Class 18

Recursion, Arrays

Example 1

- findPower
  - compute an integer raised to a power using recursion

Example 2

- evenDigits
  - returns an integer consisting only of the even digits
  - if no even digits exist, returns 0

Arrays

- If a variable is a box in which we can store a specific data type (such as int, double, char) then we can think of an array as a row of boxes
- The row of boxes can be almost any quantity and type, however all the boxes must be of the same type

Working with arrays

- Must declare an array before we can use it
- Model: BASETYPE NAME[CAPACITY]
  - int someInts[10];

- Arrays can also be initialized to specific values
- Example: int someints[5] = {7, 8, 12, 9, 2};

Working with arrays

- Store a value in the first box of the array
  - someInts[0] = 11;
- Store a value in the second box of the array
  - someInts[1] = 15;

- Notice that the first element in an array is stored at index 0

- To process entire arrays, we typically use for loops, with the counter starting at 0

Example 3

- User enters five numbers
- Print the numbers in reverse order

Terminology

- An array called someInts gives us access to lots of variables like someInts[1], someInts[2], someInts[3], etc.
- These variables are called the ELEMENTS or ENTRIES of the array
- The number in [] is called the INDEX of the element
- The index can be an actual number, e.g., someInts[3], a variable like someInts[i], or an expression like someInts[n+2]

Example 4

- Goal:
  - Initialize array to five quiz scores
  - Compute the average of the scores
  - Print all scores that are above average

Example 5

- Goal:
  - Initialize array to 10 random numbers between 1 and 20
  - Print the array
  - Print the even numbers in the array from left to right
  - Print the odd numbers in the array from right to left