Unit 4 ~ Conditions

Unit 4A

Statements that branch to perform one action or another depending upon a condition are used to help a program make decisions.

The If Statement

The if statement executes a set of statements when a condition is true.

```
if (<condition>) {
    <statements>
} //end if
```

Note the ***comparison operator*** is double ==, not a single =.

= means assignment
== means comparison

For example,

```
if (guess == SECRET_NUM) {
    System.out.println("You guessed it!");
}
```

The *condition* of an if statement is a ***boolean expression*** which evaluates to either **true** or **false.** These expressions are formed with one or more of the 6 ***relational operators*** shown below.

Relational Operator	
==	equal
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
!=	not equal to

Another example

```
if (gameOver == true)
```

is equivalent to

```
if (gameOver)
```

Never use floats in a condition. Most often, a *roundoff* error will occur.
For example, if you use if (5.0 * 5 == 25) the answer would be **false**. Floating points cannot be accurately represented in binary.

eview: SurfsUp – part 1 of 3

Create a SurfsUp application that prompts the user for the wave height and then displays "Great day for surfing!" when the waves are 6 feet and over.

The if-else statement

```
if (<condition>) {
    <statements A>
} else {
    <statements B>
}
```

In this statement, statements A would execute if the condition is true. If it is not true, statements B would execute.

Your Turn

Review: SurfsUp – part 2 of 3

Modify the SurfsUp application to display "Great day for surfing!" when the waves are 6 feet or over and "Go body boarding!" when the waves are less than 6 feet.

Nested If Statements

An **if statement** can contain another **if statement.**

```
if (guess == SECRET_NUM) {        //correct
   System.out.println("You guessed it!");
} else {
   if (guess < SECRET_NUM) {      //too low
      System.out.println("Too low.");
   } else {                       //too high
      System.out.println("Too high.");
   }
}
```

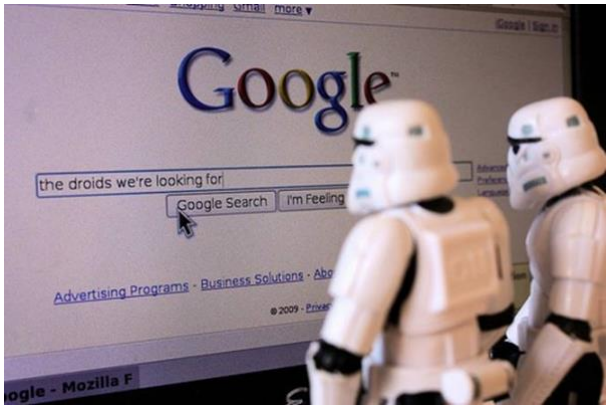
Statements placed within the same type of statements are called *nested.*

Make sure to use careful indenting.

Your Turn

Review: Stages

Create a Stages application that prompts the user for an age. For an age over 18, `adult` is displayed. For an age less than or equal to 18, `toddler` is displayed when the age is less than or equal to 5, `child` when the age is less than or equal to 10, `preteen` when the age is less than or equal to 12, and `teen` when the age is over 12.



The if-else-if Statement

This statement is used to decide among 3 or more actions and takes the form:

```
if (<condition A>) {
    <statements A>
} else if (<condition B>) {
    <statements B>   //you can have more else-if statements here

} else {
<statements C>  //this last one is optional
}
```

The logic used in this statement is important. The conditions must be properly ordered.

For example:

```
if (guess == SECRET_NUM) {          //correct
   System.out.println("You guessed it!");
} else if (guess < SECRET_NUM) {    //too low
   System.out.println("Too low.");
} else {                            //too high
   System.out.println("Too high.");
}
```

Review: SurfsUp – part 3 of 3

Modify the SurfsUp application to display "Great day for surfing!" when the waves are 6 feet or over, "Go body boarding!" when the waves are between 3 and 6 feet, "Go for a swim." when the waves are from 0 to 3 feet, and "Whoa! What kind of surf is that?" otherwise.

Compound Boolean Expressions

We can have multiple conditions by using **logical operators.** Logical operators operate on boolean data types, providing a boolean result. They are also used in decision making.

Operator	Operation
!	Logical Not
&&	Logical And
||	Logical Or

The Not (!) operator simply negates a boolean value. It is very useful for 'toggling' boolean variables.

