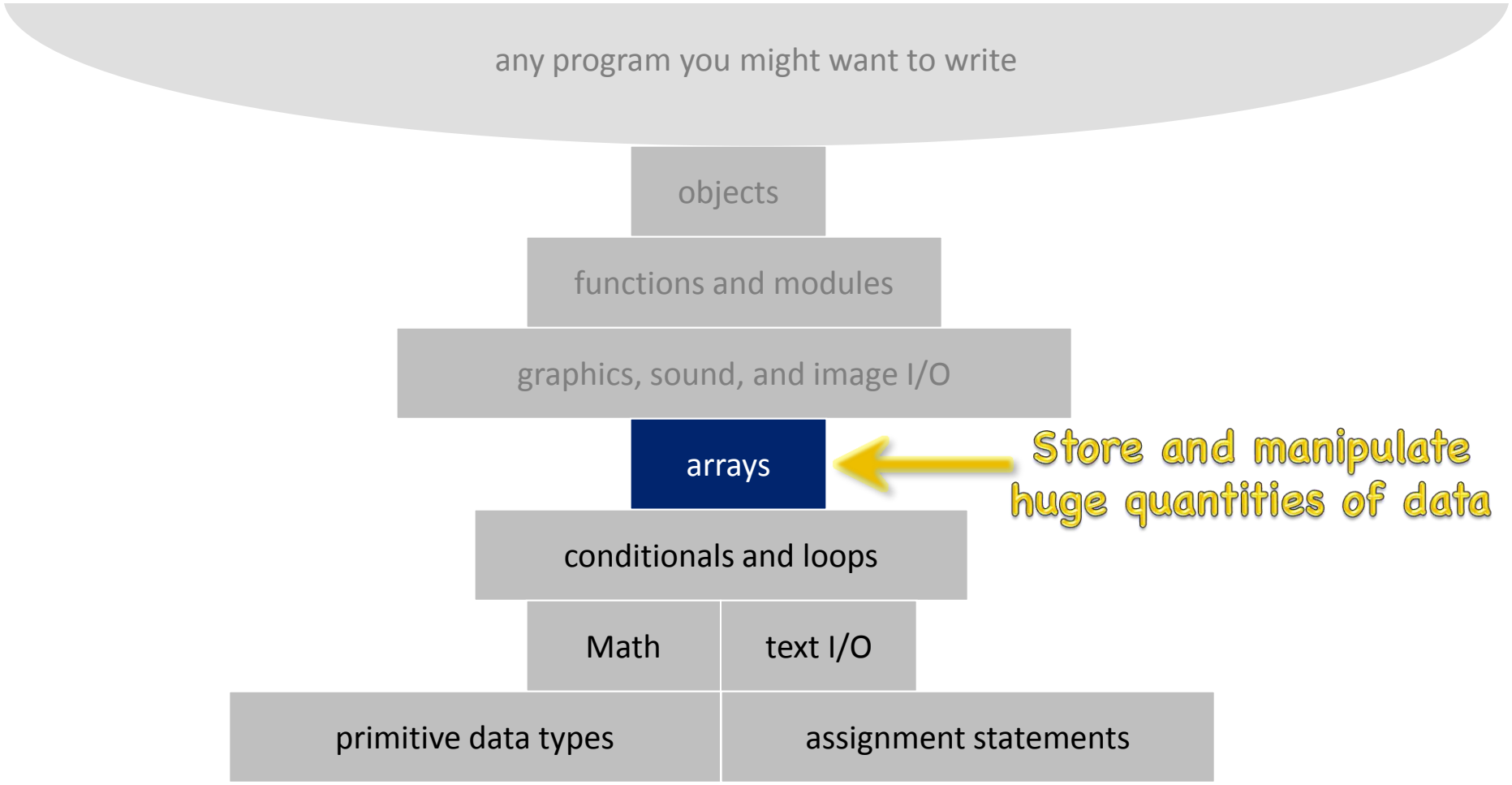
Arrays



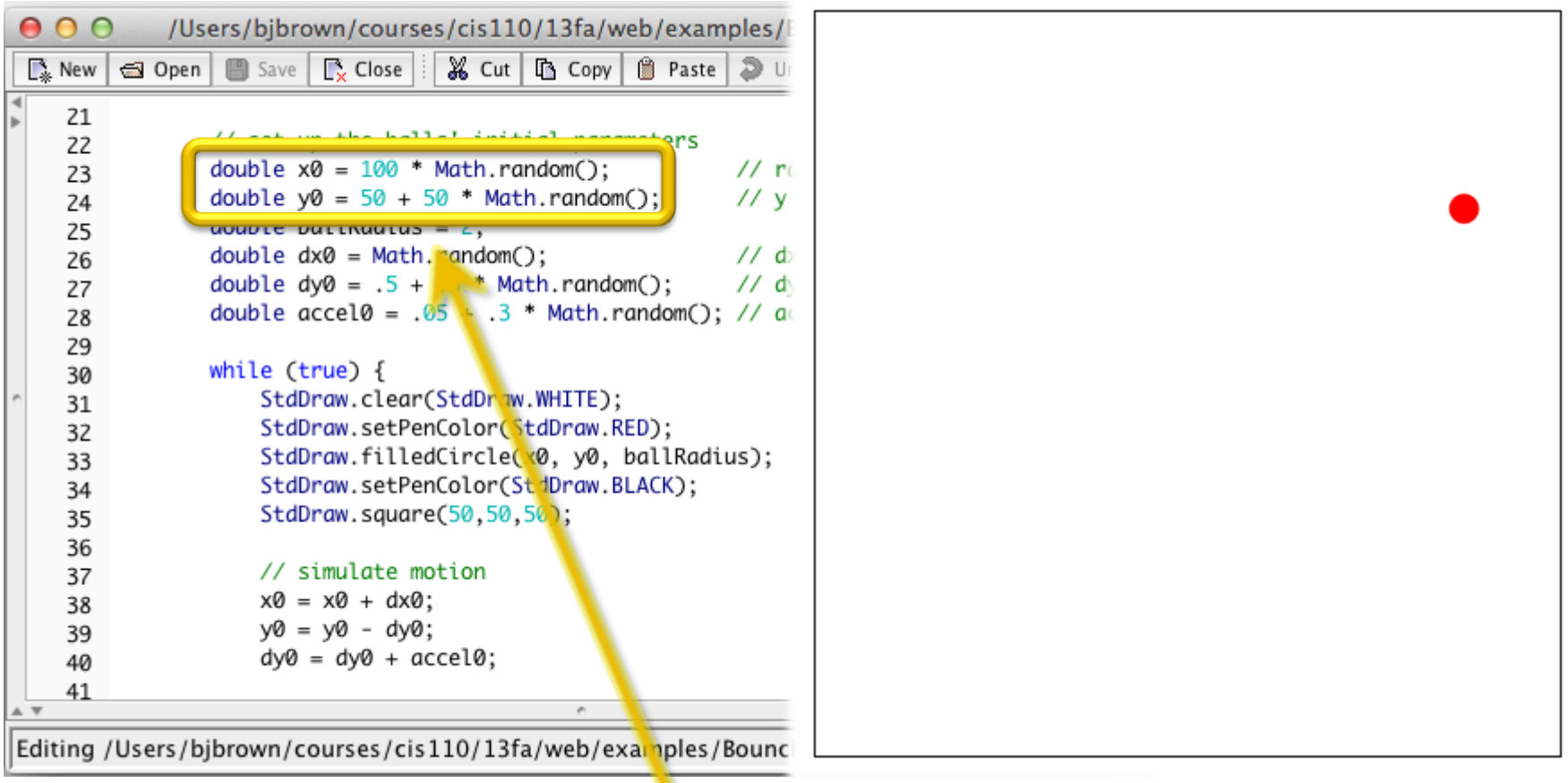
CIS 110 We Have .CLASS

Penn Engineering

Programming in Java

