Arrays

Lecture 14, Tue Feb 28 2006

based on slides by Kurt Eiselt

http://www.cs.ubc.ca/~tmm/courses/cpsc111-06-spr
News

- Assignment 2
  - corrections to ASCIIArtiste.java posted
  - definitely read WebCT bboards!
Reading

■ This week: 8.1, 8.5-8.7, topics 6.3 and 6.4
public class WhileDemo
{
    public static void main (String[] args)
    {
        int limit = 3;
        int counter = 1;
        while (counter <= limit)
        {
            System.out.println("The square of " + counter + " is " + (counter * counter));
            counter = counter + 1;
        }
        System.out.println("End of demonstration");
    }
}

■ while version
Recap: For Loop Example

```java
public class ForDemo {
    public static void main (String[] args) {
        for (int counter = 1; counter <= 3; counter = counter + 1) {
            System.out.println("The square of "+ counter + " is "+ (counter * counter));
        }
        System.out.println("End of demonstration");
    }
}
```

- **for version**
public class DoDemo
{
    public static void main (String[] args)
    {
        int limit = 3;
        int counter = 1;
        do
        {
            System.out.println("The square of " + counter + " is " + (counter * counter));
            counter = counter + 1;
        } while (counter <= limit);
        System.out.println("End of demonstration");
    }
}

**do version**
Recap: For Statement

for (initialization; boolean expression; increment)
  body

- **Body** of loop can be
  - single statement
  - whole block of many statements in curly braces

- **Control flow**
  - first time through: initialization
  - boolean expression evaluated
  - if expression true, body executed; if false, end
  - increment processed
  - boolean expression evaluated
  - if true, body executed; if false, end
  - ....
Recap: For Versus While Statement

- **how for statement works**
  - initialization
  - boolean expression
    - true
    - statement
    - increment
    - false

- **how while statement works**
  - boolean expression
  - true
  - statement
  - false

- flowcharts can be somewhat deceptive
  - need initialization and incrementing/modifying in while loop too
  - although syntax does not require it in specific spot
Recap: `do` Statement

- Body always executed at least once
- Order of four things can change, but need them all
Objectives

- More practice with loops
- Understand when and how to use arrays
  - and loops over arrays
Flipping Coins

- Did **while** version last time
- Let's try **for** version now
Keeping Track of Things

Cans of pop sold this month

185
92
370
485
209
128
84
151
32
563

What’s the gross income?
What’s the net profit?
Is Bubba stealing loonies?
Keeping Track of Things

Cans of pop sold this month

185
92
370
485
209
128
84
151
32
563

In other words, how can I organize the data above in my computer so that I can access it easily and do the computations I need to do?
Answer: Arrays

- use **arrays**: common programming language construct
  - grouping related data items together
  - meaningful organization such that each individual data item can be easily retrieved or updated

Cans of pop sold this month:

- 185
- 92
- 370
- 485
- 209
- 128
- 84
- 151
- 32
- 563
Answer: Arrays

- use **arrays**: common programming language construct
  - grouping related data items together
  - meaningful organization such that each individual data item can be easily retrieved or updated
- collection of variables
  - all of same type
  - share common name
- each variable holds single value
Using Arrays

- Collection of variables has single name
- how do we access individual values?
Using Arrays

- Collection of variables has single name
  - how do we access individual values?
- Each value stored at unique numbered position
  - number called index of array element
    - aka subscript
- cansSold name of this array
  - holds 10 values
Using Arrays

- To access individual value in array
  - use array name followed by pair of square brackets
  - inside brackets, place index of array element we want to access
- Reference to array element allowed anywhere that variables can be used
- Example:

  ```java
  System.out.println(cansSold[4]);
  ```

- Prints value 209
Array Declaration and Types

- Just like ordinary variable, must
  - declare array before we use it
  - give array a type

- Since `cansSold` contains integers, make integer array:
  ```
  int[] cansSold = new int[10]
  ```