Examples:

If a = true, then !a = false
If a = false, then !a = true

Review: Delivery

Create a Delivery application that prompts the user for the length, width, and height of a package, and then displays "Reject" if any dimension is greater than 10, and "Accept" if all dimensions are less than or equal to 10.

Unit 4 Worksheet A[1]

CODE	OUTPUT

`int a = 30;`
`if(a < 10)`
`  out.println("hello");`
`out.println("always happens");`
	always happens

`int a = 8;`
`if(a < 10)`
`  out.println("hello");`
`else if(a > 10)`
`  out.println("goodbye");`
	hello

`int b = 4;`
`if(b < 2)`
`  out.println("one");`
`else if(b < 4)`
`  out.println("two");`
`else if(b < 8)`
`  out.println("three");`
	three

`int c = 5, d = 8;`
`if(c < 10 && d > 10);`
`  out.println("hello");`
`out.println("goodbye");`
	goodbye

`int c = 5, d = 8;`
`if(c < 10 && d > 10)`
`  out.println("hello");`
`out.println("goodbye");`
	goodbye

`int e = 7, f = 3;`
`if(e > 5 || f > 10)`
`  out.println("hello");`
`else if(e > 2)`
`  out.println("goodbye");`
	hello

`int g = -6;`
`if(g < 10){`
`  if(g > 0){`
`  out.println("one");`
`  }`
`  else{`
`  out.println("two");`
`  }`
`}`
	two

`int g = 10;`
`if(g < 10){`
`  if(g > 0){`
`  out.println("one");`
`  }`
`  else if(g > 5){`
`  out.println("two");`
`  }`
`}`
`else if(g >= 10){`
`  if(g > 15){`
`  out.println("three");`
`  }`
`  else if(g < 15){`
`  out.println("four");`
`  }`
`}`
	four

[1] if else if worksheet

Lab 05A – DISCOUNT DETERMINER

Lab Goal : This lab was designed to teach you how to use if statements.

Lab Description : Determine the amount of discount a person should receive. If the bill is more than 2000, then the person should receive a 15% discount. If the bill is 2000 dollars or less, the person does not receive a discount of any kind.

Sample Data:

```
500
2500
4000
333.33
95874.2154
```

Files Needed ::
Discount.java
Lab05a.java

Sample Output :

```
Enter the original bill amount :: 500
Bill after discount :: 500.00

Enter the original bill amount :: 2500
Bill after discount :: 2125.00

Enter the original bill amount :: 4000
Bill after discount :: 3400.00

Enter the original bill amount :: 333.333
Bill after discount :: 333.33

Enter the original bill amount :: 95874.2154
Bill after discount :: 81493.08
```

```
//EXAMPLE IF CODE

int x = 90;
int y = 0;

if(x>50){
   y=22;
}
if(x<=50){
   y=44;
}
```

FORMATTING OUTPUT

```
double dec = 9.541724;

                  //printf is a void method
System.out.printf("%.3f\n",dec);                        //outs 9.542

                  //format is a String return method
                  //format is quite useful when writing toString() methods
System.out.println(String.format("%.3f",dec));         //outs 9.542
```

Lab 05D – CHARACTER ANALYZER

Lab Goal : This lab was designed to teach you how to use if statements.

Lab Description : Analyze a character to determine if the character is lowercase, uppercase, or a digit. You must know the ASCII values for 'a', 'A', and '0'.

Sample Data:

```
A
1
a
7
D
2
X
z
9
```

Files Needed ::
CharacterAnalyzer.java
Lab05d.java

Sample Output :

```
Enter a letter :: A
A is an uppercase character. ASCII == 65

Enter a letter :: 1
1 is a number. ASCII == 49

Enter a letter :: a
a is a lowercase character. ASCII == 97

Enter a letter :: 7
7 is a number. ASCII == 55

Enter a letter :: D
D is an uppercase character. ASCII == 68

Enter a letter :: 2
2 is a number. ASCII == 50

Enter a letter :: X
X is an uppercase character. ASCII == 88

Enter a letter :: z
z is a lowercase character. ASCII == 122

Enter a letter :: 9
9 is a number. ASCII == 57
```

```
//EXAMPLE NESTED IF CODE

int x = 90;
int y = 0;

if(x>50)
{
   if(y>20)
   {
      out.println("fun");
   }
}
```

Lab 07A – GRADE CHECK

Lab Goal : This lab was designed to teach you more about if statements, if else statements, and if else if else statements.