Section 1.4

Why Arrays?



CIS 110 We Have .CLASS

Penn Engineering

Programming in Java

Section 1.4

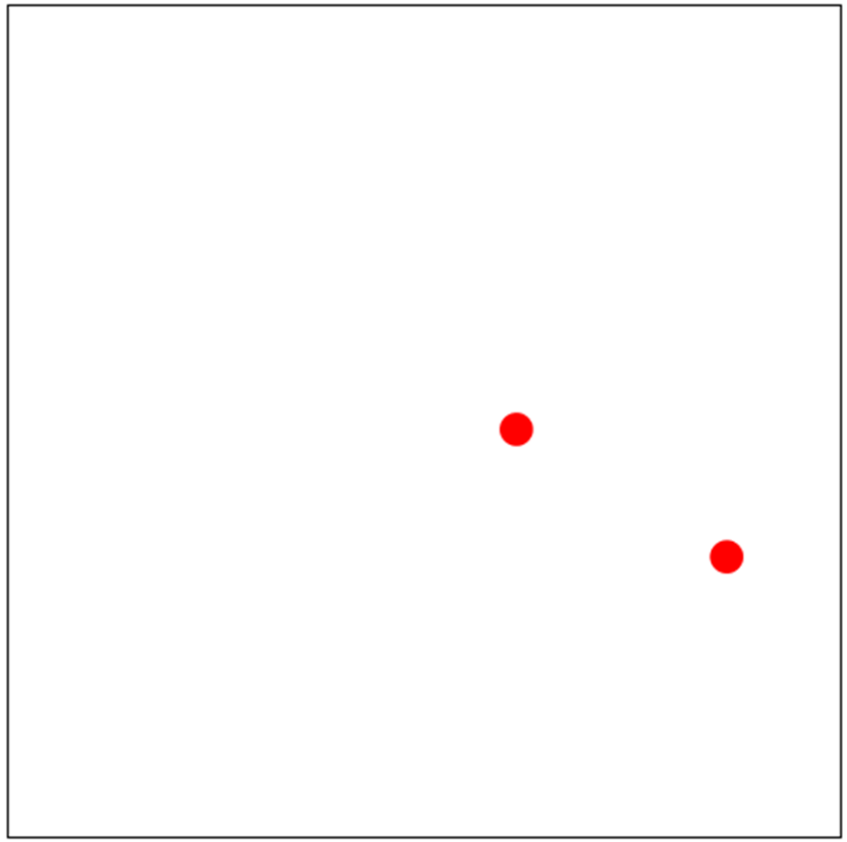
Why Arrays?

