Arrays of Objects

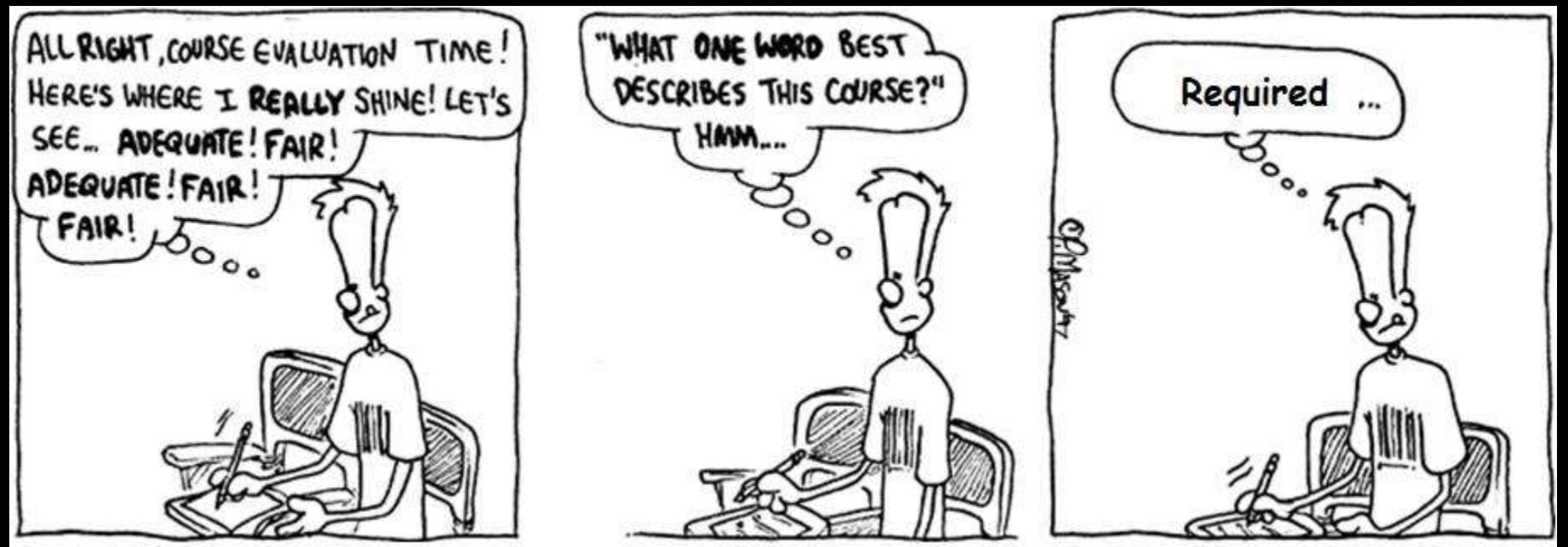
GEEN163

“Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.”

-Linus Torvalds

Course Evaluations

- Course evaluations are available on Blackboard
- Be sure to complete all evaluations for all classes



Programming Project

- This week's program should be done by teams of two students
- There are three classes, each with several small methods
- The program involves a GUI with graphics

Elements of an Array

- You can use the elements of an array any where you might use a regular variable

```
mouse[22] = 63;
mouse[first] = mouse[first] + 21;
rat = mouse[ mouse[1] ];
System.out.println("Big = "+mouse[7]);
```

- When you use an element of an array, it must have an [index]

Loops

- When you are dealing with the elements of an array, you almost always use a loop

```
double[] bee = new double[4];
double wasp = 0.0
for (int bug = 0; bug < 4; bug++) {
        wasp += bee[bug];
}
```

Whole Arrays

- There is a limited number of things you can do with a whole array
- You can pass an array without an index to a method

```
double copper;
double[] gold = new double[8];
copper = someMethod( gold );
```

Arrays as Objects

- An array is an object in Java
- You can call a method on the array
- Arrays have an integer class variable length that contains the length of the array

```
float[] wood = new float[5];
int stone;
stone = wood.length;              // stone = 5
```

Half Empty or Half Full

- An array might not contain as many useful values as its full capacity

```
double[] moose = new double[100];
for (int calf = 0; calf < 50; calf++) {
        moose[calf] = calf * 47.0;
}
```

