Lecture 13: debugging, do-while and for loops
Feb 12 2015
Debugging

• Logic errors are called **bugs**. The process of finding and correcting errors is called **debugging**.

• To locate the part of the program where the bug is located, you’ll want to use a combination of
  – *hand-tracing* the program (finding errors by simply reading the program)
  – insert *print statements* in order to show the values of the variables or the execution flow of the program.
  – use a *debugger utility* (a particularly good idea for a large, complex program)
Debugger

• Debugger is a program that facilitates debugging. You can use a debugger to
  – Execute a single statement at a time.
  – Trace into or stepping over a method.
  – Set breakpoints.
  – Display variables.
  – Display call stack.
  – Modify variables
Software Development Process

- Requirement Specification
- System Analysis
- System Design
- Implementation
- Testing
- Deployment
- Maintenance
A formal process that seeks to understand the problem and document in detail what the software system needs to do. This phase involves close interaction between users and designers.

Most of the examples in this book are simple, and their requirements are clearly stated. In the real world, however, problems are not well defined. You need to study a problem carefully to identify its requirements.
System Analysis

Seeks to analyze the business process in terms of data flow, and to identify the system’s input and output.

Part of the analysis entails modeling the system’s behavior. The model is intended to capture the essential elements of the system and to define services to the system.
System Design

The process of designing the system’s components.

This phase involves the use of many levels of abstraction to decompose the problem into manageable components, identify classes and interfaces, and establish relationships among the classes and interfaces.
Implementation

The process of translating the system design into programs. Separate programs are written for each component and put to work together.

This phase requires the use of a programming language like Java. The implementation involves coding, testing, and debugging.
Testing

Ensures that the code meets the requirements specification and weeds out bugs.

An independent team of software engineers not involved in the design and implementation of the project usually conducts such testing.
Deployment makes the project available for use.

For a Java applet, this means installing it on a Web server; for a Java application, installing it on the client's computer.
Maintenance

A software product must continue to perform and improve in a changing environment. This requires periodic upgrades of the product to fix newly discovered bugs and incorporate changes.
**do-while** loops

Syntax:

```java
do {
    // loop-body;
    Statement(s);
} while (loop-continuation-condition)
```
**while** vs. **do-while**

**while** loop:
- **Loop Continuation Condition?**
  - true: **Statement(s) (loop body)**
  - false: end loop

**do-while** loop:
- **Statement(s) (loop body)**
- **Loop Continuation Condition?**
  - true: **Statement(s) (loop body)**
  - false: end loop

**Example**:
```java
int count = 0;
while (count < 100) {
    System.out.println("Welcome to Java!");
    count++;
}
```
while vs. do-while

```java
int count = 0;
while (count < 4) {
    System.out.println("Welcome to Java "+ count);
    count++;
}

int count = 0;
do {
    System.out.println("Welcome to Java "+count);
    count++;
} while (count < 4)
```
for loops

Syntax:
for (initial-action; loop-continuation-condition; action-after-each-iteration)
{
  // loop body;
  Statement(s);
}

Example:
int i;
for (i=0; i < 100; i++) {
  System.out.println("Welcome to Java");
}

Diagram:

Initial-Action

Loop Continuation Condition?
  true
  Statement(s) (loop body)
  Action-After-Each-Iteration
  false

i = 0

(i < 100)?
  true
  System.out.println("Welcome to Java");
  i++
  false
Note

• The initial-action in a for loop can be a list of zero or more comma-separated expressions.

• The action-after-each iteration in a for loop can be a list of zero or more comma separated statements.

Therefore, the following two for loops are correct. They are rarely used in practice, however.

```java
for (int i = 1; i < 100; System.out.println(i++));

for (int i = 0, j = 0; (i + j < 10); i++, j++) {
    // Do something
}
```