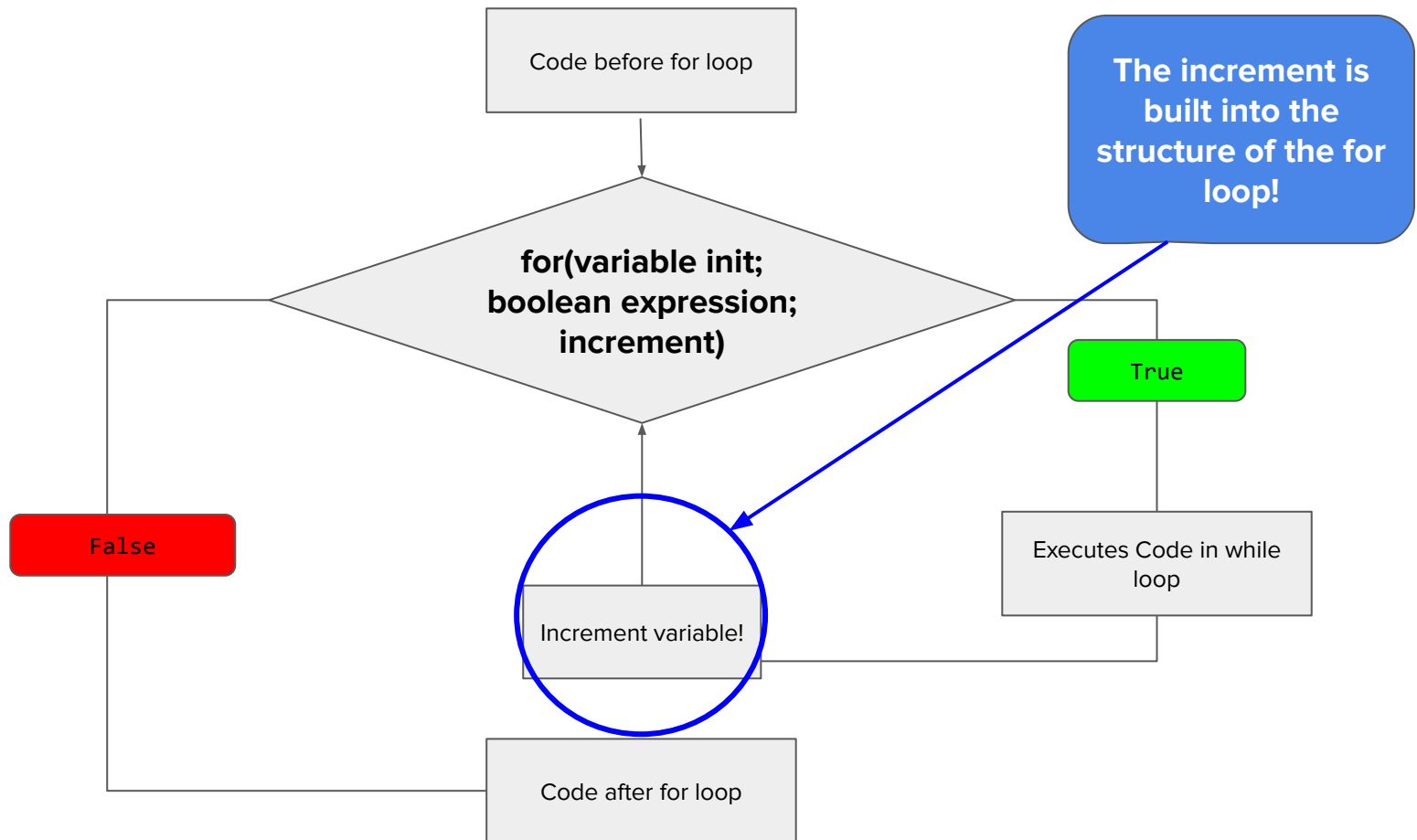
# for Loop Flowchart

---



# for Loop Flowchart

---



# for Loops

---

```
for(variable initialization; boolean expression; increment)
{
```



# for Loops

---

```
for(variable initialization; boolean expression; increment)
{
```

The initialization.  
This variable  
controls the for  
loop execution

```
for(int i = 0; boolean expression; increment)
{
```

# for Loops

---

```
for(variable initialization; boolean expression; increment)
{
```

If the expression is  
false, the for loop  
will not execute

```
for(int i = 0; i < 3; increment)
{
```