Lab Description : Take a number and calculate its letter grade.

A is >= 90
B is >= 80 and <90
C is >= 75 and <80
D is >= 70 and <75
F is <70

Files Needed ::
Grade.java
Lab07a.java

Sample Data :

```
78
92
31
82
77
73
55
65
```

Sample Output :

```
Enter a number grade :: 78
78 is a C

Enter a number grade :: 92
92 is a A

Enter a number grade :: 31
31 is a F

Enter a number grade :: 82
82 is a B

Enter a number grade :: 77
77 is a C

Enter a number grade :: 73
73 is a D

Enter a number grade :: 55
55 is a F

Enter a number grade :: 65
65 is a F
```

```
//EXAMPLE IF ELSE IF

int x=90;
int y=0;

if(x>50){
   y=1;
}
else if(x>40){
   y=2;
}
else if(x>30){
   y=3;
}
```

Lab 07C – DECODER

Lab Goal : This lab was designed to teach you more else if statements.

Lab Description : You are to decode each letter. The following explains how to decode each letter.

LowerCase becomes UpperCase
UpperCase becomes LowerCase
Digits 0-9 become A-J
Any other characters become #

Files Needed ::
Decoder.java
Lab07c.java

Sample Data :

```
a
A
b
0
T
*
H
T
```

Sample Output :

```
Enter a letter :: a
a decodes to A

Enter a letter :: A
A decodes to a

Enter a letter :: b
b decodes to B

Enter a letter :: 0
0 decodes to A

Enter a letter :: t
t decodes to T

Enter a letter :: *
* decodes to #

Enter a letter :: h
h decodes to H

Enter a letter :: T
T decodes to t
```

The Character class might prove useful.

ASCII values

A - 65
a - 97
0 - 48

Unit 4 Quiz 1

Check for Theory Reviews on Blackboard

Comparing Strings

Strings are *objects* and cannot be compared with the == sign. There are 2 methods to compare Strings:

equals ( ) Used to compare two strings with the method to see if they are exactly the same, this includes any blanks or spaces within the string.
Example: *(check to see if the user entered a dog name of Snoopy)*
```
if (dogname.equals("Snoopy");
System.out.println ( "The user entered Snoopy.);
```

compareTo() Compares strings to determine alphabetic location. Returns a zero if the two strings are equal, a negative if the first string is alphabetically before the compared string, and a positive if the first string is alphabetically after the compared string.
Example:
```
String subject = "mathematics";
boolean answer;
answer = subject. compareTo ("biology"); // returns is a positive
answer = subject. compareTo("philosophy"); // answer is negative
answer = subject.compareTo (" mathematics"); //answer is zero
```

Lab 06B – STRING EQUALITY

Lab Goal : This lab was designed to teach you more about the String class and how if statements are used.

Lab Description : Compare two strings to see if each of the two strings contains the same letters in the same order.

Useful methods :: `.equals()` `compareTo()`

Sample Data :
```
hello goodbye
one two
three four
TCEA UIL
State Champions
ABC ABC
ABC CBA
Same Same
```

Files Needed ::
```
StringEquality.java
Lab06b.java
```

Sample Output :
```
hello does not have the same letters as goodbye

one does not have the same letters as two

three does not have the same letters as four

TCEA does not have the same letters as UIL

State does not have the same letters as Champions

ABC has the same letters as ABC

ABC does not have the same letters as CBA

Same has the same letters as Same
```

Unit 4B Worksheet 1[2]

DIRECTIONS : Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
String one = "abcdefghijklm";
String two = "01234567890";
String three = "01 23 45 67 89 0";
```

