# *Programming I (420-B10-HR)*

# *Lab 8 – The if Statement*

Date assigned: Tuesday, October 13, 2015

Date due: **Tuesday, October 13, 2015**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

1. use a simple **if** statement to compare two values;
2. use compound statements in a Java **if** statement;
3. use a nested **if** statement;
4. use the **equals()** and **equalsIgnoreCase()** methods to compare String values;
5. use a **switch** statement.

**Statements Used:**

1. Simple *if ... else* statement:

**if** condition

true statement**;**

**else**

false statement**;**

2. Compound statements in an *if ... else* statement:

**if** condition

**{**

block of statements to be executed if condition is true;

**} // end if**

**else**

**{**

block of statements to be executed if condition is false;

**} // end else**

3. Nested *if ... ... else* statement:

**if** condition

**if** condition

true statement**;**

**else**

false statement**;**

**else**

**if** condition

true statement**;**

**else**

false statement**;**

4. *Switch* statement:

**switch (**integralVariable**)**

**{**

**case** integralValue1**:**

statement1**;**

break;

**case** integralValue2**:**

statement2**;**

break;

**. . .**

**case** integralValue*n***:**

statement*n***;**

break;

**default:**

defaultStatement**;**

} // end switch

**To Be Handed In:**

1. The ***username*\_B10\_L08\_if** folder should be zipped and uploaded to **Moodle**.

**To Start:**

1. Log on to **Moodle**, go to the **Programming I** course page and complete the **Lab 8 Terminology Quiz**.
2. Download and unzip the **B10\_L08\_ if** folder from **Moodle** to your **H:\420-B10\Labs** folder. Rename it to ***username*\_B10\_L08\_ if**.
3. Start **Eclipse**. Use your **H:\420-B10\Labs** folder as the workspace.
4. Create a **New Java Project** called ***username*\_B10\_L08\_ if**.

# Using the Simple *if ... else* Statement

***Purpose*:** Use a simple if ... else to compare two values.

***To Do***:

## Open the **Grader** class and change the author name to yourself.

## Replace the **getStatus()** method with the following:

public String getStatus()

{

if (mark < 60.0)

return "Poor.";

else

return "Passed";

} // getStatus()

## Run the **GraderUser** program with the following test data. Complete the table with the output produced.

|  |  |
| --- | --- |
| ***Mark*** | ***Output*** |
| 59.0 | **poor** |
| **60.0** | **passed** |
| **75.5** | **passed** |
| **100.0** | **passed** |
| **101.0** | **passed** |

## Add an if...else statement to the **setMark()** method in the **Grader** class to set the mark if the parameter **m** is less than or equal to 100. Otherwise use **System.err.println()** to print ***m* is an** **invalid mark. The mark must be less than or equal 100** in red. (The value for ***m*** should be displayed, not the letter ***m***.)

## Test your modified program.

# Using Compound Statements in an if…else

Purpose: Learn to use a compound statement in an if…else statement.

To Do:

## Run **TemperatureUser** using the test data in the table shown here. Complete the table with the output produced.

| **Temp** | **Units** | **Converted Temperature** | **Converted Units** |
| --- | --- | --- | --- |
| 32 | F | 0 | C |
| 212 | F | 100 | C |
| 0 | C | 32 | F |
| 100 | C | 212 | F |

## Open the **Temperature** class and change the author to yourself.

## Change the { and } lines in the **if** statement in the **setTemperature()** method to comments as shown below. ( *Note*, you can add and delete comments to and from a line quickly using **CTRL-/** )

if (units == 'C')

// {

newDegrees = convertToFahrenheit(); //1

newUnits = 'F'; //2

// }

if (units == 'F') //3

// {

newDegrees = convertToCelsius(); //4

newUnits = 'C'; //5

// }

## Run the **TemperatureUser** program for **temp = 0** and **units = C** and for **temp = 32** and **units = F**.

Complete the following table by placing a check in the box if the corresponding statement number is executed. (See above for statement numbers.)

|  |  |  |
| --- | --- | --- |
| Statement | If **units** is '**C**' | If **units** is '**F**' |
| 1 | Yes |  |
| 2 | No |  |
| 3 |  | yes |
| 4 |  | yes |
| 5 |  | yes |

What is the effect of the **{** and **}**?

\_\_Makes sure every line is run\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A *compound statement* is any number of single statements contained between braces ({ }). A *compound statement* may be used anywhere in a Java program that a single statement may be used.

## Remove the comment //’s you added in step 3.

## Modify the **Temperature** class to allow the user to convert a **Kelvin** temperature to **Celsius**. (**Note**: a Kelvin temperature is 273.16 degrees higher than the corresponding Celsius temperature.) Call your new method **convertKelvinToCelsius()**.