/Users/bjbrown/courses/cis110/13fa/web/examples/

New Open Save Close Cut Copy Paste

```
21
22          // set up the balls' initial parameters
23          double x0 = 100 * Math.random();          // r
24          double x1 = 100 * Math.random();          // r
25          double y0 = 50 + 50 * Math.random();      // y
26          double y1 = 50 + 50 * Math.random();      // y
27          double ballRadius = 2;
28          double dx0 = Math.random();               // d
29          double dx1 = Math.random();               // d
30          double dy0 = .5 + .5 * Math.random();     // d
31          double dy1 = .5 + .5 * Math.random();     // d
32          double accel0 = .05 + .3 * Math.random(); // a
33          double accel1 = .05 + .3 * Math.random(); // a
34
35          while (true) {
36              StdDraw.clear(StdDraw.WHITE);
37              StdDraw.setPenColor(StdDraw.RED);
38              StdDraw.filledCircle(x0, y0, ballRadius);
39              StdDraw.filledCircle(x1, y1, ballRadius);
40              StdDraw.setPenColor(StdDraw.BLACK);
41              StdDraw.square(50,50,50);
```

Editing /Users/bjbrown/courses/cis110/13fa/web/examples/Bounc



Two bouncing balls

CIS 110 We Have .CLASS

Penn Engineering

Section 1.4

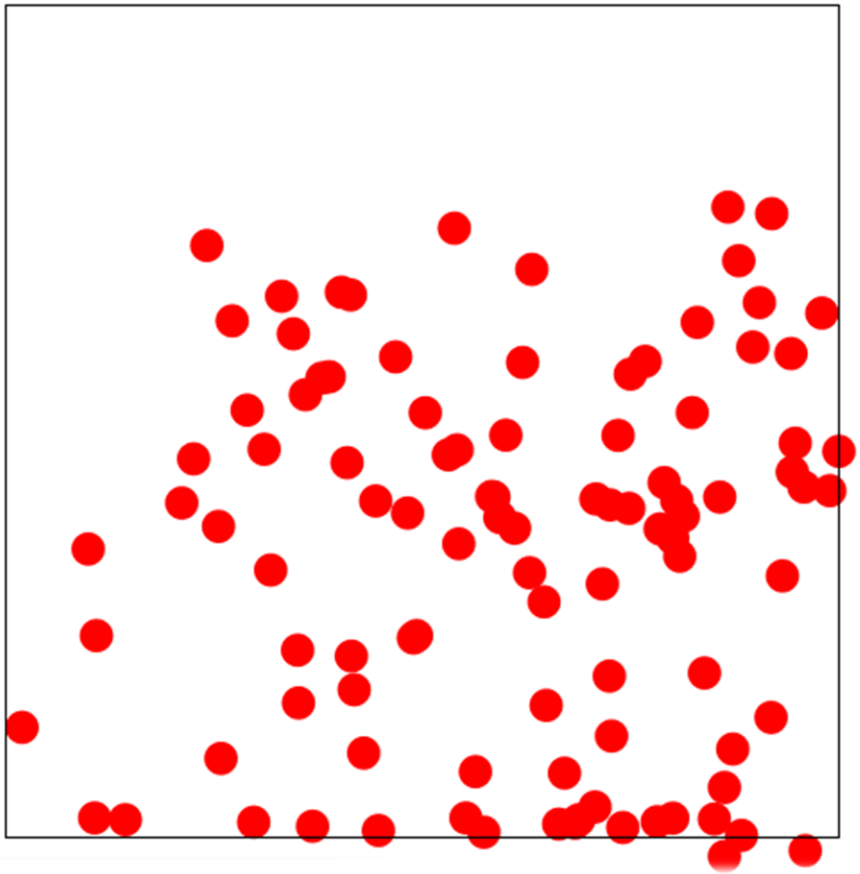
Why Arrays?

/Users/bjbrown/courses/cis110/13fa/web/examples/B

New Open Save Close Cut Copy Paste

```
21
22          // set up the balls' initial parameters
23          double x0 = 100 * Math.random();          // r
24          double x1 = 100 * Math.random();          // r
25          double x2 = 100 * Math.random();          // r
26          double x3 = 100 * Math.random();          // r
27          double x4 = 100 * Math.random();          // r
28          double x5 = 100 * Math.random();          // r
29          double x6 = 100 * Math.random();          // r
30          double x7 = 100 * Math.random();          // r
31          double x8 = 100 * Math.random();          // r
32          double x9 = 100 * Math.random();          // r
33          double x10 = 100 * Math.random();          //
34          double x11 = 100 * Math.random();          //
35          double x12 = 100 * Math.random();          //
36          double x13 = 100 * Math.random();          //
37          double x14 = 100 * Math.random();          //
38          double x15 = 100 * Math.random();          //
39          double x16 = 100 * Math.random();          //
40          double x17 = 100 * Math.random();          //
```

Editing /Users/bjbrown/courses/cis110/13fa/web/examples/Bounc



100 bouncing balls

CIS 110 We have .CLASS

Penn Engineering

Programming in Java

Section 1.4

Declaring Arrays

Array: Indexed sequence of values of the same type

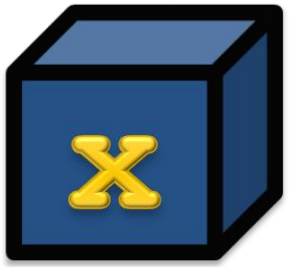
```
// easy alternative
double[] x = new double[100];
```