- The array moose has 100 elements, but only 50 values have been used

What is displayed?

```
double[] dog = { 3.0, 4.0, 5.0 };
for (int pup = 0; pup < 3; pup++) {
    dog[pup] = dog[pup] / 2.0;
}
for (int i = 0; i < 3; i++) {
    System.out.print( dog[i]+" " );
}
```

A. 0.0 1.0 2.0

B. 1.5 2.0 2.5

C. 3.0 4.0 5.0

D. none of the above

What is displayed?

```
double[] dog = { 3.0, 4.0, 5.0 };
for (double hound : dog) {
    hound = hound / 2.0;
}
for (int i = 0; i < 3; i++) {
    System.out.print( dog[i]+" " );
}
```

A. 0.0 1.0 2.0

B. 1.5 2.0 2.5

C. 3.0 4.0 5.0

D. none of the above

Arrays of Objects

- The elements of an array can be primitive type like int, double, long, float, char or boolean
- An array can also have objects as elements

```
Widget[] things = new Widget[47];
```

- This creates an array, things, that can contain 47 objects of the class Widget

Empty Array

- When you create an array of double or int, the array initially contains many doubles or ints
- When you create an array of objects, the array is initially empty.

Creating an Array of Objects

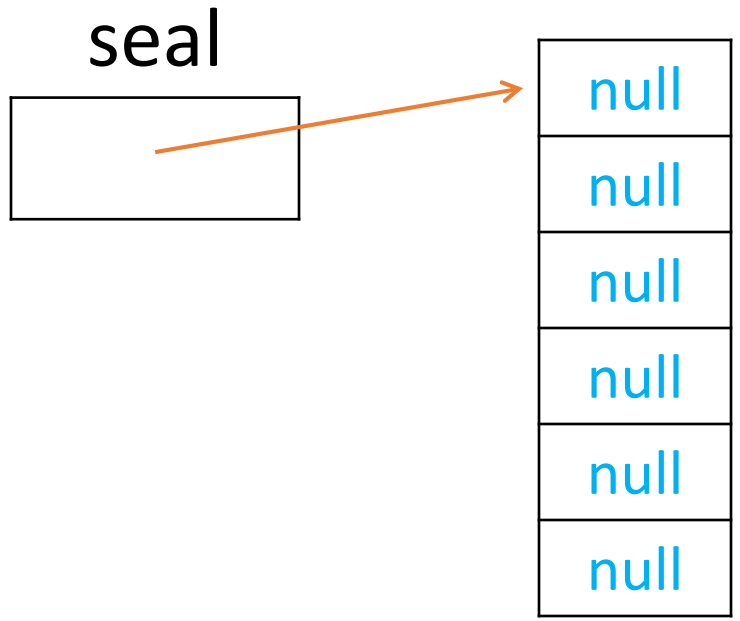
- When you declare an array of objects. The variable does not yet hold the array

Widget[] seal;

Creating an Array of Objects

- An array is created the same way you create other objects. The **new** create the array

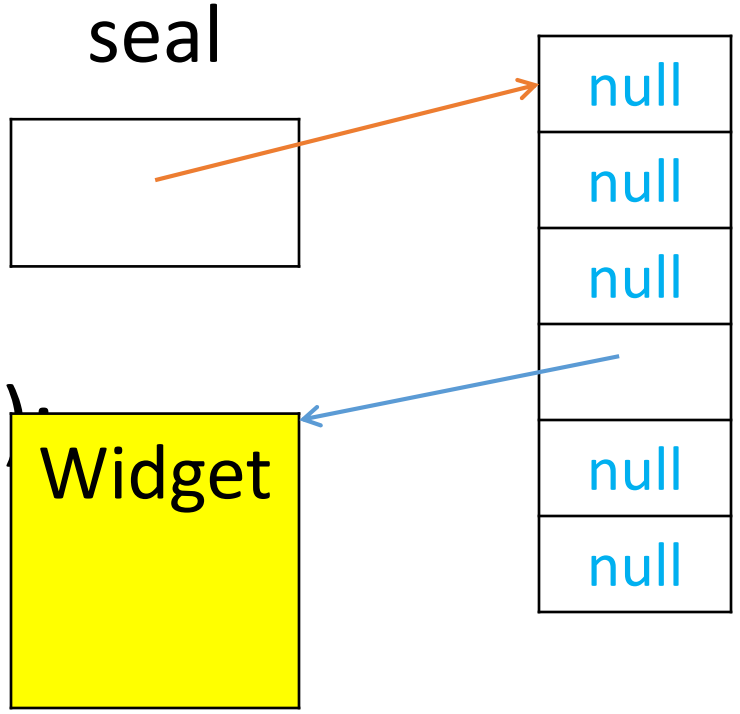
```
Widget[] seal;
seal = new Widget[6];
```