```
System.out.print( one.length());                  // LINE 1
System.out.print( two.length());                  // LINE 2
System.out.print( three.length());                // LINE 3
System.out.print( one.charAt(0) );                // LINE 4
System.out.print( one.charAt(1));                 // LINE 5
System.out.print(one.charAt(one.length()-1));     // LINE 6
System.out.print( one.charAt(9) );                // LINE 7
System.out.print( one.substring(0,4) );           // LINE 8
System.out.print( one.substring(5) );             // LINE 9
System.out.print( one.substring(9) );             // LINE 10
System.out.print( one.substring(2,7));            // LINE 11
System.out.print( one.indexOf("abc") );           // LINE 12
System.out.print( one.indexOf("e") );             // LINE 13
System.out.print( one.indexOf("hij") );           // LINE 14
System.out.print( two.indexOf("56"));             // LINE 15
System.out.print( two.indexOf("24"));             // LINE 16
System.out.print( one.indexOf('c'));              // LINE 17
System.out.print( two.indexOf('r'));              // LINE 18
System.out.print( two.indexOf('0'));              // LINE 19
System.out.print( three.indexOf("45"));           // LINE 20
```

1. 13
2. 11
3. 16
4. a
5. b
6. m
7. j
8. abcd
9. fghijklm
10. jklm
11. cdefg
12. 0
13. 4
14. 7
15. 5
16. -1
17. 2
18. -1
19. 0
20. 6

[2] String Worksheet 1

Lab 06E – STRING LENGTH

Lab Goal : This lab was designed to teach you more about the String class and how if statements are used.

Lab Description : Compare two Strings to see if each of the two Strings contains the same number of letters.

Useful methods :: **`.length()`**

Sample Data :

```
hello goodbye
one two
three four
TCEA UIL
State Champions
ABC DEF
four five
whoot what
```

Files Needed ::
`StringLengthCheck.java`
`Lab06e.java`

Sample Output :

```
hello does not have the same # of letters as goodbye

one has the same # of letters as two

three does not have the same # of letters as four

TCEA does not have the same # of letters as UIL

State does not have the same # of letters as Champions

ABC has the same # of letters as DEF

four has the same # of letters as five

whoot does not have the same # of letters as what
```

Lab 07D – MOUSE BUTTON TESTER

Lab Goal : This lab was designed to teach you more else if statements.

Lab Description : Write a program that will test to see which mouse button was pressed. If the left mouse button is pressed, draw a rectangle at that location. If the right mouse button is pressed, draw a circle at that location. Feel free to change the colors, sizes, and shapes however you want.

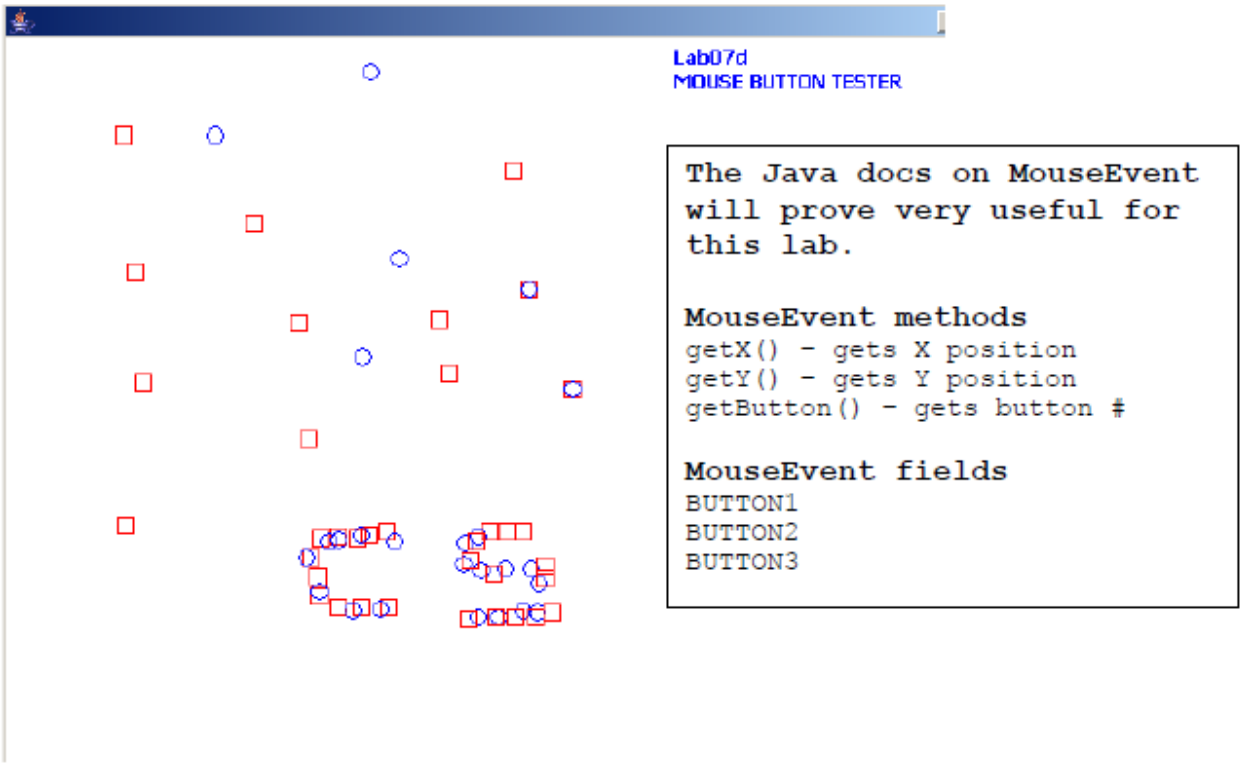
Sample Data :