Declaring Arrays

Array: Indexed sequence of values of the same type

```
// easy alternative
double[] x = new double[100];
```

x will contain an array of many doubles

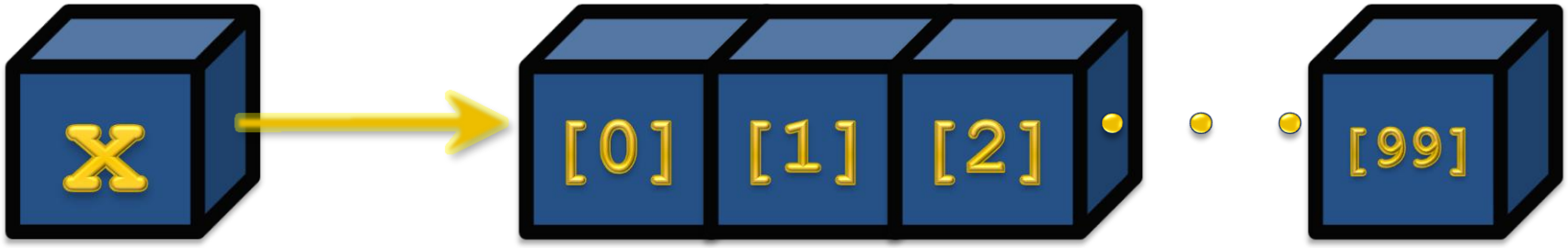


Declaring Arrays

Array: Indexed sequence of values of the same type

```
// easy alternative
double[] x = new double[100];
```

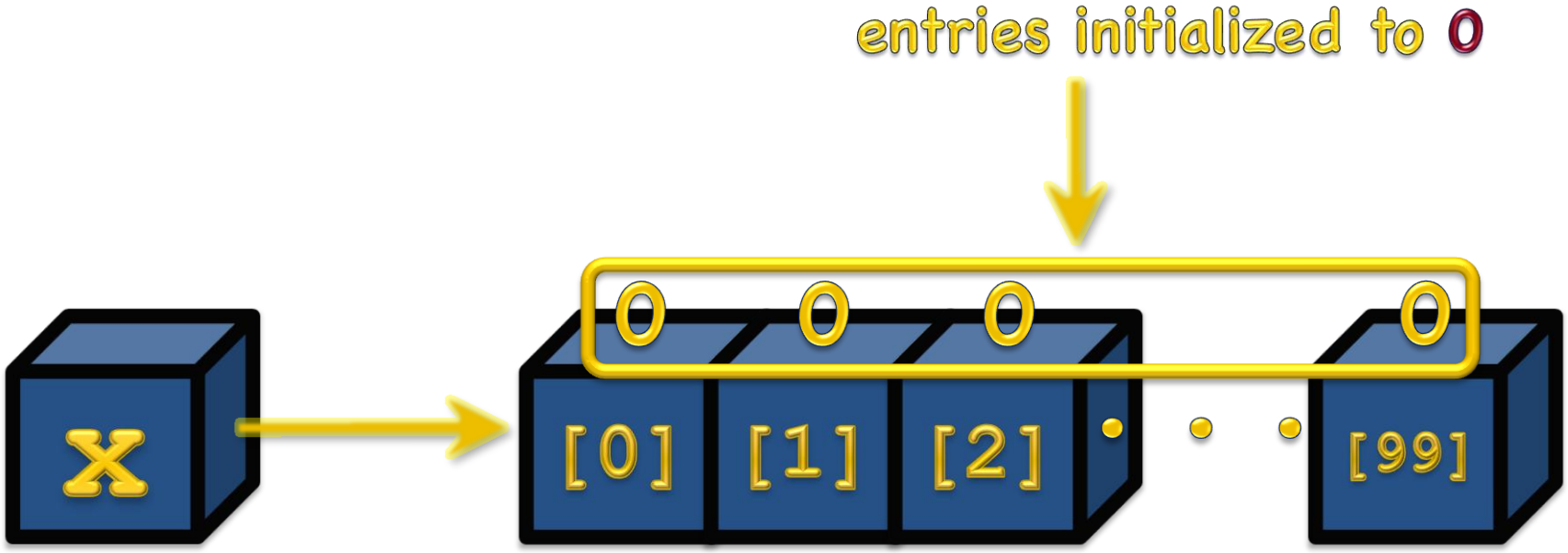
- find space for 100 doubles
- store location in x



