

Functions and Loops - More Details

Atul Prakash

Advanced Loops

- "break" statement to prematurely terminate a loop
- "continue" statement to jump to the next iteration of the loop

Example of Break

```
while (true) {  
    String stringguess = helper.getUserInput("Please enter your guess: ");  
    int guess = Integer.parseInt(stringguess);  
    game.shoot(guess);  
    if (game.isGameDone()) { // game is done  
        break;  
    }  
}
```

Q: What if user types in a negative guess?

Use of continue

```
while (true) {  
    String stringguess = helper.getUserInput\("Please enter your guess: "\);  
    int guess = Integer.parseInt(stringguess);  
    if (guess < 0) {  
        System.out.println("Illegal input value");  
        continue;  
    }  
    game.shoot(guess);  
    if (game.isGameDone()) { // game is done  
        break;  
    }  
}
```

Function Return Values

- Unlike Python, Java can only return one value from a function. "return" statement.
- But the value can be an object reference (e.g., array, Java lists, etc.)
- Thus, not a big limitation

Function Parameters

- Passed by copy
- Object references: references are copied -- just like assignments
- If object is modified, the change is reflected in the caller.

Example

```
public static void onedimensional() {
    int nums[];
    nums = new int[10];

    for (int i = 0; i < 10; i++) { // Initialize the array
        nums[i] = i;
    }
    square(nums); // Pass nums argument (object reference)
    for (int i = 0; i < 10; i++) { // print out the array
        System.out.printf("index = %d, array content = %d\n", i, nums[i]);
    }
}

public static void square(int[] a) {
    for (int i = 0; i < a.length; i++) {
        a[i] = (int) Math.pow(a[i], 2); // Java does not have power operator.
    }
    return;
}
```