- Looks like variable declaration, except:
Array Declaration and Types

- Just like ordinary variable, must
  - declare array before we use it
  - give array a type
- Since `cansSold` contains integers, make integer array:
  ```java
  int[] cansSold = new int[10]
  ```
- Looks like variable declaration, except:
  - empty brackets on the left tell Java that `cansSold` is an array...
Array Declaration and Types

- Just like ordinary variable, must
  - declare array before we use it
  - give array a type
- Since `cansSold` contains integers, make integer array:

```
int[] cansSold = new int[10]
```

- Looks like variable declaration, except:
  - empty brackets on the left tell Java that `cansSold` is an array...
  - the number in the brackets on the right tell Java that array should have room for 10 elements when it's created
Array Declaration and Types

- Just like ordinary variable, must
  - declare array before we use it
  - give array a type

- Since `cansSold` contains integers, make integer array:

```
int[10] cansSold = new int[10]
```

- Looks like variable declaration, except:
  - empty brackets on the left tell Java that
cansSold is an array...
  - the number in the brackets on the right tell Java that array should have
  room for 10 elements when it's created
  - **DO NOT** put size of array in brackets on the left
Array Declaration and Types

- Just like ordinary variable, must declare array before we use it
  - give array a type
- Since cansSold contains integers, make integer array:

```
int[10] cansSold = new int[10]
```
- Looks like variable declaration, except:
  - empty brackets on the left tell Java that cansSold is an array...
  - the number in the brackets on the right tell Java that array should have room for 10 elements when it's created
- DO NOT put size of array in brackets on the left

<table>
<thead>
<tr>
<th>cansSold</th>
</tr>
</thead>
<tbody>
<tr>
<td>0  185</td>
</tr>
<tr>
<td>1   92</td>
</tr>
<tr>
<td>2  370</td>
</tr>
<tr>
<td>3  485</td>
</tr>
<tr>
<td>4  209</td>
</tr>
<tr>
<td>5  128</td>
</tr>
<tr>
<td>6   84</td>
</tr>
<tr>
<td>7  151</td>
</tr>
<tr>
<td>8   32</td>
</tr>
<tr>
<td>9  563</td>
</tr>
</tbody>
</table>
Array Declaration and Types

```java
public class ArrayTest1
{
    public static void main(String[] args)
    {
        final int ARRAYSIZE = 10;
        int[] cansSold = new int[ARRAYSIZE];

        cansSold[0] = 185;
        cansSold[1] = 92;
        cansSold[2] = 370;
        cansSold[3] = 485;
        cansSold[5] = 128;
        cansSold[6] = 84;
        cansSold[7] = 151;
        cansSold[8] = 32;
        cansSold[9] = 563;

        // do useful stuff here
        System.out.println("Element 4 is " + cansSold[4]);
    }
}
```
**Array Declaration and Types**

```java
public class ArrayTest2 {
    public static void main(String[] args) {
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        // do useful stuff here
        System.out.println("Element 4 is "+ cansSold[4]);
    }
}
```

- Can also use **initializer list**
- Right side of declaration does not include type or size
  - Java figures out size by itself
- Types of values on right must match type declared on left
- Initializer list may only be used when array is first declared
### Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

<table>
<thead>
<tr>
<th>cansSold</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>185</td>
</tr>
<tr>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>370</td>
</tr>
<tr>
<td>3</td>
<td>485</td>
</tr>
<tr>
<td>4</td>
<td>209</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>151</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>563</td>
</tr>
</tbody>
</table>
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```java
public class ArrayTest3 {
    // implementation goes here
}
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```
public class ArrayTest3 {
    public static void main(String[] args) {
        public static void main(String[] args) {
        }
    }
    }
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
    }
}
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