Declaring Arrays

Array: Indexed sequence of values of the same type

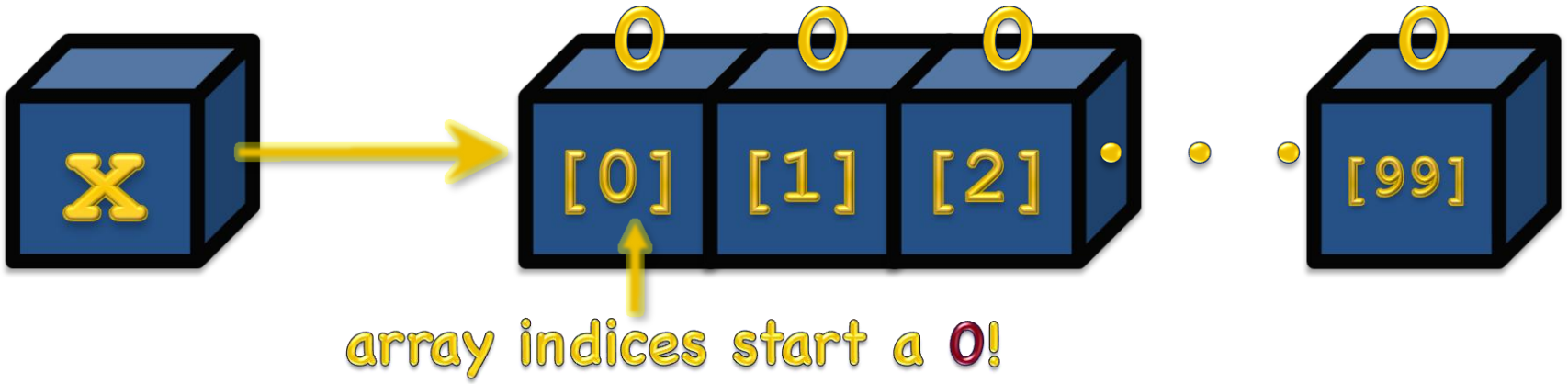
```
// easy alternative
double[] x = new double[100];
```



Declaring Arrays

Array: Indexed sequence of values of the same type

```
// easy alternative
double[] x = new double[100];
```



CIS 11 We Have .CLASS

Penn Engineering

Programming in Java

Section 1.4

Using Arrays

/Users/bjbrown/courses/cis110/13fa/web/examples/PrintArguments.java

New | Open | Save | Close | Cut | Copy | Paste | Undo | Redo | Find | Compile

```
22 public class PrintArguments {
23     public static void main(String[] args) {
24         // make an array of the same length as args to copy to
25         String[] argsCopy = new String[args.length];
26
27         // print each element of the new array, copy in the
28         // corresponding element from args, then print it again
29         for (int i = 0; i < args.length; i++) {
30             System.out.println("Before " + i + ": " + argsCopy[i]);
31             argsCopy[i] = args[i];
32             System.out.println("After  " + i + ": " + argsCopy[i]);
33         }
34     }
35 }
```

Editing /Users/bjbrown/... Bracket matches: public class PrintArguments { 22:0

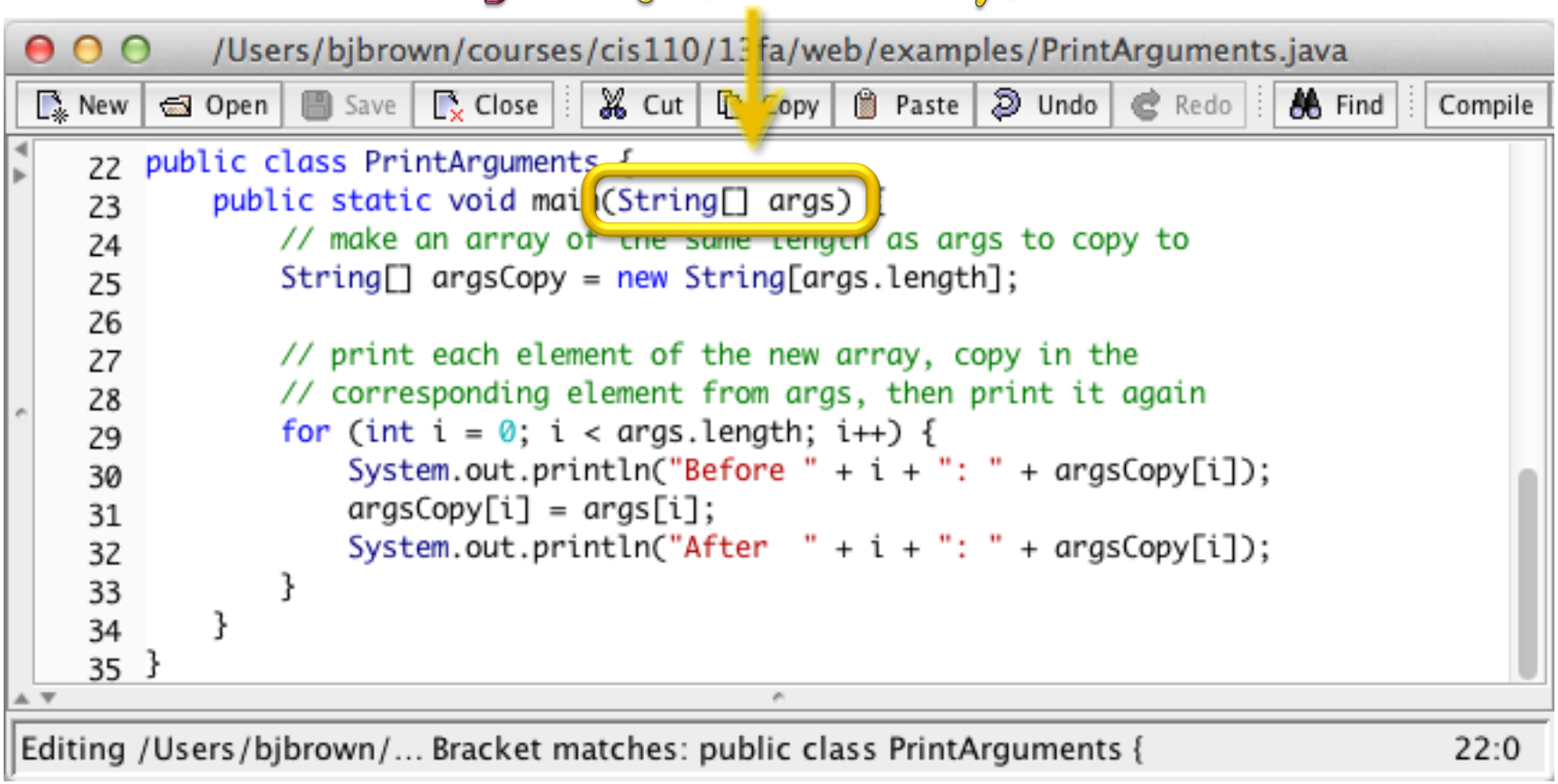
CIS 110 We have .CLASS

Penn Engineering

Section 1.4

Using Arrays

args is just an array!



