

Introduction to the Course

Implementing the Caesar Cipher

Breaking the Caesar Cipher

Introduction	5 min
Arrays	9 min
Random Numbers and Arrays	11 min
Counting with Arrays	10 min
Developing an Algorithm	5 min
Summary	3 min
Programming Exercise: Breaking the Caesar Cipher	10 min
Practice Quiz: Breaking the Caesar Cipher	11 questions

Object Oriented Caesar Cipher

Review

File saved

So we'll just copy this whole thing.

```
public void simpleSimulate(int rolls){
    Random rand = new Random();
    int twos = 0;
    int twelves = 0;

    for(int k=0; k < rolls; k++){
        int d1 = rand.nextInt(6) + 1;
        int d2 = rand.nextInt(6) + 1;
        if (d1 + d2 == 2){
            twos += 1;
        }
        else if (d1 + d2 == 12){
            twelves += 1;
        }
    }

    System.out.println("2's=\t" + twos + "\t" + 100.0 * twos/rolls);
}
```

Random Numbers and Arrays

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

Interactive Transcript

Search Transcript

English

0:03  
Computers are very good at modeling and simulation, in part because computers can perform billions of mathematical operations every second. Often simulation and modeling relies on generating random numbers. In this coding example we'll see a simple use of arrays to help determine how really random the java.util.Random class is. Computers don't use natural random phenomena, but model randomness of what's called pseudo-randomness. Using mathematics to model what might be random events in nature. In this example, we will simulate rolling a pair of dice. Many times. By generating random numbers. We would like to know how many twos do we roll, how many sevens do we roll, we will count how many times each type of roll occurs. So I've got this code right here called Simple Stimulate and I can run it and say how many rolls I want to do, and right now, it counts how many times I get two, and how many times I get 12. So let's run it and see what it does. So we'll come over here.

1:11  
It's already compiled.

1:14  
I'll create the object.

1:18  
And we'll run simple simulate. And here we actually get to choose how many times, so I'll say 10,000 times. And we'll just run it.

1:29  
And you can see I got two 298 times, and I got 12 271 times.

1:37  
It'd be nice to run it and count all the possible rolls you could get from two to 12, so let's look at doing that.

1:50  
So what I'm gonna do is I'm gonna cut and paste this and we'll write a different method. [So we'll just copy this whole thing.](#)

2:01  
And then we'll put it up here.

2:14  
And we'll just call this one Simulate.

2:20  
There we go and so now we're gonna have to modify it so the first thing is we have generate a rand and then we need counters for two, three, four, five, six, seven, eight, nine, 10, 11, 12. That's a lot of counters. So what I would like to do instead is to use an array of counters, so that's what we'll do. So we'll replace this here with an array. So we want an array of counters so it'll be int array instead. We'll have to give it a name. We'll call it counts.

2:57  
And then we'll have to create the array.

3:01  
It's an integer array and then what size is it gonna be? So, two through 12, we need at least 11 counters but it would be kind of nice if we set it up so that when we threw a four, we would count four in the fourth slot of the array. So what we'll do is we'll create the array with size 13 which will allow us to use from zero to 12 and really we'll just use two through 12 and we'll ignore the zero and one counters.

3:33  
So I'll make my array of size 13 I don't need this 12 counter anymore. So we'll get rid of that.

3:41  
Okay. So now we're gonna have to adjust the code to handle using this array of counters.

3:48  
Okay. So we're still rolling whatever number of times we typed in for rolls. So that's the same. We're still throwing two dice, d1 and d2, and we are gonna to add them together. But, as you can see here, we have this if statement, where we have to check and see, was it two? Was it 12? And we could add all these other if statements. Was it three, four, five and so on? But instead, we can just notice, we just add the two together. That's also the index position of the counter and that's all we have to do. So we can get rid of this if statement.

4:23  
And instead we're just gonna add code, so we're gonna update or counter that position by one. So our array is called counts, we're gonna add in the d1 + d2 slot, we're just gonna add 1, + = 1. That's it, that replaced a huge if statement.

4:49  
Let's see, so now we're gonna output. So you can see we have two lines of output for twos and 12s. What we're gonna do instead is it would be nice if we just had one print statement and we used it over and over again for each position in our counter array. So we're gonna add a four loop here.

5:12  
And remember, our counter started at two. So we'll have our k start at two. Cuz the first, lowest thing you can get is two ones, from each dice. And that's two. So that'll be the lowest, and then it'll be k less than or equal to 12.

5:34  
Because we're gonna go from two to 12. And then we'll update it the same way, k++.

5:46  
Okay and then we just need one of these output statements to modify.

5:51  
Or I'll just, I'll just type a new one, will just say here so System.out.println, and what we wanna do is we wanna print k cuz that'll be the counter, so k for two through 12, right. So we will print k and then we'll print, we can still do the quotes.

6:16  
Equals and a tab. \t is the tab.

6:22  
Plus, and then we wanna know, how many twos were there, or how many tens were there, etc. And that's just counts k. That's how many of type k there were that we counted.

6:42  
Plus, we can add another tab.

6:50  
And then we can add the rest of this, 100 times counts of k divided by rolls and I'll just put that in parenthesis. I guess it doesn't matter.

7:19  
We do need to add a semicolon at the end.

7:32  
And then I can get rid of these two print statements here.

7:37  
Oops, there we go. Okay so I think we have everything now, so we've got, we now have our loop that prints for each one, two through 12 what it does. Let's see if we have it compiling.

8:03  
Oh, we're missing one right paren, right here. So we can add that.

8:10  
Oops, we'll add it and then put our semicolon. There we go. Let's try compiling it again. And it says it compiles. There's our code. So now let's run it and see what happens.

8:28  
It's already compiled. We can create our object.

8:32  
And then we can run it, simple simulate, this is called simulate.

8:37  
And then we'll run it again, let's run it 100,000 times. There it goes, okay, so now you can see we're getting lots of numbers. Lots of counts for twos, threes, fours, and so on.

8:51  
How do you know it actually works? We just got lots of numbers. It looks pretty good in the sense that if you wrote seven you're going to get that more likely than any other thing because seven is two and five, one and six, three and four and it does have the highest number but to really be sure what we'll do is we'll run a simple example and we'll add print statement. Like, let's run it ten times and let's see what the actual roles are and we can see if our counters are really working correctly. So, let's do that real quick. We'll go back to the code here.

9:25  
So, this is really adding just for testing it to make sure that the counters are really the counters we think they are. So whenever we roll a dice, which is right here, I'm just gonna add a print statement here.

9:43  
So what I'll do is I'll print out roll is d1.

9:55  
Plus.

10:01  
D2 which equals.

10:10  
D1 + d2.

10:16  
So, that's all we're doing, we're just printing out the roll. Then we're just gonna run it a very small number of times, so that we can see our counters are actually working correctly. So, we'll compile that and that worked. So, let's run it now.

10:36  
Okay, so I'm just gonna run it ten times. Throw the dice ten times. And let's see what we get.

10:44  
So, we got it looks like one, two, three sevens and if you look, it counted seven three times. We got one, two, three tens and it counted ten three times. We got a four once, we got a four. So it looks like it is counting them correctly. So this is way you can kind of test it on a small amount of data and that seems to work. All right so anyway, I hope you enjoyed dice rolling. Thank you.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

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