<table>
<thead>
<tr>
<th>cansSold</th>
</tr>
</thead>
<tbody>
<tr>
<td>0   185</td>
</tr>
<tr>
<td>1   92</td>
</tr>
<tr>
<td>2   370</td>
</tr>
<tr>
<td>3   485</td>
</tr>
<tr>
<td>4   209</td>
</tr>
<tr>
<td>5   128</td>
</tr>
<tr>
<td>6   84</td>
</tr>
<tr>
<td>7   151</td>
</tr>
<tr>
<td>8   32</td>
</tr>
<tr>
<td>9   563</td>
</tr>
</tbody>
</table>

```java
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0;
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length;)
        {
        }
    }
}
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```
cansSold
0  185
1   92
2 370
3 485
4 209
5 128
6  84
7 151
8  32
9 563
```

```java
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209,
                          128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
```
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209,
                          128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
            totalCans = totalCans + cansSold[i];
    }
}
```
Using Arrays and Loops

- Write program to
  - create array
  - find total number of cans sold
  - print result

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209,
                          128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

Tracing Arrays and Loops

```java
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

`totalCans` 0
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

TotalCans = 0

CansSold.length = 10

CansSold =

<table>
<thead>
<tr>
<th></th>
<th>185</th>
<th>92</th>
<th>370</th>
<th>485</th>
<th>209</th>
<th>128</th>
<th>84</th>
<th>151</th>
<th>32</th>
<th>563</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>185</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>370</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>485</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>209</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>151</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>563</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tracing Arrays and Loops

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```
```java
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```
Tracing Arrays and Loops

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```
Tracing Arrays and Loops

cansSold.length = 10

cansSold
0 185
1  92
2 370
3 485
4 209
5 128
6  84
7 151
8  32
9 563

public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209,
                         128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

totalCans = 185
```java
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

- Is i < 10?  
  - yes, 1 < 10
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        } 
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
public class ArrayTest3  
{
   public static void main(String[] args)  
   {
      int totalCans = 0;
      int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

      for (int i = 0; i < cansSold.length; i++)  
      {
         totalCans = totalCans + cansSold[i];
      }

      System.out.println("We've sold " + totalCans + " cans of pop");
   }
}
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold "+ totalCans + " cans of pop");
    }
}
```
# Tracing Arrays and Loops

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

<table>
<thead>
<tr>
<th>cansSold.length</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>cansSold</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>185</td>
</tr>
<tr>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>370</td>
</tr>
<tr>
<td>3</td>
<td>485</td>
</tr>
<tr>
<td>4</td>
<td>209</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>151</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>563</td>
</tr>
</tbody>
</table>

| i    | 3   |
| totalCans | 647 |
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold "+totalCans+" cans of pop");
    }
}
Tracing Arrays and Loops

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

And so on...
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

And so on...
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

And so on...
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

And so on...
Tracing Arrays and Loops

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }
        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

And so on...
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

And so on...
public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209,
                         128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++)
        {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold "+ totalCans + " cans of pop");
    }
}

cansSold.length 10

cansSold
0 185
1 92
2 370
3 485
4 209
5 128
6 84
7 151
8 32
9 563

i 10

totalCans 2299
Tracing Arrays and Loops

canSold.length: 10

canSold

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>185</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>370</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>485</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>209</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>151</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>563</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

public class ArrayTest3
{
    public static void main(String[] args)
    {
        int totalCans = 0;
        int[] canSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < canSold.length; i++)
        {
            totalCans = totalCans + canSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}

i: 10

totalCans: 2299

Is i < 10?

- no, 10 not < 10
Tracing Arrays and Loops

```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i < cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

- "We've sold 2299 cans of pop" printed out
```java
public class ArrayTest3 {
    public static void main(String[] args) {
        int totalCans = 0;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};

        for (int i = 0; i <= cansSold.length; i++) {
            totalCans = totalCans + cansSold[i];
        }

        System.out.println("We've sold " + totalCans + " cans of pop");
    }
}
```

What would happen if we made this little change?
Tracing Arrays and Loops

What would happen if we made this little change?

```
java.lang.ArrayIndexOutOfBoundsException: 10
```
Something To Remember