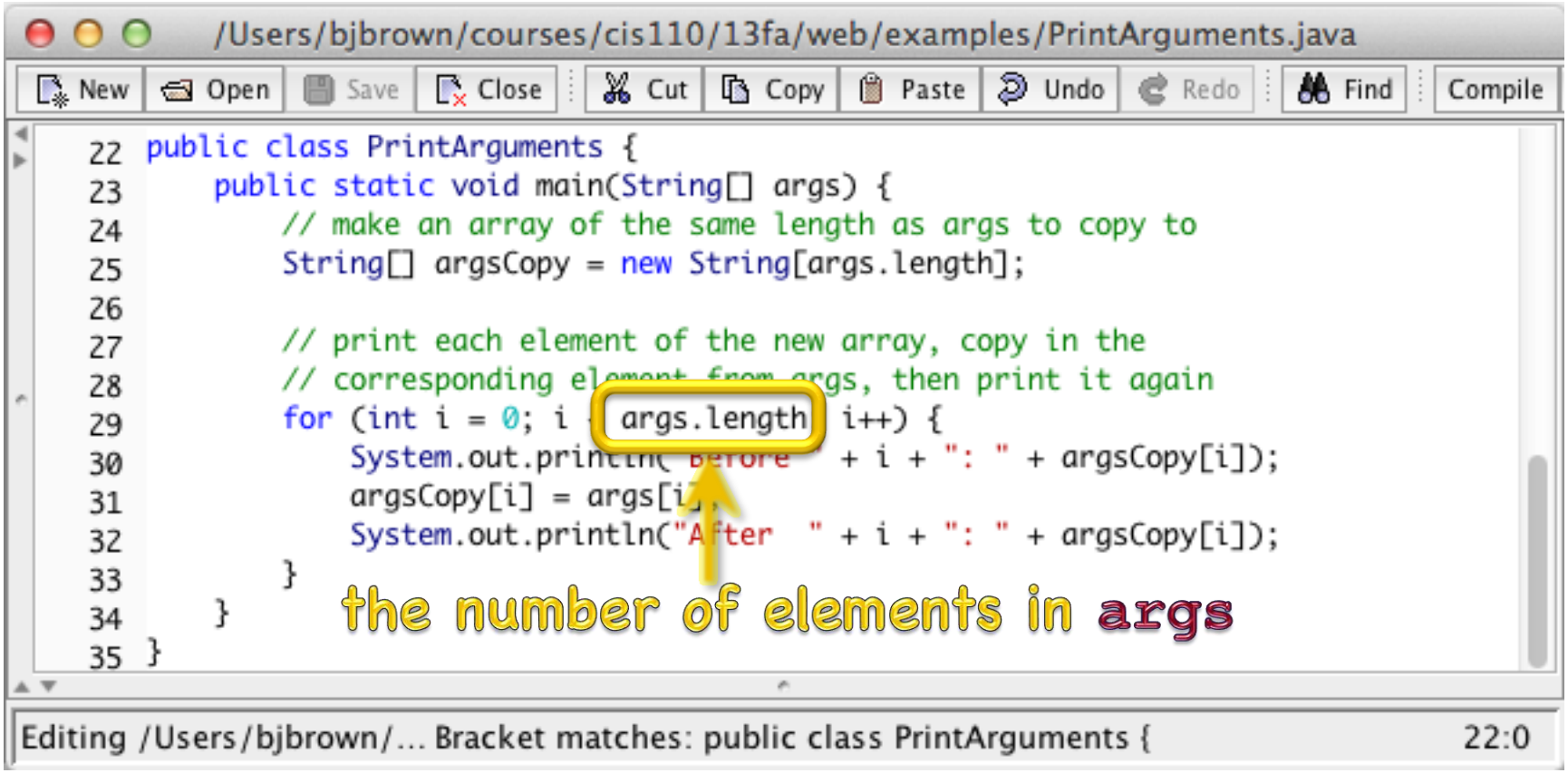
CIS 110 We have .CLASS

Penn Engineering

Programming in Java

Section 1.4

Using Arrays



CIS 110 We have .CLASS

Penn Engineering

Programming in Java

Section 1.4

Using Arrays

/Users/bjbrown/courses/cis110/13fa/web/examples/PrintArguments.java

New | Open | Save | Close | Cut | Copy | Paste | Undo | Redo | Find | Compile

```
22 public class PrintArguments {
23     public static void main(String[] args) {
24         // make an array of the same length as args to copy to
25         String[] argsCopy = new String[args.length];
26
27         // print each element of the new array, copy in the
28         // corresponding element from args, then print it again
29         for (int i = 0; i < args.length; i++) {
30             System.out.println("Before " + i + ": " + argsCopy[i]);
31             argsCopy[i] = args[i];
32             System.out.println("After  " + i + ": " + argsCopy[i]);
33         }
34     }
35 }
```