```
lots of mouse clicks
```

Files Needed ::

```
MouseButtonTester.java
Lab07d.java
```

Sample Output :



Unit 4 Quiz 2

Check for Theory Reviews on Blackboard

The switch Statement

This is a conditional control structure that uses the result of an expression to determine which statements to execute.

It takes the form:

```
switch (<integer expression>) {
  case x:
      <statement>;
      break;
  ...
  default:
      <statements>;
      break;
}
```

Notes:

- The expression must evaluate to an integer.
- The break statement is necessary to move control to the next statement after the switch statement. Without the break, the following case statements will execute.
- The default statement is optional, and executed when none of the case statements execute.

LOOK UP

Look at the link below for examples using integers and character integers.
http://mathbits.com/MathBits/Java/Conditionals/switchtemplates.htm

The Math Class

A last set of functions we need are mathematical functions (yes, programming involves math!) Each of the Java math functions comes from the Java Math class. This means is that each function name must be preceded by **Math.** to work properly. Some of these functions and the returned values are:

The Math Class	
Math Function	**Value Returned**
Math.abs	Returns the absolute value of a specified number
Math.acos, etc.	Returns a double value containing the angle whose cosine is the specified number, etc.
Math.max	Returns the larger of two numbers
Math.min	Returns the smaller of two numbers
Math.PI	A constant that specifies the ratio of the circumference of a circle to its diameter
Math.pow	Returns the result of raising the first argument to the power of the second argument – an exponentiation.
Math.round	Returns the number nearest the specified value
Math.sign	Returns an Integer value indicating the sign of a number
Math.sqrt	Returns a double value specifying the square root of a number

Examples:

Math.abs(-5.4) returns the absolute value of –5.4 (returns 5.4)
Math.max(7, 10) returns the larger of the two numbers (returns 10)
Math.pow(4, 3) returns 4 raised to the 3rd power
Math.sqrt(4.5) returns the square root of 4.5

Unit 4C Worksheet A[3]

Show the output of each block of code below.

1.
```
int a=14;
switch(a%4){
   case 2 : out.println("rem 2");
   case 3 : out.println("rem 3");
   case 4 : out.println("rem 4");
}
```

OUTPUT
rem 2
rem 3
rem 4

2.
```
int b=14;
switch(b%2){
   case 2 : out.println("rem 2");
   case 3 : out.println("rem 3"); break;
   case 4 : out.println("rem 4");
   default : out.println("rem 0"); break;
}
```

OUTPUT
rem 0

3.
```
int c=3, d=14;
switch(d/c){
   case 1 :
   case 2 :
   case 3 : out.println("1,2,or3"); break;
   case 4 : out.println("4"); break;
   case 5 : out.println("5"); break;
}
```

OUTPUT
4

4.
```
char e = 'B';
switch(e){
   case 'A' : out.println("ascii " + (int)e);
   case 'B' : out.println("ascii " + (int)e);
   case 'C' : out.println("ascii " + (int)e);
   case 'D' : out.println("ascii " + (int)e);
}
```

OUTPUT
ascii 66
ascii 66
ascii 66

5.
```
int f=3, g=14;
switch(f)
{
   case 3 : out.println("hello" + g%7);
   case 4 : out.println("goodbye" + g%5);
}
```

OUTPUT
hello0
goodbye4

6.
```
char h = 'a';
switch(h){
   case 97 : out.println("a's ascii==97"); break;
   case 98 : out.println("b's ascii==98"); break;
}
```

OUTPUT
a's ascii==97

[3] Switch Case Worksheet

Lab 07E – MORSE CODE

Lab Goal : This lab was designed to teach you more about switch case statements.

Lab Description : Convert a given letter to morse code.

A	.-	U	..-
B	-...	V	...-
C	-.-.	W	.--
D	-..	X	-..-
E	.	Y	-.--
F	..-.	Z	--..
G	--.	0	-----
H	....	1	.----
I	..	2	..---
J	.---	3	...--
K	-.-	4	....-
L	.-..	5	.....
M	--	6	-....
N	-.	7	--...
O	---	8	---..
P	.--.	9	----.
Q	--.-		
R	.-.		
S	...		
T	-		

Files Needed ::
MorseCode.java
Lab07e.java

Sample Data :

