COP 2250 – Java Programming 1 – Chapter 3 - Selections

Boolean Operations

* Recall that a Boolean variable can have only one of two values: true or false.
* Choose self-documenting names for Booleans like “isTeenager” or “hasCash”.
* Booleans are often used with operators to create expressions.

Comparison (aka Relational) Operators (page 76)

* These operators test a relationship between two values and return a Boolean.

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| < | less than |
| > | greater than |
| <= | less than or equal to |
| >= | greater than or equal to |
| == | equal to |
| != | not equal to |

if Statements (page 78-87)

* If statements enable Java programs to branch and execute along alternate paths.
* These use the operators above to create Boolean expressions.

Simple ifs do not use an else block.

if(Boolean expression) {

// statement(s) to run if expression is true

}

* Try SimpleIfDemo on the textbook’s web site.

Two-way if-else blocks consist of both if and else blocks.

if(Boolean expression) {

// statement(s) to run if expression is true

}

else {

}

// statements to run if expression is false

Nested ifs and Mult-Way if-else Statements

* else if blocks are used for more than two possibilities.
* if and/or else blocks can be nested within other ifs to any depth desired.
* using good indenting form will promote readability.
* NOTE the preferred format for multiple alternative ifs on page 82.

Common Errors and Pitfalls

* See page 83-87

|  |  |
| --- | --- |
| **Avoid This** | **Do This Instead** |
| **if(num == true) {**  **}** | **if(num) {**  **}** |
| **if(num == false) {**  **}** | **if(!num) {**  **}** |

Generating Random Numbers (See page 87)

* These can be generated with the random( ) method of the Math class.
* To generate a random number between 0 and 100, use:

int number = (int)(Math.random( ) \* 101);

* To generate a random number between 1 and 100, use:

int number = (int)(Math.random( ) \* 100) + 1;

* Try SubtractionQuiz on the textbook’s web site.
* Try ComputeAndInterpretBMI on the textbook’s web site.
* Try ComputeTax on the textbook’s web site.

System.exit(0);

* This statement will immediately stop an executing Java program (see page 92).

Boolean (Logical) Operators (page 93)

* These connect two Boolean values to create a new Boolean value.

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| ! | logical NOT |
| && | logical AND |
| || | logical OR |
| ^ | logical XOR |

* Study the truth tables on pages 93-94.
* Use && when both expressions must be true
* Use || when either expression can be true
* Use ^ when either expression can be true, but not both
* && performs a short-circuit test. If the left side of the && evaluates as false, the other side is not evaluated (since the entire expression will be false)
* NOTE: if condition is compound, each part must be capable of standing alone.

|  |  |
| --- | --- |
| **Incorrect** | **Correct** |
| **if(num == 4 || 8)** | **if(num == 4 || num == 8)** |

* Try LeapYear on the textbook’s web site
* Try Lottery on the textbook’s web site

Exercise

Write a program called Teenager.java that prompts the user to enter his/her age (as an integer) and then reports whether the user is a teenager or not. Use a compound expression as the if condition.

Exercise

To vote in the U.S.A, a person must be a citizen and at least 18 years old. Write a Java program called Voting.java that prompts the user for his/her age, asks the citizen question, and reports on voting eligibility. Use nested ifs.

The switch statement (See page 100)

* This control structure often offers better coding than a series of if … else if blocks when your program requires multiple branches. The general format is:

switch ( expression ) {

case constant 1:

statement(s)

break;

case constant 2:

statement(s)

break;

case constant 3:

statements(s)

break;

…

default:

statement(s) to run if no case above is matched

} // end of switch

* note the colon after each switch case label.
* the break statements are required, otherwise code in following cases is executed, too.
* the default statements are executed if no cases are a match.
* NOTE: switch case labels can be stacked, if desired. See page 101.

Conditional Expressions (See page 103)

* This amazingly terse operator is an alternative to an if…else set.
* The syntax is:

(Boolean expression) ? true statement : false statement

* Example:

int numberOne = 8;

System.out.println((numberOne < 10) ? “Less than 10” : “10 or more”);

* The output from the statements above is: Less than 10

Exercise

Write a program named OddEven.java that prompts the user to enter an integer, and then reports whether the entry is odd or even. Use a conditional expression to generate the odd or even response.

Operator Precedence (See page 104)

* NOTE Table 3.8 on page 105.

Debugging

* Note the bullet list on page 106.
* See also the “Eclipse Debugging” video in D2L.