Using a Debugger
Arrays
Lecture 21

Borrowing from slides by Alan Hu, Kurt Eiselt, Paul Carter, and Tamara Munzner

News
- Assignment 2
  - Due this Friday, Oct 30
- Midterm 2 coming up next week
  - Wednesday, Nov 4, 6:30-7:30
  - Note: the rooms will be different
  - Material: conditionals, loops
    - Arrays will NOT be on the midterm

Reading Assignments
- Reading for this week: arrays
  - Edition 3: Ch. 7.1, 7.5-7.7
  - Edition 2: Ch. 8.1, 8.5-8.7

Objectives for Today
- Learn different strategies for debugging code
- Gain an initial understanding of arrays

Example from Last Time
- Modify the following code to compute the correct payment for paying off a loan in a given term

```java
double balance = initBalance;
int years = 0;
while (years < term) {
    intAccrued = balance*intRate/100;
    balance += (intAccrued - payment);
    years++;
}  
```

Loan Amortization
- How do we compute the right payment amount to pay off a loan after some number of years?
- Use “Nested Intervals” approach:
  - Have two guesses: tooLow and tooHigh
  - Try a guess halfway in between.
  - Compute the loan balance using guess.
  - If guess was too high, then tooHigh = guess else tooLow = guess.
  - Repeat
Finding Bugs

- How can we find the bugs that we encountered last time
  - Stare at the code until we “get it”
  - Manually trace the code, keeping track of the variables
  - ???
  - ???

Finding Bugs

- How can we find the bugs that we encountered last time
  - Stare at the code until we “get it”
  - Manually trace the code, keeping track of the variables
  - Print variable contents at various key places in the program
  - Use a debugger (special program for tracking down errors)

Loan Amortization

- How do we know that our program will terminate?
  - Inner Loop: years starts at 0 and counts up to term. Will always execute exactly term times.
  - Outer Loop: The gap between tooLow and tooHigh gets cut in half each iteration.

- These are termination arguments (aka ranking functions). You should always know why your loops will terminate.

Loan Amortization

- How do we know that our program will compute the correct result?
  - Inner Loop: At each iteration, balance is always the correct value after years years.
  - Outer Loop: At each iteration, tooLow is always less than the correct value, and tooHigh is always greater than the correct value.

- These are called “loop invariants”. Very helpful to understand loops. (Great to put in comments!)

Arrays

- Arrays let you create a bunch of variables (all of the same type), that you refer to by array name and number (called the index).

Example:
  ```java
  int[] x = new int[10];
  ```
  instead of
  ```java
  int x0, x1, x2, x3, x4, x5, x6, x7, x8, x9;
  ```
Declaring Arrays

- Must declare, just like any other variable.
- Declarations look like creating objects:
  Scanner s = new Scanner(System.in);

  int[] x = new int[10];

  The variable name. In both cases, these are reference variables.

Declaring Arrays

- Must declare, just like any other variable.
- Declarations look like creating objects:
  Scanner s = new Scanner(System.in);

  int[] x = new int[10];

  The keyword new to allocate memory

Declaring Arrays

- Must declare, just like any other variable.
- Declarations look like creating objects:
  Scanner s = new Scanner(System.in);

  int[] x = new int[10];

  The constructor for the object or array

Declaring Arrays

- Must declare, just like any other variable.
- Declarations look like creating objects:
  Scanner s = new Scanner(System.in);

  int[] x = new int[10];

  Type of the variable

- Syntax:
  type[] variableName = new type[size];

- Examples:
  double[] dataSet = new double[30];
  int[] quizScores = new int[5];
  String[] args;
Using Arrays
- Syntax:  
  `variableName[index-expression]`
- If `var` is an array (reference) variable of type `type[]`, then `var[expr]` is a variable of type `type`.
- Examples:
  ```java
  double[] dataSet = new double[30];
  dataSet[0] = dataSet[1]/2;
  ```

Array Length
- Syntax:  (note no parentheses)  
  `variableName.length`
- If `var` is an array (reference) variable, then `var.length` is an int containing number of entries in array.
- Examples:
  ```java
  double[] dataSet = new double[30];
  dataSet.length == 30
  ```
  Note that indices go 0 .. dataSet.length-1!

Using Arrays
```java
int[] x= new int[10];
x[0]= 5;
x[1]= 7;
x[9]= x[0]+x[1];
```

Array Initializers
- Array creation and initialization can be combined:
  ```java
  int[] x= {1,2,3};
  String[] colors= {"red", "green", "blue"};
  ```

Teen Talk Barbie™
- In 1993, Mattel introduced Teen Talk Barbie™, a doll programmed to speak random phrases. Each doll had a random selection of 4 phrases from a list that included:
  - You can never have enough clothes.
  - Let's go shopping!
  - Math class is tough.
  - (Talking GI Joe doll said “Vengeance is mine!” and “Dead men tell no lies.”)

Customizable Talking Doll
- Imagine a customizable doll instead.
- When first turned on, parents put in five phrases.
- After that, each time activates doll, it randomly says one of the phrases.
- How would you program this?
  - For simplicity, let’s just use strings.