

Finding a Gene in DNA

Finding All Genes in DNA

▶	Introduction	48 sec
▶	Conceptual Understanding	4 min
▶	While Loops	9 min
▶	While Loop Syntax and Semantics	3 min
▶	Coding While Loops	6 min
▶	Three Stop Codons	5 min
▶	Coding Three Stop Codons - Part I	7 min
▶	Coding Three Stop Codons - Part II	4 min
▶	Logical And / Or	8 min
▶	Coding And / Or	6 min
▶	Finding Multiple Genes	5 min
▶	Translating to Code	8 min
📖	Programming Exercise: Finding Many Genes	10 min
★	Practice Quiz: Finding All Genes in DNA	4 questions

Debugging Code

Using the StorageResource Class

Review

As long as...

```
while (x < y) {
    System.out.println(x);
    x = x + 3;
}
```

- While loops
 - Generally for "As long as (condition)"

a step in your algorithm like,
as long as some condition is true.

Duke UNIVERSITY

While Loop Syntax and Semantics

Have a question? Discuss this lecture in the week forums.



Interactive Transcript

Search Transcript

English ▾

0:03

To implement our improved gene-finding algorithm, you need to learn about while loops, the code that generally comes up when you have [a step in your algorithm like, as long as some condition is true](#). You can see an example of a while loop here. Let's look at this example to delve into the syntax and semantics of the while loop.

0:23

All while loops start out with the keyword while.

0:27

Then, they have some condition, sometimes called a guard, in parentheses. This is the condition that you want to check to see if you should continue doing the loop.

0:39

Then you have the loop body in curly braces. These are the statements that should be executed each iteration of the loop.

0:47

Let's walk through this example to understand the semantics of the loop. What it means? For example, let us assume that x was previously declared and initialized to zero. And y was initialized to 7. When execution reaches the while loop, the first thing that happens is Java evaluates the condition, is x less than y.

1:11

In this particular case, that means is 0, less than 7? 0 is less than 7, so the condition evaluates to true.

1:20

Since the condition is true, execution will enter the loop body and continue executing statements there. The first statement prints x which is 0. So, we would output the line 0. The second statement sets x to x + 3 which would update the value of x to be 3. Now, execution has reached the end of the loop body. The close calibrates marks its end.

1:46

Then execution goes back up to the start of the loop, and we are right back where we started, but the variables have different values. X is 3 instead of 0. We continue following the same rules. Evaluate the condition, 3 is less than 7. The condition is true so we enter the loop body. Print x which is 3, and update x to be 6. Now we've reach the end of the loop again so we go back up to the start of it. And again follow the same rules, check the condition, it is true so we enter the loop body print x which is 6. Update x to be 9, and now we have reached the end of the loop body again, so we go back to the start. Yet again we follow the same rules. Now when we evaluate the condition, something different happens. Is 9 less than 7? No, so, the condition is false. Instead of entering the loop body, we go past it. We would then continue executing whatever statements come after the loop.

2:52

Great, now you've learned the basics of while loops, their syntax, the grammatical rules for writing them, and their semantics, their meaning. Now you're ready to go code your improved gene finding algorithm.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?