Ch5: for, while, do, switch

29 January 2007
CMPT167
Dr. Sean Ho
Trinity Western University

Review of last time

- Operator precedence
- Java primitive types
- public/private keywords
- set/get methods
- Subclasses, instances, attributes
- Interfaces

What's on for today

- More on applets; running applets
- While loops
- Do/while loops
- For loops as while loops
- Switch/case statement
- Labeled blocks

More on applets

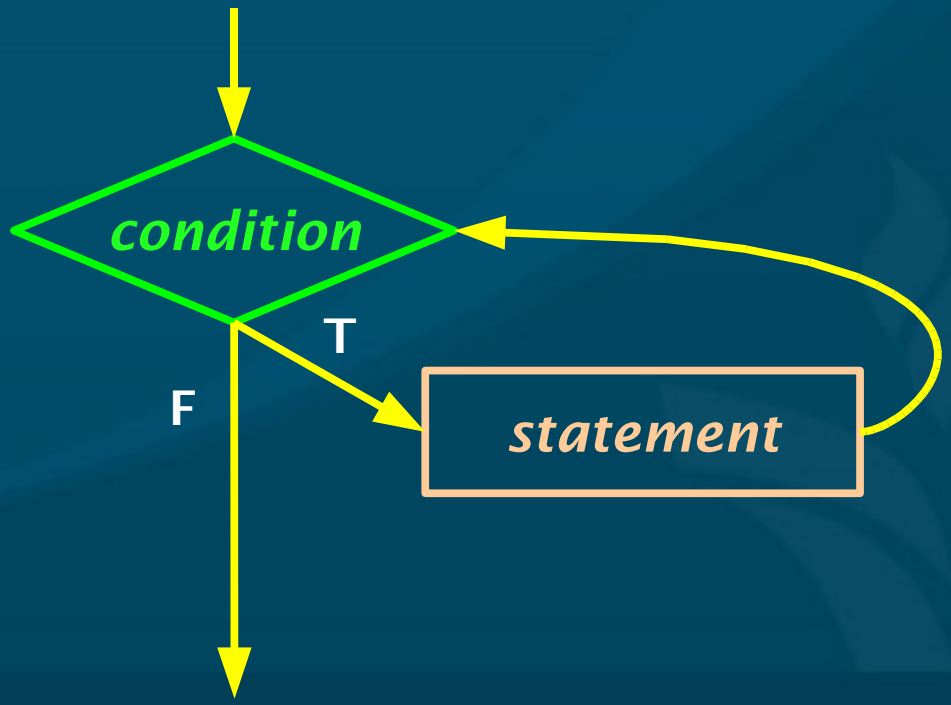
- Addition example applet ("Lab0")
  - import java.applet.Applet; // the Applet class
  - import java.awt.*; // abstract window toolkit
  - public class Addition extends Applet implements ActionListener {
- Only one public class per file
  - Named same as the file
- Extends: subclass inherits from Applet
- Implements: ActionListener interface
  - Must implement the following method:
  - public void actionPerformed(ActionEvent e)

Running an applet

- Compile an applet using javac as usual
- Run: in Eclipse: it will pop up a window
- Run: in a web page:
  - Write a small HTML file embedding the applet:
    - <object> (IE), or <applet>, or both
    - See Addition.html and TicTacToe.html
    - See Sun's recommendations
- Run: using appletviewer:
  - appletviewer Addition.html
    - HTML file, not the applet .class file directly

While loops

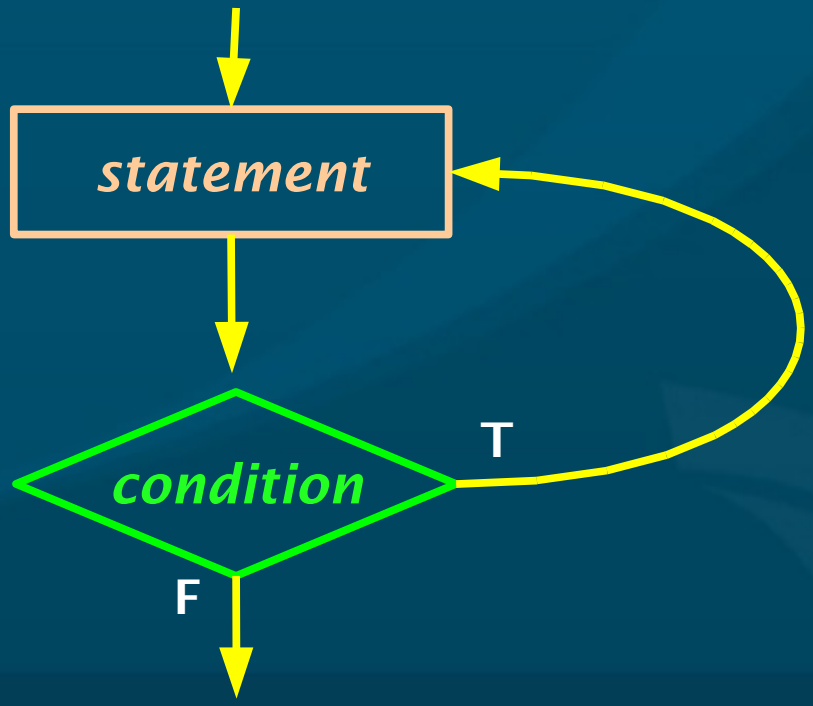
- while (*condition*) *statement*;
- As usual, statement can be a {} block
- *condition* evaluates to a boolean
- Top-of-loop testing



do/while loops

- do *statement* while (*condition*);

- As usual, statement can be a {} block
- *condition* evaluates to a boolean
- Bottom-of-loop testing



For loops as while loops

- Pretty much every for loop:
  - for (*init*; *condition*; *increment*) *statement*;
- ... can be expressed as an equivalent while loop:
  - *init*;
  - while (*condition*) {
    - *statement*;
    - *increment*;
  - }

break/continue

- Use break to terminate a loop early:
  - for (i=0; i<10; i++) {
    - if (i==5) break; // quit at 5
  - }
- Use continue to skip to the next iteration of the loop:
  - for (i=0; i<10; i++) {
    - if (i==5) continue; // don't print 5
    - System.out.print(i);
  - }

Switch statement

- switch (*expression*) {
  - case *val1*: *statement; ...;* break;
  - case *val2*: *statement; ...;* break;
  - *...*
  - default: *statement; ...;*
- }

- Similar to a nested if/else structure
  - But *expression* is only evaluated once
- If omit a break, execution continues to next case:
  - case *val1:*
  - case *val2*: *statement; ...;* break;

Labeled blocks

- Blocks can be named
- break/continue can specify a name:
  - Go to start/end of named block

```
main: {
    for (row=0; row<n_rows; row++) {
        for (col=0; col<n_cols; col++) {
            if (row+col == 12) break main;
        }
    }
}
```

TODO

- Lab1b due this Wed 31Jan:
  - Repetition structure
  - Swing program: see "SayHello" example, or
  - Java Applet: see "Lab0" (Addition) template