```
A
B
3
Z
8
F
0
```

Sample Output :

```
. -
A is . - in morse!

- . . .
B is - . . . in morse!

. . . - -
3 is . . . - - in morse!

- - . .
Z is - - . . in morse!

- - - . .
8 is - - - . . in morse!

. . - .
F is . . - . in morse!

- - - - -
0 is - - - - - in morse!
```

```
//EXAMPLE SWITCH CASE

int x=90,y=0;
switch(x)
{
   case 70 : y=5; break;
   case 80 : y=10; break;
   case 90 : y=15; break;
   case 100 : y=20; break;
}
```

Unit 4C Worksheet B[4]

DIRECTIONS : Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
double z = 123.456;      long x = 7;      int a = 5, b = 2;      char var = 'H';
```

`System.out.print( 3 + 3 * 3 );`	`// LINE 1`	12
`System.out.print( a * (a % b) );`	`// LINE 2`	5
`System.out.print( b / a );`	`// LINE 3`	0
`System.out.print( 'A'+ 5 * b);`	`// LINE 4`	75
`System.out.print( 1 % 5 );`	`// LINE 5`	1
`System.out.print( a % b );`	`// LINE 6`	1
`System.out.print( b % a );`	`// LINE 7`	2
`System.out.print( 'A'+ 5 );`	`// LINE 8`	70
`System.out.print( (double)( a / b) );`	`// LINE 9`	2.0
`System.out.print( (double)a / b );`	`// LINE 10`	2.5
`System.out.print( var + 5);`	`// LINE 11`	77
`System.out.print((char)(var + 5));`	`// LINE 12`	M
`a=var+2;` `System.out.println( a );`	`// LINE 13`	74
`z=var+5;` `System.out.println( z);`	`// LINE 14`	77.0
`var='A'+4;` `System.out.println( var);`	`// LINE 15`	E
`a*=2+5;` `System.out.println( a );`	`// LINE 16`	518
`var=(char)(z-25);` `System.out.println( var );`	`// LINE 17`	4
`a++;` `System.out.print( a );`	`// LINE 18`	519
`b--;` `System.out.print( b );`	`// LINE 19`	1
`++x;` `System.out.print( x  );`	`// LINE 20`	8
`System.out.print( --var );`	`// LINE 21`	3

[4] math/calc worksheet 1

Unit 4C Worksheet C[5]