# for Loops

---

```
for(variable initialization; boolean expression; increment)  
{
```

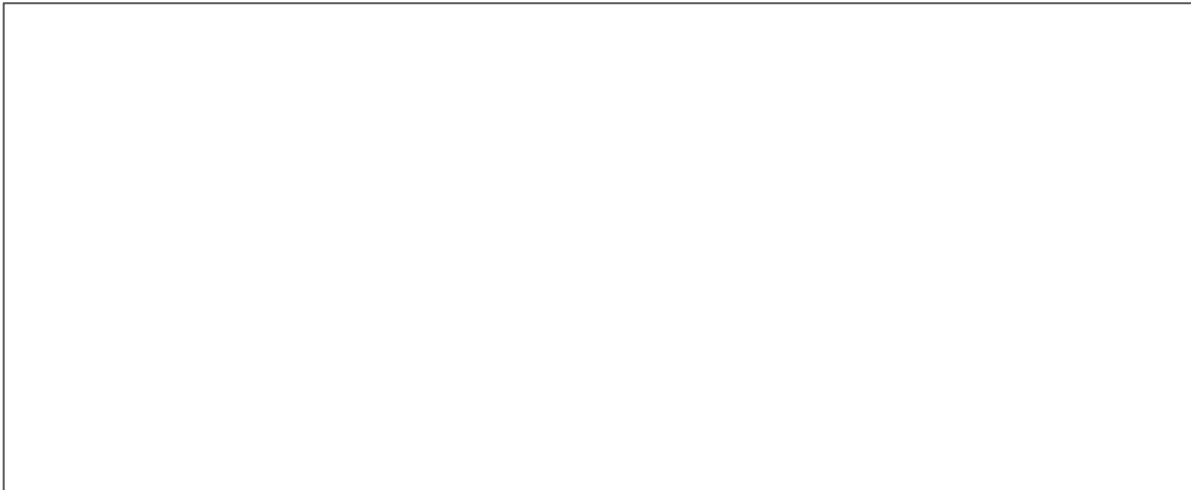
```
for(int i = 0; i < 3; i++)  
{
```

The increment changes the value of the loop control variable. It executes after each for loop execution

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```



# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 0

The loop control variable is initialized first, then tested against the boolean expression

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 0

The loop control variable is tested against the boolean expression. If true, then the code will execute!

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 0

0

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

0

i = 1

Once the loop is done executing, the increment is executed, changing the loop control variable's value



# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 1

The next call is  
back to the boolean  
expression, NOT  
the loop control  
initialization.