Creating an Array of Objects

- You have to create each of the element objects in the array The **new** creates each element

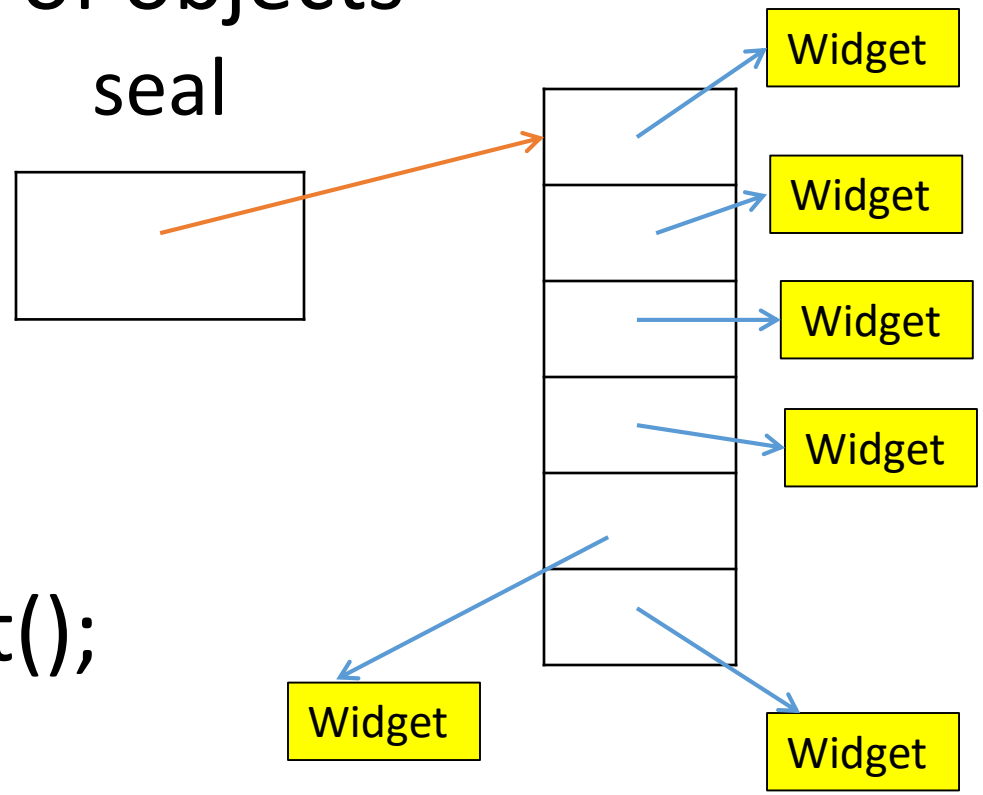
```
Widget[] seal;
seal = new Widget[6];
seal[3] = new Widget();
```



Creating an Array of Objects

- You should create a new object for each element of the array of objects

```
Widget[] seal;
seal = new Widget[6];
for(int i=0; i<6; i++)
   seal[i] = new Widget();
```



NullPointerException

- If you do not create the objects of an array, you may get a NullPointerException error

```
String[] stuff = new String[4];
stuff[2].length;                    // error
```

Try it

- Create an array of 6 Roster objects and fill each element with an object

Possible Solution

```
Roster[] book = new Roster[6];
for (int worm = 0; worm < 6; worm++) {
        book[worm] = new Roster();
}
```

Creating Objects from Files

- A program might declare an array of objects in the beginning and then create the object later
- Consider a program that reads a data file and creates an object for each line of data

Example Class

- Consider a class that holds data about classrooms

```
public class Classroom {
        private String   building;
        private int      roomNum;
        private int      capacity;
}
```

Write with your Team

- Write a constructor for Classroom that initializes the class instance variables

Possible Solution

```
public Classroom( String name, int roomNum, int cap) {
        building = name;
        this.roomNum = roomNum;
        capacity = cap;
}
```

Getters

- Write three short methods that return the value of the three class instance variables

Possible Solution

```
public String getBuilding() {
        return building;
}
public int getRoomNum() {
        return roomNum;
}
public int getCapacity() {
        return capacity;
}
```

Building an Array of Objects

```
java.io.File frog = new java.io.File("mydata.txt");
java.util.Scanner inFile = new java.util.Scanner(frog);
Classroom[] space = new Classroom[500];
int numRooms = 0;
while (inFile.hasNext()) {
        String bldg = inFile.next();
        int room    = inFile.nextInt();
        int cap     = inFile.nextInt();
        space[numRooms] = new Classroom( bldg, room, cap);
        numRooms++;
}
```

Search Example

- Assume we have an array of Classroom objects called space with numRooms values in the array
- We want to display all rooms with capacity greater than 150

Which **if** statement will select object's whose capacity is greater than 150?