Editing /Users/bjbrown/... Bracket matches: public class PrintArguments { 22:0

Strings default to special value `null` (no value)

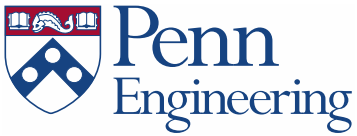
CIS 110 We have .CLASS

Penn Engineering

Section 1.4

Interactions Pane Exercises

```
> int[] arr = new int[4]
> arr.length


> arr[arr.length]



> for (int i = 0; i < arr.length; i++)
      System.out.println(i);



> System.out.println(arr)
```



Section 1.4

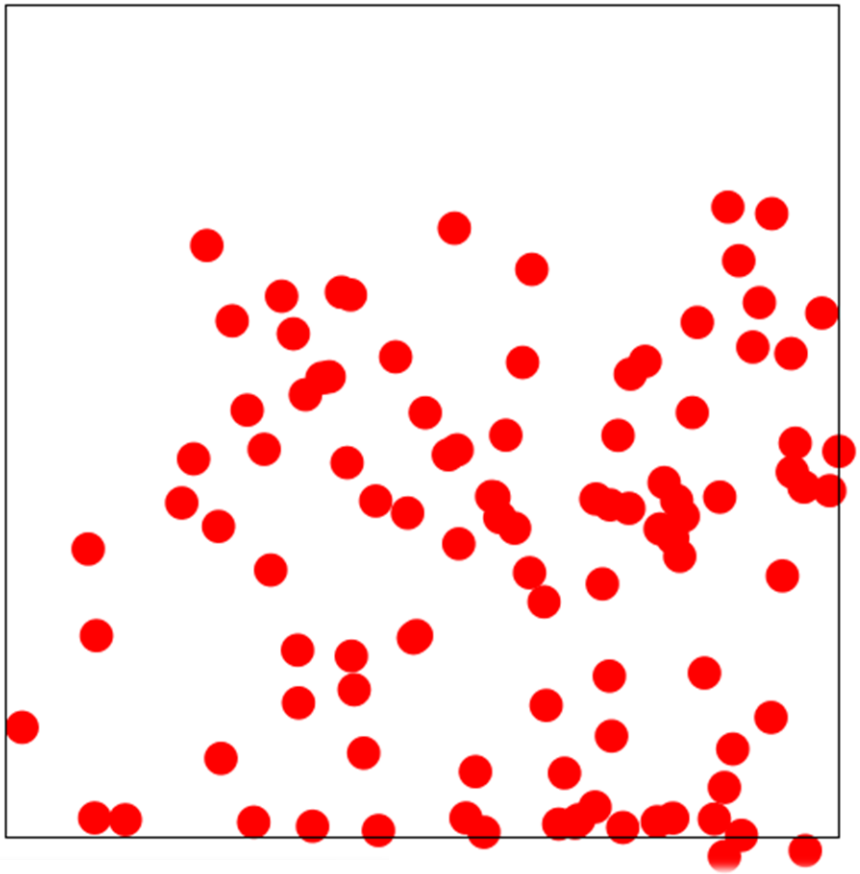
N Bouncing Balls

/Users/bjbrown/courses/cis110/13fa/web/examples/

New | Open | Save | Close | Cut | Copy | Paste

```
24
25          // set up the parallel arrays to store ball inf
26          double[] x  = new double[nBalls];
27          double[] y  = new double[nBalls];
28          double[] dx = new double[nBalls];
29          double[] dy = new double[nBalls];
30          double[] accel = new double[nBalls];
31
32          double ballRadius = 2; // all balls are the sa
33
34          // set up the balls' initial parameters
35          for (int i = 0; i < nBalls; i++) {
36              x[i] = 100 * Math.random();          // ra
37              y[i] = 50 + 50 * Math.random();      // y
38              dx[i] = Math.random()                // dx
39              dy[i] = .5 + .5 * Math.random();     // dy
40              accel[i] = .05 + .3 * Math.random(); // ac
41          }
42
43          while (true) {
44              // draw the balls
```

Editing /Users/bjbr...Bracket matches: for (int i = 0; i < nBalls;