DIRECTIONS : Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
System.out.print( 1 % 5 );                    // LINE 1
System.out.print( 5 % 2 );                    // LINE 2
System.out.print( 7.2 % 3 );                  // LINE 3
System.out.print( 'A' + 9 );                  // LINE 4
System.out.print( 9.3 % 3 );                  // LINE 5
System.out.print( 'c' + 5 );                  // LINE 6
System.out.print( Math.pow(3,3) );            // LINE 7
System.out.print( Math.ceil(3.4) );           // LINE 8
System.out.print( Math.floor(4.6) );          // LINE 9
System.out.print( Math.sqrt(12) );            // LINE 10
System.out.print( Math.round(12.34) );        // LINE 11
System.out.print( Math.round(12.56) );        // LINE 12
System.out.print( Math.max(12,34));           // LINE 13
System.out.print( Math.min(12,34));           // LINE 14
System.out.print( Math.max(12.3,45.6));       // LINE 15
System.out.print( Math.min(12.3,45.6));       // LINE 16
System.out.print( Math.abs(-213));            // LINE 17
System.out.print( Math.abs(213));             // LINE 18
System.out.print( Math.ceil(Math.sqrt(17)));  // LINE 19
System.out.print( Math.random() );            // LINE 20
System.out.print( Math.random()*10 );         // LINE 21
System.out.print( Math.random()*20+10 );      // LINE 22
```

1 _____

1 _____

1.2000000000000002 _____

74 _____

0.3 _____

104 _____

27 _____

4.0 _____

4.0 _____

3.4641016151377544 _____

12 _____

13 _____

34 _____

12 _____

45.6 _____

12.3 _____

213 _____

213 _____

5.0 _____

MIN 0.0 MAX 0.99

MIN 1.0 MAX 0.99

MIN 10.0 MAX 29.99

Unit 4 Quiz 3

Check for Theory Reviews on Blackboard

[5] Math Class Worksheet

Lab x – ROCK – PAPER - SCISSORS

Lab Goal : This lab was designed to teach you how to design and use classes, instantiate objects, use GUI Components, use if statements, and multiple selection statements (switch case / if else if).

Lab Description : Write a program in Java to allow the play of the game Rock-Paper-Scissors. Each time a button is pressed, a game of Rock-Paper-Scissors will be played. You may use a variant of this game, but you must have at least 3 different symbols in the game. You will receive extra credit if you use more than 3 symbols.

Part One :: Use the `Player.java` file to create the player. Complete the `Player` constructors, set methods, get methods, and the toString. Use `PlayerRunner.java` to test your `Player` class.

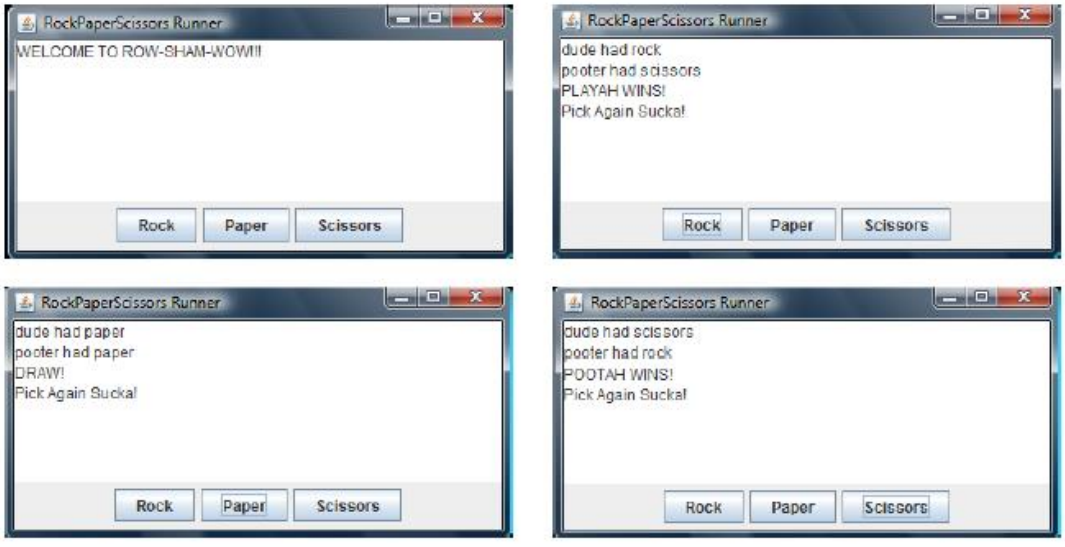
Part Two :: Use the `Computer.java` file to create the computer player. Complete the `Computer` constructors, methods, and the toString. Use `ComputerRunner.java` to test your `Computer` class.

Part Three :: Use the `RockPaperScissorsRunner.java` to complete the project. Each time a button is pressed, set the choice for the player based on that button. Then, compare the player choice to the computer choice using the computer's didIWin() method. Display the results of the game. Add in scoring to show how many games each player has won. Use fonts to make the text appear in color and in a different font style.

BONUS :: Add in sound to this lab using MP3s (Javazoom - JLayer).
Change the background color of the JTextArea and the Font style and color (JTextArea).

Files Needed ::
`Player.java`
`PlayerRunner.java`
`Computer.java`
`ComputerRunner.java`
`RockPaperScissorsRunner.java`

Sample Output :



You may be asked to complete more labs.

Test on Unit 4

Check for Theory Review on Blackboard and study from your worksheets.