- Assume there is a for loop with i as the index

A. if (capacity[i] > 150)
B. if (space.getCapacity(i) > 150)
C. if (space[i].getCapacity() > 150)
D. if (space[i].capacity > 150)
E. if (space[150] > capacity)

What is wrong with this program?

```
for (int i = 0; i < space.length; i++) {
   if (space[i].getCapacity() > 150) {
      System.out.println(space[i].getBuilding() +
          " " + space[i].getRoomNum());
   }
}
```

A. It should be <= space.length

B. space.length should be numRooms

C. You cannot call methods inside println

D. Nothing

Stringing Methods Together

- Sometimes a program will need to put many parts together

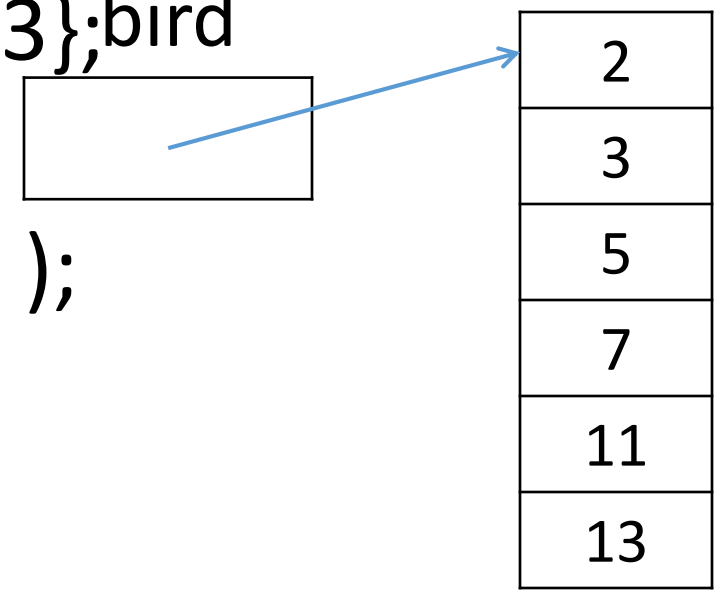
```
if (space[i].getBuilding().equals("Graham"))
```

- Java interprets this from left to right
  - Get the i^{th} element of space (a Classroom object)
  - Get the building name from the object (a String)
  - Compare the name to "Graham" (a boolean)

Passing Array Reference

- When you pass an array to a method, the reference is copied to the method argument

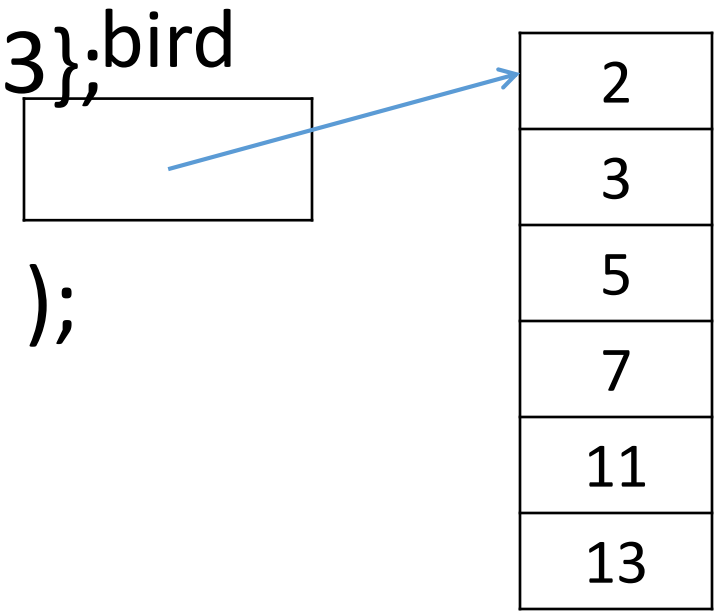
```
int[] bird = {2, 3, 5, 7, 11, 13};
aMethod( bird );
System.out.println( bird[2] );
        . . .
void aMethod(int[] fish) {
   fish[2] = 47;
}
```



