Lecture 6 (wed), Lecture 5 (sat)

Topics
Discuss Lab 2 due next week – uploading files to hills
Ch 3.1 – 3.4 Conditional code structure
Rewrite flowcharts as logic diagrams on pp.118,123,129,134,141
Relational operators

Test 1

1. Rewrite flowcharts:

   ![Flowchart Diagram]

   p. 118

2. Relational Operators

   `<`, `>`, `<=`, `>=` (relational/comparison) `==`, `!=` (equality)

   I disagree with the author’s precedence list. There is no precedence, since you can only use one relational operator in an expression, unless you use a logical operator. Logical operators have their own rules, which we will see next week.

3. Conditional code structure - Using if and if-else

   Control Structures determine the flow of control of the program. (sequence, selection, and repetition) Recall the logic diagram shapes.

   selection: if, if-else, switch

   1 or more alternate actions (branches) are executed depending upon the evaluation of a boolean expression, which must evaluate to true or false (not 1 or 0 as in C++)

   //“if” syntax:
   
   if (boolean expression)
       Statement;

   true
   false

   Cold?

   true

   false

   Wear a coat
Examples:
if (nNumIn >= MAX)
    nNumIn = MAX;

if (nNumIn >= MAX) nNumIn = MAX;

if (sReply == “yes”) {
    sReply = JOptionPane.showInputDialog(“are you sure? Type yes or no”);
    nTotal = 10;
}

/*if – else” syntax:*/
if (boolean expression)
    statement;
else
    statement;

Example:
if (nTotal > 100)
    System.out.println(“congratulations, you won the game!”);
else
    System.out.println(“Sorry, you must score 100 to win”);

// “if-else” when bodies contain more than one statement
// enclose the body within braces (define it as a block)
if (boolean expression) {
    statement;
    statement;
} else {
    statement;
    statement;
}

Example:
if (nTotal < 50 ) {
    g.setColor(Color.red);
    g.drawString(“Too bad – you lost!”, 100, 150);
} else {
    g.setColor(Color.blue);
g.drawString(“Congratulations – you won!”, 100, 150);

Note that blocks need not be symmetrical:

if (boolean expression) {
    statement;
    statement;
} else
    statement;

4. Nested conditional structures:

if (boolean expression 1)
    if (boolean expression 2)
        statement;
    else
        statement;

Problem of the “dangling else”:

with nested if-else, the else will match with the closest if, unless you use braces
to show matching:
// this code contains a logic error
if (nNum < 10)
    if (nNum < 5)
        System.out.println("your number is smaller than 5");
else
    System.out.println("your number is greater or equal to 10");

I meant for the else to match the outer if, but it will match the inner if. So if nNum
is 8, the output will be "your number is greater or equal to 10". Bad or incorrect
output means there is a logic error.
To force matching, put braces around the outer if’s block, because the if-else match
cannot cross block boundaries.

if (nNum < 10) {
    if (nNum < 5)
        System.out.println("your number is smaller than 5");
} else
    System.out.println("your number is greater or equal to 10");

Using the braces here to force the matching is like using parentheses in an expression to force the order of evaluation.

5. **Boolean evaluation:**
This is not legal in Java:

```java
int nNum = 0;
if (nNum) //syntax error, int is not boolean
```

But this is legal:

```java
boolean bDone = true;
if (bDone)// ok to use boolean variable, called a boolean flag
```