- Array `cansSold` created with 10 elements
  - Indices (plural of index) are 0 through 9
  - In general, array of size n will have indices ranging from 0 through n-1
  - When you number things, you're used to beginning with 1
  - Computer folks begin with 0
    - leads to "off by one" errors, even among computer veterans
import java.util.Scanner;

public class ArrayTest3b {
    public static void main(String[] args) {
        final int ARRAYSIZE = 10;
        int[] cansSold = new int[ARRAYSIZE];
        Scanner scan = new Scanner(System.in);

        for (int i = 0; i < cansSold.length; i++) {
            System.out.print("Enter machine "+(i+1));
            cansSold[i] = scan.nextInt();
        }

        // do useful stuff here
        System.out.println("Element 4 is "+cansSold[4]);
    }
}

<table>
<thead>
<tr>
<th>cansSold</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 185</td>
</tr>
<tr>
<td>92</td>
</tr>
<tr>
<td>370</td>
</tr>
<tr>
<td>485</td>
</tr>
<tr>
<td>209</td>
</tr>
<tr>
<td>128</td>
</tr>
<tr>
<td>84</td>
</tr>
<tr>
<td>151</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>563</td>
</tr>
</tbody>
</table>
Averaging Loop Example

- Let's say we want to write a program that prints average of values in some arbitrarily large array
  - like the one to the left called numbers

- Will require loop

- Simple task for looping in the context of an array
  - how will we make this happen?
PrintMax Loop Example

- Now instead of average, we want to find and print maximum value from some arbitrarily large array
  - Similar loop, but with some extra tweaks.
Histogram Loop Example

Now use same data as basis for histogram

Write one loop to look at value associated with each row of array

- for each value print a line with that many asterisks
- For example, if program reads value 6 from the array, should print line of 6 asterisks
  - Program then reads the value 8, prints a line of 8 asterisks, and so on.

Need outer loop to read individual values in the array

Need inner loop to print asterisks for each value

<table>
<thead>
<tr>
<th>numbers</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>6</td>
<td>8</td>
<td>11</td>
<td>18</td>
<td>20</td>
<td>17</td>
<td>14</td>
<td>10</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>******</th>
<th>*******</th>
<th>**********</th>
<th>****************</th>
<th>****************</th>
<th>****************</th>
<th>*************</th>
<th>**********</th>
<th>*****</th>
<th>**</th>
</tr>
</thead>
</table>
# Storing Different Data Types

cansSold

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>185</td>
</tr>
<tr>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>370</td>
</tr>
<tr>
<td>3</td>
<td>485</td>
</tr>
<tr>
<td>4</td>
<td>209</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>151</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>563</td>
</tr>
</tbody>
</table>
Storing Different Data Types

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 185</td>
<td>0 201.25</td>
</tr>
<tr>
<td>1 92</td>
<td>1 100.50</td>
</tr>
<tr>
<td>2 370</td>
<td>2 412.75</td>
</tr>
<tr>
<td>3 485</td>
<td>3 555.25</td>
</tr>
<tr>
<td>4 209</td>
<td>4 195.00</td>
</tr>
<tr>
<td>5 128</td>
<td>5 160.00</td>
</tr>
<tr>
<td>6 84</td>
<td>6 105.00</td>
</tr>
<tr>
<td>7 151</td>
<td>7 188.75</td>
</tr>
<tr>
<td>8 32</td>
<td>8 40.00</td>
</tr>
<tr>
<td>9 563</td>
<td>9 703.75</td>
</tr>
</tbody>
</table>

Could use two arrays of same size but with different types
Storing Different Data Types

- Write program to compare what's been collected from each machine vs. how much should have been collected?

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 185</td>
<td>0 201.25</td>
</tr>
<tr>
<td>1 92</td>
<td>1 100.50</td>
</tr>
<tr>
<td>2 370</td>
<td>2 412.75</td>
</tr>
<tr>
<td>3 485</td>
<td>3 555.25</td>
</tr>
<tr>
<td>4 209</td>
<td>4 195.00</td>
</tr>
<tr>
<td>5 128</td>
<td>5 160.00</td>
</tr>
<tr>
<td>6 84</td>
<td>6 105.00</td>
</tr>
<tr>
<td>7 151</td>
<td>7 188.75</td>
</tr>
<tr>
<td>8 32</td>
<td>8 40.00</td>
</tr>
<tr>
<td>9 563</td>
<td>9 703.75</td>
</tr>
</tbody>
</table>