0

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 1

```
0  
1
```

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 2

```
0  
1
```

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 2

```
0  
1
```

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 2

```
0  
1  
2
```

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

i = 3

```
0  
1  
2
```

# for Loop Example

---

```
for(int i = 0; i < 3; i++)  
{  
    System.out.println(i);  
}
```

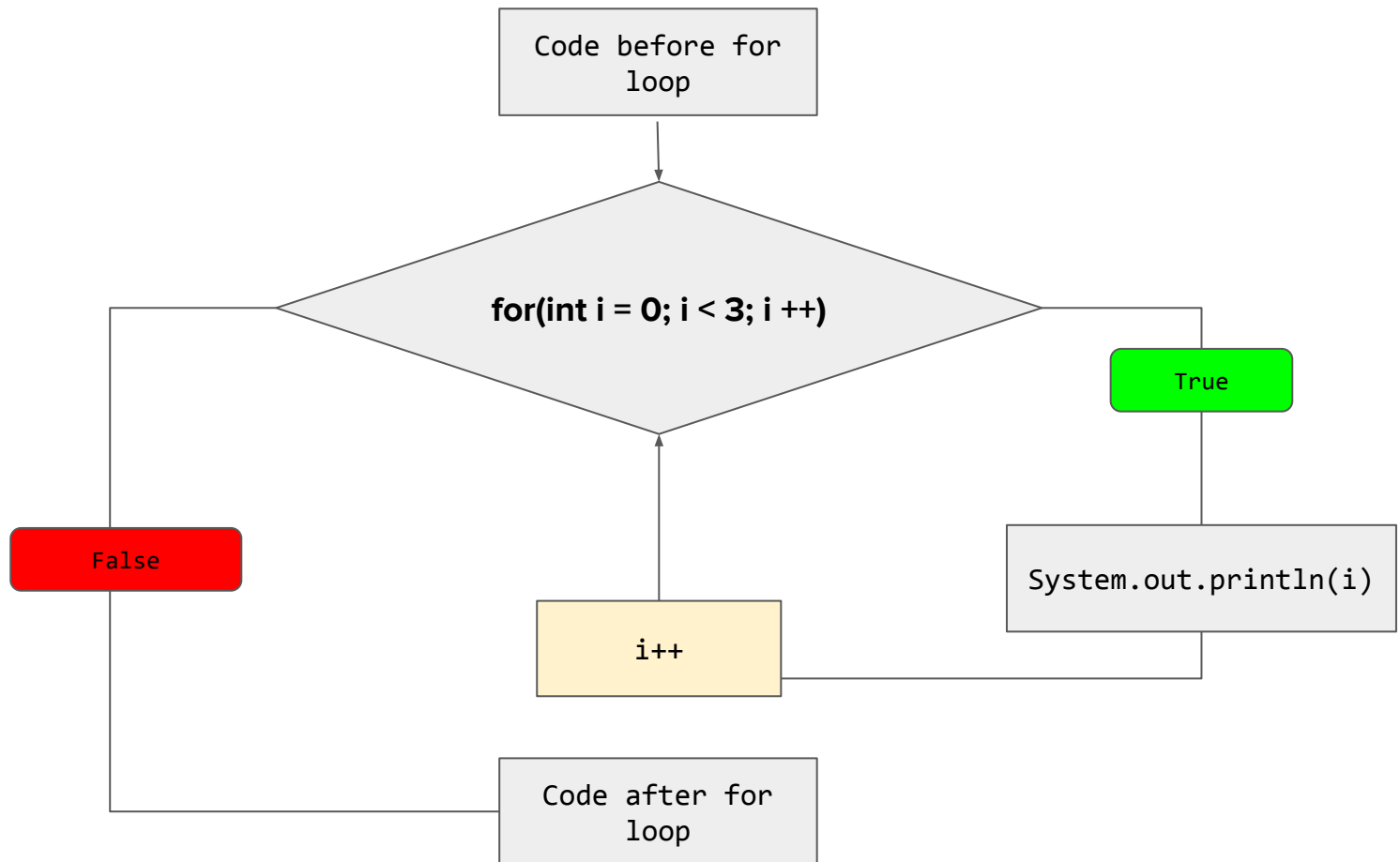
i = 3

Since the boolean expression is now false, the for loop stops executing!

0  
1  
2

# for Loop Flowchart

---





# for Loop Example

---

**The loop control variable doesn't have to start at zero:**

```
for(int i = 1; i < 3; i++)  
{  
    System.out.println(i);  
}
```

```
1  
2
```

# for Loop Example

---

**The loop control variable doesn't have to start at zero:**

```
int startingNum = 2;  
  
for(int i = startingNum; i < 3; i++)  
{  
    System.out.println(i);  
}
```

2

# for Loop Example

---

**The increment can also be manipulated:**

```
for(int i = 0; i < 20; i+=5)
{
    System.out.println(i);
}
```

```
0
5
10
15
```

```
for(int i = 3 ; i > 0; i--)
{
    System.out.println(i);
}
```

```
3
2
1
```

# for Loop Example

---

**Increment is easily modified using compound assignment operators!**

```
for(int i = 0; i < 20; i+=5)
{
    System.out.println(i);
}
```

```
for(int i = 3 ; i > 0; i--)
{
    System.out.println(i);
}
```



Use Compound  
Assignment  
Operators!

# for Loop Example

---

**The boolean expression can have a subtle impact on the output of a program!**

```
for(int i = 0; i < 20; i+=5)
{
    System.out.println(i);
}
```

```
0
5
10
15
```

```
for(int i = 0; i < 21; i+=5)
{
    System.out.println(i);
}
```

```
0
5
10
15
20
```

# for Loop Example

---

**If we wanted to include  $i = 20$ , we'd need to change the boolean expression:**

```
for(int i = 0; i <= 20; i+=5)
{
    System.out.println(i);
}
```

```
0
5
10
15
20
```

```
for(int i = 0; i < 21; i+=5)
{
    System.out.println(i);
}
```

```
0
5
10
15
20
```

# Off by One Error

---

When a for loop iterates one too few or one too many times, it's referred to as an **Off by One Error**.

