Looping Structures
The Plan

While not everyone understands:

1. Motivate loops
2. For loops
3. While loops
4. Do-while loops
5. Equivalence
6. Application of Simulated Collision
7. Practice Problems
Motivation

Why loop?

Sometimes you need to do things again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and again, and finally you get tired of typing.
Motivation

Okay, so that's not all. You also loop in order to:

- Group repeatedly executed code for uniformity
- Make the number of repetitions easily changeable
- Repeat events which the number of executions is known only dynamically
- Combine with selection statements to make more complex algorithms
for Loop

```java
for(int i=0; i<10; i++)
{
    System.out.println(i);
}
```
for Loop

for (int i=0; i<10; i++)
{
    System.out.println(i);
}

Initialization: i=0
Condition: i<10
Update: i++
for Loop

for (int i=0; i<10; i++)
{
    System.out.println(i*0.1);
}

Initialization -> Condition -> Update

True -> False

Scale factor
for Loop

for (int i=0; i<10; i++)
{
    System.out.println(i*0.1+5);
}

Initialization
Condition
Update
True
False
Scale factor
Translation
while Loop

int i=0;
while(i<10)
{
    System.out.println(i);
    i++;
}
while Loop

```
int i=0;
while(i<10)
{
    System.out.println(i);
    i++;
}
```
while Loop

double i=0;
while(i<1)
{
    System.out.println(i);
    i+=0.1;
}

Why might this be a problem?
do-while Loop

```
int i = 0;
do
{    System.out.println(i);
    i++;
}while (i <= 10);
```
do-while Loop

```java
int i = 0; // Initialization

do {
    System.out.println(i); // Update
    i++; // True
}
while (i <= 10); // Condition
```
int i=0;
do
{
    System.out.println(i);
i++;
}while(i<=10);

Notice this semicolon
was not here in the while loop!
```java
int i = 0;
while (i < 10) {
    System.out.println(i);
    i++;
}

for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```
When to use which loop?

Is it known how many times the loop will execute prior to executing the loop body?
- Yes: for
- No: Is it important for the loop body to always execute at least once?
  - Yes: do-while
  - No: while
When to use which loop?

Real answer:

Use which ever structure is most convenient, because all loop structures can be represented as any other loop structure.

Why are there multiple loop structures then?

Simple answer – for the programmer’s convenience.
Application of Simulated Collision

double velocity=3;
double position=1;
double timeStep=0.1;
//simulate for about 5 seconds
double time=0;
while(time<5)
{
    System.out.println("("+time+","+position+")");
    time+=timeStep;
    position+=velocity*timeStep;
}
Application of Simulated Collision

double velocity0=3;  
double position0=1;  
double timeStep=0.1;  
//simulate for about 5 seconds  
double t=0;  
while(position0!=position1)  
{
    System.out.println("p0 is ("+t+", "+position0+")");
    System.out.println("p1 is ("+t+", "+position1+")");
    t+=timeStep;
    position0+=velocity0*timeStep;
    position1+=velocity1*timeStep;
}
Application of Simulated Collision

double velocity0=3;
double position0=1;
double timeStep=0.1;
//simulate for about 5 seconds
double t=0;
while(position0<position1)
{
    System.out.println("p0 is "+t+", "+position0+");
    System.out.println("p1 is "+t+", "+position1+");
t+=timeStep;
    position0+=velocity0*timeStep;
    position1+=velocity1*timeStep;
}

Problem fixed, right?
Application of Simulated Collision

double velocity0=-3; double velocity1=2;
double position0=10; double position1=1;
double timeStep=0.1;
//simulate for about 5 seconds
double t=0;
while(position0<position1)
{
    System.out.println("p0 is ("+t+", "+position0+")");
    System.out.println("p1 is ("+t+", "+position1+")");
    t+=timeStep;
    position0+=velocity0*timeStep;
    position1+=velocity1*timeStep;
}

What about now?
(notice velocity and position change)
Practice Problems

- Write a loop to print out from 10 to 100 inclusive counting by 10s

- Write a loop that starts with an arbitrary double x and divides it by 2 repeatedly until it is less than 1. Output the number of times the loop executed. What is being computed?

- Write a loop that sums the first x integers where x is a positive integer. Print out the results.

- Write a loop that takes an integer x starting with value 1 and doubles x so long as x is positive. Bonus question: why doesn’t this loop infinitely? Super Bonus question: why does it loop infinitely when x is a double?