Could use two arrays of same size but with different types
Storing Different Data Types

- Write program to compare what's been collected from each machine vs. how much should have been collected?

```
public class ArrayTest4
{
    public static void main(String[] args)
    {
        double expected;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        double[] cashIn = {201.25, 100.50, 412.75, 555.25, 195.00, 160.00, 105.00, 188.75, 40.00, 703.75};
        for (int i = 0; i < cansSold.length; i++)
        {
            expected = cansSold[i] * 1.25;
            System.out.println("Machine "+ (i + 1) + " off by "+ (expected - cashIn[i]));
        }
    }
}
```
Storing Different Data Types

- Write program to compare what's been collected from each machine vs. how much should have been collected?

```
public class ArrayTest4
{
    public static void main(String[] args)
    {
        double expected;
        int[] cansSold = {185, 92, 370, 485, 209, 128, 84, 151, 32, 563};
        double[] cashIn = {201.25, 100.50, 412.75, 555.25, 195.00, 160.00, 105.00, 188.75, 40.00, 703.75};
        for (int i = 0; i < cansSold.length; i++)
        {
            expected = cansSold[i] * 1.25;
            System.out.println("Machine "+(i+1)+" off by "+(expected-cashIn[i]));
        }
    }
}
```

Could use two arrays of same size but with different types

What happens when we run the program?
## Storing Different Data Types

### cansSold vs cashIn

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 185</td>
<td>0 201.25</td>
</tr>
<tr>
<td>1 92</td>
<td>1 100.50</td>
</tr>
<tr>
<td>2 370</td>
<td>2 412.75</td>
</tr>
<tr>
<td>3 485</td>
<td>3 555.25</td>
</tr>
<tr>
<td>4 209</td>
<td>4 195.00</td>
</tr>
<tr>
<td>5 128</td>
<td>5 160.00</td>
</tr>
<tr>
<td>6 84</td>
<td>6 105.00</td>
</tr>
<tr>
<td>7 151</td>
<td>7 188.75</td>
</tr>
<tr>
<td>8 32</td>
<td>8 40.00</td>
</tr>
<tr>
<td>9 563</td>
<td>9 703.75</td>
</tr>
</tbody>
</table>

Somebody has been stealing from the machines after all! We need an anti-theft plan…
# Arrays With Non-Primitive Types

- Great if you're always storing primitives like integers or floating point numbers.
  - What if we want to store String types too?
  - Remember that String is an object, not a primitive data type.

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>201.25</td>
</tr>
<tr>
<td>1</td>
<td>100.50</td>
</tr>
<tr>
<td>2</td>
<td>412.75</td>
</tr>
<tr>
<td>3</td>
<td>555.25</td>
</tr>
<tr>
<td>4</td>
<td>195.00</td>
</tr>
<tr>
<td>5</td>
<td>160.00</td>
</tr>
<tr>
<td>6</td>
<td>105.00</td>
</tr>
<tr>
<td>7</td>
<td>188.75</td>
</tr>
<tr>
<td>8</td>
<td>40.00</td>
</tr>
<tr>
<td>9</td>
<td>703.75</td>
</tr>
</tbody>
</table>
Arrays With Non-Primitive Types

- Then we create an array of objects
  - In this case, objects will be Strings
  - Array won't hold actual object
    - holds references: pointers to objects

```java
String[] location = new String[10];
```
Arrays of Objects

- Now we can put references to Strings in our String array.

```java
location[0] = "Chan Centre";
```
## Arrays of Objects