# for Loop Example

---

**This program is meant to count from 1 - 10**

```
for(int i = 1; i < 10; i++)  
{  
    System.out.println(i);  
}
```





# for Loop Example

---

**This program is meant to count from 1 - 10**

```
for(int i = 1; i < 10; i++)  
{  
    System.out.println(i);  
}
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

This is an Off by One Error, because the user forgot to include 10 in their for loop!

# for Loop Example

---

**This program is meant to count from 1 - 10**

```
for(int i = 1; i <= 10; i++)  
{  
    System.out.println(i);  
}
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

# for Loops vs while Loops

---

for loops can be written as while loops, and vice versa!

# for Loops vs while Loops

---

```
for(int i = 0; i < 5; i++)  
{  
    System.out.println(i);  
}
```

```
0  
1  
2  
3  
4
```

```
int i = 0;  
while(i < 5)  
{  
    System.out.println(i);  
    i++;  
}
```

```
0  
1  
2  
3  
4
```

# for Loop Example

---

## Loop Control Variable

```
for(int i = 0; i < 5; i++)  
{  
    System.out.println(i);  
}
```

```
0  
1  
2  
3  
4
```

```
int i = 0;  
while(i < 5)  
{  
    System.out.println(i);  
    i++;  
}
```

```
0  
1  
2  
3  
4
```

# for Loop Example

---

## Boolean Expression

```
for(int i = 0; i < 5; i++)  
{  
    System.out.println(i);  
}
```

```
0  
1  
2  
3  
4
```

```
int i = 0;  
while(i < 5)  
{  
    System.out.println(i);  
    i++;  
}
```

```
0  
1  
2  
3  
4
```

# for Loop Example

---

## Increment

```
for(int i = 0; i < 5; i++)  
{  
    System.out.println(i);  
}
```

```
0  
1  
2  
3  
4
```

```
int i = 0;  
while(i < 5)  
{  
    System.out.println(i);  
    i++  
}
```

```
0  
1  
2  
3  
4
```

# for Loops vs. while Loops

---

What's the difference between the two?



# for Loops vs. while Loops

---

while loops work better for programs with an **undetermined** amount of iterations.

for loops are best for programs with a **predetermined** number of iterations

# while Loops = Undetermined

---

**The user enters a new password until they enter one that is 8 characters long. It's unclear how many attempts it will take for them to get it correct. while loops make this easy because there is no increment!**

```
Scanner input = new Scanner(System.in);
String password = "";
while(password.length() < 8)
{
    System.out.println("Enter a password");
    password = input.nextLine();
}
return password;
```

# for Loops = Determined

---

**A user wants to know which numbers between 0-100 are divisible by 3.  
This is easier to implement using a for loop because the increment is  
built in to the loop!**

```
for(int i = 0; i <= 100; i++)  
{  
    if(i%3 == 0)  
    {  
        System.out.println(i)  
    }  
}
```

# for Loops = Determined

---

**A user wants to know which numbers between 0-100 are divisible by 3.  
This is easier to implement using a for loop because the increment is  
built in to the loop!**

```
for(int i = 0; i <= 100; i++)  
{  
    if(i%3 == 0)  
    {  
        System.out.println(i)  
    }  
}
```

Can you think of another way  
to write this for loop without  
using an if statement?

# for Loops = Determined

---

**A user wants to know which numbers between 0-100 are divisible by 3.  
This is easier to implement using a for loop because the increment is  
built in to the loop!**

```
for(int i = 3; i <= 100; i+=3)
{
    System.out.println(i)
}
```

# Now It's Your Turn!

---

# Concepts Learned this Lesson

---

Term	Definition
<b>for Loops</b>	<pre>for(variable; boolean expression; increment) {     //code executes until false }</pre>
<b>Off by One Error</b>	When a for loop iteration is off by one too many or one too few.

# Standards Covered

---

- **(LO) CON-2.E** Represent iterative processes using a for loop.
- **(EK) CON-2.E.1** There are three parts in a for loop header: the initialization, the Boolean expression, and the increment. The increment statement can also be a decrement statement.
- **(EK) CON-2.E.2** In a for loop, the initialization statement is only executed once before the first Boolean expression evaluation. The variable being initialized is referred to as a loop control variable.
- **(EK) CON-2.E.3** In each iteration of a for loop, the increment statement is executed after the entire loop body is executed and before the Boolean expression is evaluated again.
- **(EK) CON-2.E.4** A for loop can be rewritten into an equivalent while loop and vice versa.
- **(EK) CON-2.E.5** “Off by one” errors occur when the iteration statement loops one time too many or one time too few.