Passing Array Reference

- When you pass an array to a method, the reference is copied to the method argument

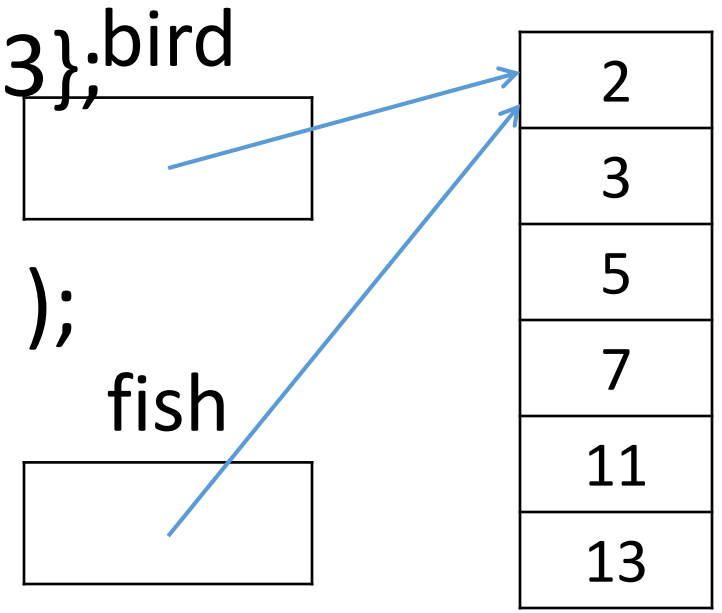
```
int[] bird = {2, 3, 5, 7, 11, 13};
aMethod( bird );
System.out.println( bird[2] );
      . . .
void aMethod(int[] fish) {
  fish[2] = 47;
}
```



Passing Array Reference

- When you pass an array to a method, the reference is copied to the method argument

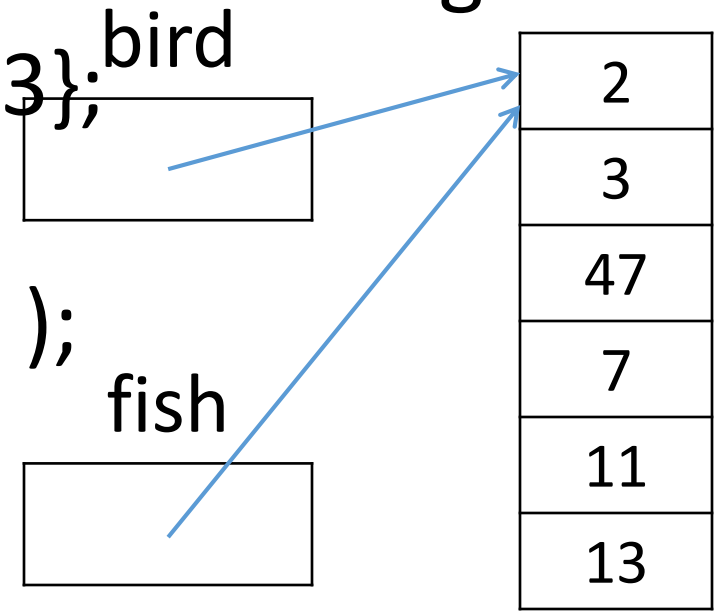
```
int[] bird = {2, 3, 5, 7, 11, 13};
aMethod( bird );
System.out.println( bird[2] );
      . . .
void aMethod(int[] fish) {
  fish[2] = 47;
}
```



Passing Array Reference

- When you pass an array to a method, the reference is copied to the method argument