declare arrays to track balls

CIS 110 We Have .CLASS

Penn Engineering

Section 1.4

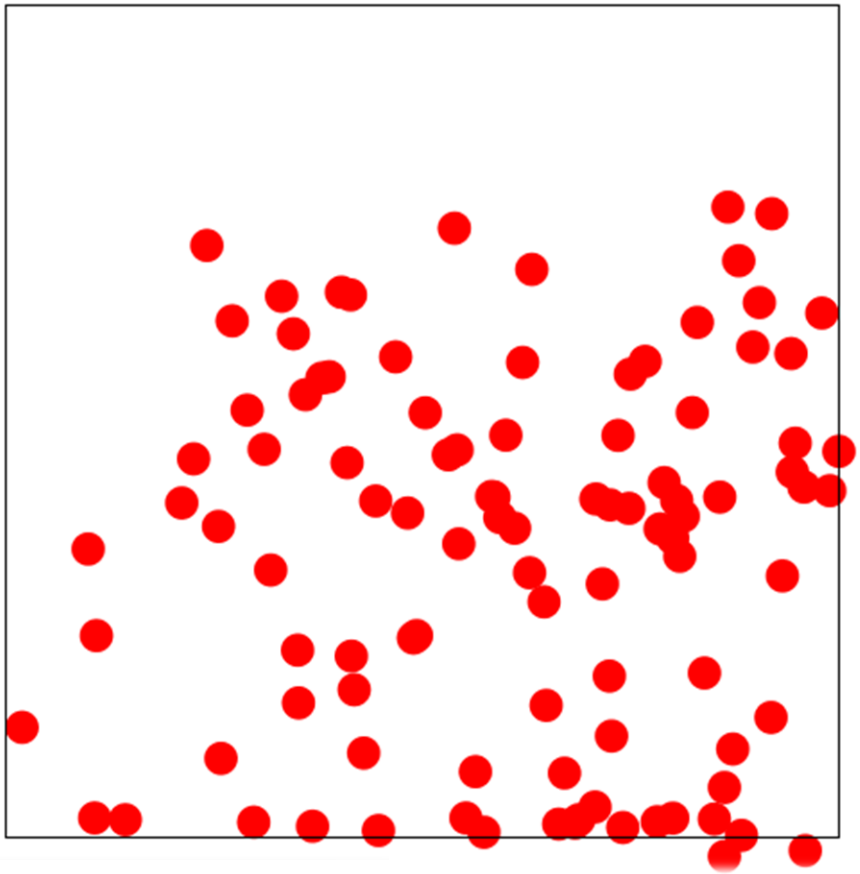
N Bouncing Balls

/Users/bjbrown/courses/cis110/13fa/web/examples/

New | Open | Save | Close | Cut | Copy | Paste

```
24
25          // set up the parallel arrays to store ball inf
26          double[] x  = new double[nBalls];
27          double[] y  = new double[nBalls];
28          double[] dx = new double[nBalls];
29          double[] dy = new double[nBalls];
30          double[] accel = new double[nBalls];
31
32          double ballRadius = 2; // all balls are the sa
33
34
35          for (int i = 0; i < nBalls; i++) {
36              x[i] = 100 * Math.random();
37              y[i] = 50 + 50 * Math.random();
38              dx[i] = Math.random();
39              dy[i] = .5 + .5 * Math.random();
40              accel[i] = .05 + .3 * Math.random();
41          }
42
43          while (true) {
44              // draw the balls
```

Editing /Users/bjbr...Bracket matches: or (int i = 0; i < nBalls;



initialize values with a for loop

CIS 110 We have .CLASS

Penn Engineering

Programming in Java

Section 1.4

Array-Processing Examples

create an array with random values	`double[] a = new double[N];`
	`for (int i = 0; i < N; i++)`
	`  a[i] = Math.random();`

print the array values, one per line	`for (int i = 0; i < N; i++)` `  System.out.println(a[i]);`

find the maximum of the array values	`double max = Double.NEGATIVE_INFINITY;`
	`for (int i = 0; i < N; i++)`
	`  if (a[i] > max) max = a[i];`

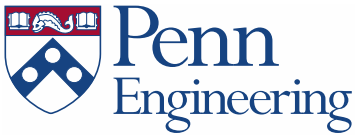
compute the average of the array values	`double sum = 0.0;`
	`for (int i = 0; i < N; i++)`
	`  sum += a[i];`
	`double average = sum / N;`

copy to another array	`double[] b = new double[N];`
	`for (int i = 0; i < N; i++)`
	`  b[i] = a[i];`

reverse the elements within an array	`for (int i = 0; i < N/2; i++)`
	`{`
	`  double temp = b[i];`
	`  b[i] = b[N-1-i];`
	`  b[N-i-1] = temp;`
	`}`

Explicit Value Initialization

```
String[] rank = {
   "2", "3", "4", "5", "6", "7", "8", "9",
   "10", "Jack", "Queen", "King", "Ace"
};

String[] suit = {
   "Clubs", "Diamonds", "Hearts", "Spades"
};

int i = (int) (Math.random() * 13);   // between 0 and 12
int j = (int) (Math.random() * 4);    // between 0 and 3

System.out.println(rank[i] + " of " + suit[j]);
```



Section 1.4

Explicit Value Initialization

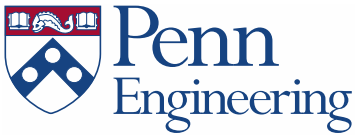
- list contents of array instead of using **new**
- array size determined by values

```
String[] rank = {
   "2", "3", "4", "5", "6", "7", "8", "9",
   "10", "Jack", "Queen", "King", "Ace"
};

String[] suit = {
   "Clubs", "Diamonds", "Hearts", "Spades"
};

int i = (int) (Math.random() * 13);   // between 0 and 12
int j = (int) (Math.random() * 4);    // between 0 and 3

System.out.println(rank[i] + " of " + suit[j]);
```

Explicit Value Initialization

```
String[] rank = {
   "2", "3", "4", "5", "6", "7", "8", "9",
   "10", "Jack", "Queen", "King", "Ace"
};

String[] suit = {
   "Clubs", "Diamonds", "Hearts", "Spades"
};

int i = (int) (Math.random() * 13);   // between 0 and 12
int j = (int) (Math.random() * 4);    // between 0 and 3

System.out.println(rank[i] + " of " + suit[j]);
```

what does this print out?



Section 1.4