Objects and Arrays

GEEN163

“It's not at all important to get it right the first time. It's vitally important to get it right the last time.”

-Andrew Hunt and David Thomas

Course Evaluations

- Course evaluations are available on Blackboard
- Be sure to complete all evaluations for all classes
- 22% of the student in GEEN163 have completed the course evaluation

Website Updated

- The GEEN163 website, http://williams.comp.ncat.edu/geen163/ has been updated

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

You CAN do it

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 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

What is Displayed?

```
String[] goat = new String[3];
String target = "";
for (int i = 0; i < 3; i++) {
    if (goat[0].equals(target)) {
        System.out.print( i );
    }
}
```

A. 0 1 2
B. 0
C. 2
D. nothing
E. none of the above

Now What is Displayed?

```
String[] goat = new String[3];
String target = "";
for (int i = 0; i < 3; i++) {
    if (target.equals(goat[0])) {
        System.out.print( i );
    }
}
```

A. 0 1 2
B. 0
C. 2
D. nothing
E. none of the above

Example Class

- Consider a class that holds data about classrooms

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

Constructor

- Here is a possible constructor for Classroom that initializes all class instance variables to parameters

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

Setters

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

Possible Solution

```
public void setBuilding( String name ) {
        building = name;
}
public void setRoomNum( int num ) {
        roomNum = num;
}
public void setCapacity( int cap ) {
        capacity = cap;
}
```

Write a Method

- Write a method that has a room name and number as parameters and returns true when if this object matches the name and number

Method Header, Write the Rest

- Write a method that has a room name and number as parameters and returns true when if this object matches the name and number

```
public boolean isRoom( String bldg, int num )
```

Possible Solution

```
public boolean isRoom(String bldg, int num ){
    if (bldg.equalsIgnoreCase(building)
                    && num == roomNum) {
        return true;
    }
    return false;
}
```

Another Possible Solution

```
public boolean isRoom(String bldg, int num ){
    return bldg.equalsIgnoreCase(building)
                    && num == roomNum;
}
```

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[1500];   // list of rooms
int numRooms = 0;                          // number of rooms
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 the capacity of a given room
- The input will be a building name and room number

Search Program

```
String name = keyboard.next();
int num = keyboard.nextInt();
look: for (int i = 0; i < numRooms; i++) {
   if (space[i].isRoom( name, num )) {
      System.out.println(space[i].getCapacity());
      break look;
   }
}
```

Search Program with extended for

```
String name = keyboard.next();
int num = keyboard.nextInt();
look: for (Classroom room : space) {
   if (room != null && room.isRoom( name, num )) {
      System.out.println(room.getCapacity());
      break look;
   }
}
```

What is displayed?

```
Classroom  here;
space[5].setCapacity( 47 );
here = space[5];
here.setCapacity( 4 );
System.out.println( space[5].getCapacity() );
```

A. 4

B. 5

C. 47

D. None of the above

String Methods

- There are many methods that can be called on objects of the String class
- length() Returns the length of this string

```
String lecture = “GEEN163”;
int howLong = lecture.length();    // 7
```

Searching Strings

- You can search a string to see if it contains the another string
- indexOf(String target) Returns the index within this string of the first occurrence of the specified String or -1 if not found

indexOf Example

- The position of a character in a string starts counting at zero

```
String major = "Computer Science";
               //  0123456789012345
```

```
int where = major.indexOf( "put" );      // 3
where = major.indexOf( "fail" );         // -1
```

Taking Strings Apart

- substring(int beginIndex, int endIndex) Returns a new string that is a substring of this string. The substring begins at beginIndex and extends to the character at position **endIndex - 1**. Thus the length of the substring is endIndex – beginIndex

- charAt(int index) Returns the char value at the specified index

substring Example

```
String major = "Computer Engineering";
               //          1111111111
               //0123456789012345678 9
```

String result = major. substring(11, 14);

result has the value "gin"

Changing Case

- toUpperCase() returns a copy of the String with all of the letters in upper case
- toLowerCase() returns a copy of the String with all of the letters in lower case

```
String gnu = "Today is 8/29/12";
String wildebeest = gnu.toUpperCase();
System.out.println(gnu +"\n"+wildebeest);
```

```
Today is 8/29/12
TODAY IS 8/29/12
```

What is the value of result?

```
String major =   "Computer Engineering";
                //   01234567890123456789
String result = "S" + major. substring( 12, 14 );
```

A. Sine
B. ine
C. Sin
D. in
E. S

Miscellaneous String Method

- trim() Returns a copy of the string, with leading and trailing whitespace omitted

```
String hair = "  What ever    ";
              // 01234567890123
String bald = hair.trim();
// bald is  "What ever";
```

Interesting String Methods

- length
- indexOf
- substring
- charAt
- toUpperCase
- toLowerCase
- trim

Write with your Team

- Given a string containing a first name and last name, break it into two strings, one with the first name and the other with the last name

String full = "Barack Obama", first, last;

The result would be

first is "Barack" and last is "Obama"

Possible Solution

```
String full = "Barack Obama", first, last;
int blankPos = full.indexOf( " " );
first = full.substring( 0, blankPos );
last = full.substring( blankPos+1, full.length() );
```

Who Dunit?

- Many GUIs have more than one active object
- You can determine where the action was taken by using the **getSource()** method of the event
- The result of **getSource()** should be compared to each GUI object

```
public void actionPerformed(
                java.awt.event.ActionEvent   frog ) {
    if ( frog.getSource() == GUIobject ) {
```

Example Use of getSource()

```
JButton doit = new JButton("OK");
JButton skip = new JButton("cancel");
        - - - - - - - - - - - - - - - - - - - - - - - - -
public void actionPerformed(
                        java.awt.event.ActionEvent   toad) {
    if (toad.getSource() == doit ) {
        // do it
    } else if (toad.getSource() == skip ) {
       // don't do it
    }
```

Course Evaluations

- Course evaluations are available on Blackboard
- Be sure to complete all evaluations for all classes
- 22% of the student in GEEN163 have completed the course evaluation

Website Updated

The GEEN163 website
http://williams.comp.ncat.edu/geen163/
has been updated

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

- Today's lecture gave you all sorts of hints