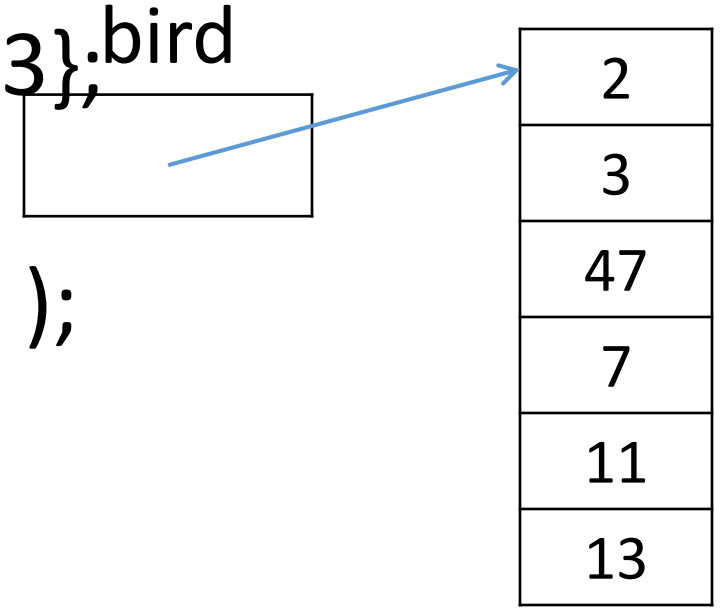
```
int[] bird = {2, 3, 5, 7, 11, 13};
aMethod( bird );
System.out.println( bird[2] );
      . . .
void aMethod(int[] fish) {
  fish[2] = 47;
}
```



Passing Array Reference

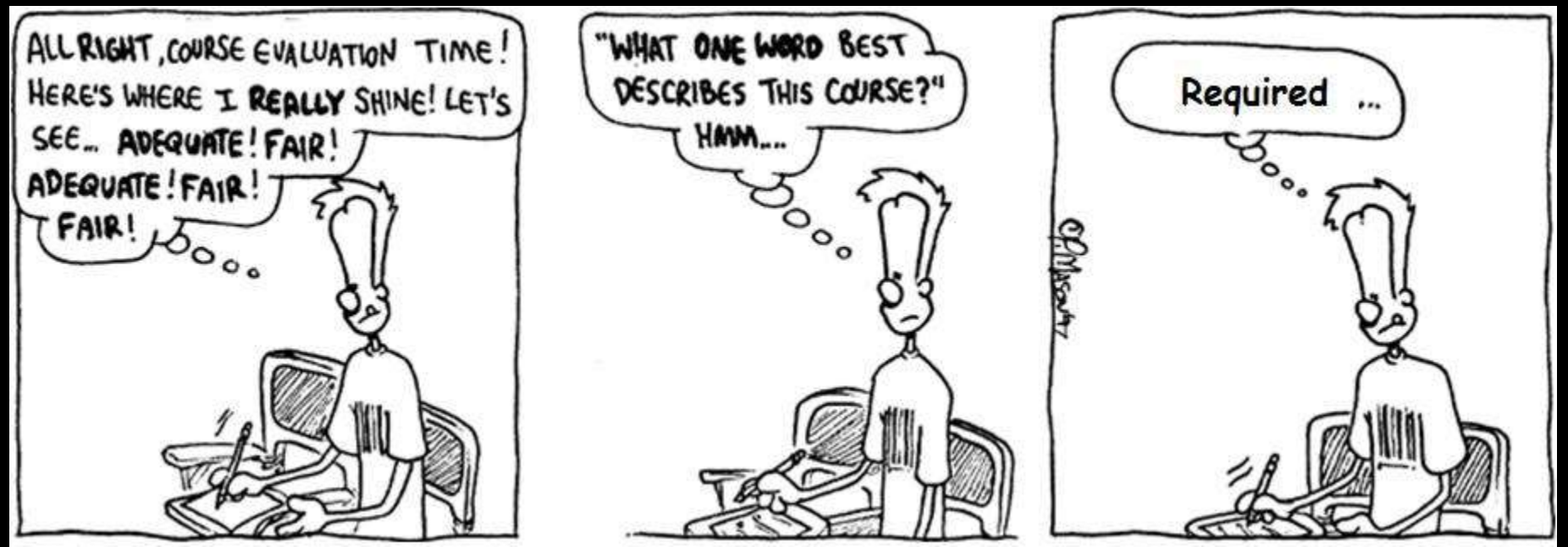
- When you pass an array to a method, the reference is copied to the method argument

```
int[] bird = {2, 3, 5, 7, 11, 13};
aMethod( bird );
System.out.println( bird[2] );
      . . .
void aMethod(int[] fish) {
  fish[2] = 47;
}
```



Course Evaluations

- Course evaluations are available on Blackboard
- Be sure to complete all evaluations for all classes



Programming Project

- This week's program should be done by teams of two students
- There are three classes, each with several small methods
- The program involves a GUI with graphics