## Add statements to the **setTemperature()** method to handle the case where units = ‘K’.

## Change the prompt in the **main()** method in **TemperatureUser** to prompt for K if a Kelvin temperature is entered.

## Test your modified program. (273.16 K should be 0 C)

# Using a Nested *if ... else* Statement

***Purpose*:** Use a nested if ... else to compare multiple values.

***To Do:***

## Open the **Grader** class. Add a **char** instance variable called **grade**. Initialize it to 'U'.

## Add the following method:

private void setGrade()

{

if (mark >= 90)

grade = 'A';

else if (mark >= 80)

grade = 'B';

else if (mark >= 60)

grade = 'C';

else

grade = 'F';

} // setGrade()

## Add a call to **setGrade()** in the **setMark()** method after the statement that sets **mark** to **m** and before the **else**. You will have to add braces to make a compound statement.

## Add an accessor (**getGrade()**) that returns the grade.

## In the **GradeUser** class, add a **println** to call the **getGrade()** method before displaying status. Include a message in the **println** to indicate that this is the grade. (i.e. If the mark is 90, the output should be: *Your grade is A*)

## Test **GraderUser** on several values.

## Modify the **setGrade()** method so that a mark between 60 and 69 has a grade of D. (Note: A mark between 70 and 79 should still be a C.) Test your changes.

# Comparing String Variables

Purpose: Use the equals() method to compare String values.

To Do:

## Open **StringComparer.java** and change the author to yourself.

## Run **StringComparer**. Type any word.

## Modify the class to print "*Hello yourself*" if "*hello*" is entered and to print "*What’s up?*" if any other word is entered.

## Run the project. Type "hello". What is output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Explanation:***

String is not a primitive data type. It is a class and therefore a String variable is a reference variable. Using == compares the two references and not the values referenced. To compare the values you must use the **equals()** method

## Modify the **if** statement to:

if (word.equals("hello"))

## Run the project again. Type "hello". What is output now? \_\_hello yourself\_\_

## Run the project again. Type "HELLO". What is output now? \_\_what’s up? \_\_\_\_\_

***Explanation:***

The **equals()** method looks for an exact match in case. To ignore the case use the **equalsIgnoreCase()** method.

## Modify the **if** statement to:

if (word.equalsIgnoreCase("hello"))

## Modify the class to display "*See you later*" if "*goodbye*" is entered. Test your changes.

# Using a switch Statement

***Purpose*:** Use a **switch** statement to select one of several alternatives.

***To Do:***

## Open the **TestSwitch** class and run itfor the following values:

|  |  |
| --- | --- |
| marcode | Output |
| M | Individual's marital status is married. |
| S | Individual's marital status is single. |
| D | Individual's marital status is divorced. |
| W | Individual's marital status is widowed. |
| m | Individual's marital status is married. |
| Q | Individual's marital status is unknown. |

## Change the author of the **Person** class to yourself.

## The **getFullStatus()** method in the **Person** class uses a nested if to select the correct alternative. Another way to do the same thing is to use the **switch** statement. Change the **getFullStatus()** as follows:

public String getFullStatus()

{

String status;

switch (maritalStatus)

{

case 'M':

{

System.out.println("in case 'M'");

status = "married";

} // case 'M'

case 'S':

{

System.out.println("in case 'S'");

status = "single";

} // case 'S'

case 'D':

{

System.out.println("in case 'D'");

status = "divorced";

} // case 'D'

case 'W':

{

System.out.println("in case 'W'");

status = "widowed";

} // case 'W'

default:

{

System.out.println("in default");

status = "unknown";

} // default} // switch

return status;

} // getFullStatus()

## Run the **TestSwitch** program again for the same values. What happens?

## \_\_\_\_It shows all of the possible entries for the marital status and what would be displayed if an invalid character was entered.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Notice that it does not execute any of the cases before the one that it matches. To get the switch statement to exit when if finds a match add the following line before the closing brace in each **case**:

**break;**

## Test your changes.

## Modify the **switch** statement to return **child** if C is entered as the marital code.

## Test your modified program.

## Open the **CalculatorSwitch** class.

## Run **CalculatorSwitch** for the following values and complete the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **num1** | **Num2** | **Opselect** | **Arithmetic operation performed** |
| 5 | 6 | 1 | 5 + 6 = 11 |
| 5 | 6 | 2 | 5 \* 6 = 30 |
| 5 | 6 | 3 | 5 / 6 = .833333 |
| 5 | 6 | 4 | 0.0 |

## Modify the **CalculatorSwitch** and **Calculator** classes to include subtraction as option 4.

## Test your modified program.

# Review Questions

Purpose: Review the if and switch statement concepts and syntax.

To Do:

## Do the **Lab 8 Review Questions** quiz on **Moodle**.