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 185</td>
<td>0 201.25</td>
<td>0</td>
</tr>
<tr>
<td>1 92</td>
<td>1 100.50</td>
<td>1</td>
</tr>
<tr>
<td>2 370</td>
<td>2 412.75</td>
<td>2</td>
</tr>
<tr>
<td>3 485</td>
<td>3 555.25</td>
<td>3</td>
</tr>
<tr>
<td>4 209</td>
<td>4 195.00</td>
<td>4</td>
</tr>
<tr>
<td>5 128</td>
<td>5 160.00</td>
<td>5</td>
</tr>
<tr>
<td>6 84</td>
<td>6 105.00</td>
<td>6</td>
</tr>
<tr>
<td>7 151</td>
<td>7 188.75</td>
<td>7</td>
</tr>
<tr>
<td>8 32</td>
<td>8 40.00</td>
<td>8</td>
</tr>
<tr>
<td>9 563</td>
<td>9 703.75</td>
<td>9</td>
</tr>
</tbody>
</table>

- Now we can put references to Strings in our String array.

    location[0] = "Chan Centre";
    location[1] = "Law School";
## Arrays of Objects

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 185</td>
<td>0 201.25</td>
<td>0</td>
</tr>
<tr>
<td>1 92</td>
<td>1 100.50</td>
<td>1</td>
</tr>
<tr>
<td>2 370</td>
<td>2 412.75</td>
<td>2</td>
</tr>
<tr>
<td>3 485</td>
<td>3 555.25</td>
<td>3</td>
</tr>
<tr>
<td>4 209</td>
<td>4 195.00</td>
<td>4</td>
</tr>
<tr>
<td>5 128</td>
<td>5 160.00</td>
<td>5</td>
</tr>
<tr>
<td>6 84</td>
<td>6 105.00</td>
<td>6</td>
</tr>
<tr>
<td>7 151</td>
<td>7 188.75</td>
<td>7</td>
</tr>
<tr>
<td>8 32</td>
<td>8 40.00</td>
<td>8</td>
</tr>
<tr>
<td>9 563</td>
<td>9 703.75</td>
<td>9</td>
</tr>
</tbody>
</table>

- Now we can put references to Strings in our String array.

```java
location[0] = "Chan Centre";
location[1] = "Law School";
location[2] = "Main Library";
```
# Arrays of Objects

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>185</td>
</tr>
<tr>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>370</td>
</tr>
<tr>
<td>3</td>
<td>485</td>
</tr>
<tr>
<td>4</td>
<td>209</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>151</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>563</td>
</tr>
<tr>
<td>0</td>
<td>201.25</td>
</tr>
<tr>
<td>1</td>
<td>100.50</td>
</tr>
<tr>
<td>2</td>
<td>412.75</td>
</tr>
<tr>
<td>3</td>
<td>555.25</td>
</tr>
<tr>
<td>4</td>
<td>195.00</td>
</tr>
<tr>
<td>5</td>
<td>160.00</td>
</tr>
<tr>
<td>6</td>
<td>105.00</td>
</tr>
<tr>
<td>7</td>
<td>188.75</td>
</tr>
<tr>
<td>8</td>
<td>40.00</td>
</tr>
<tr>
<td>9</td>
<td>703.75</td>
</tr>
</tbody>
</table>

Now we can put references to Strings in our String array.

```java
location[0] = "Chan Centre";
location[1] = "Law School";
location[2] = "Main Library";
...and so on...
```
Arrays of Objects

- Or we could have done this:

```java
String[] location = {
    "Chan Centre", 
    "Law School", 
    "Main Library", 
    .... 
};
```
### Arrays of Objects

#### Example Table

<table>
<thead>
<tr>
<th>cansSold</th>
<th>cashIn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>185</td>
</tr>
<tr>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>370</td>
</tr>
<tr>
<td>3</td>
<td>485</td>
</tr>
<tr>
<td>4</td>
<td>209</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
</tr>
<tr>
<td>7</td>
<td>151</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>563</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>201.25</td>
<td>100.50</td>
<td>412.75</td>
<td>555.25</td>
<td>195.00</td>
<td>160.00</td>
<td>105.00</td>
<td>188.75</td>
<td>40.00</td>
<td>703.75</td>
</tr>
</tbody>
</table>

#### Each individual String object in array of course has all String methods available

#### For example, what would this return?

`location[2].length()`
Each individual String object in array of course has all String methods available.

For example, what would this return?

`location[2].length()`

12
Arrays of Objects

Think about a cleaner way to do all this…