***Weeks 4 and 5***

***(notes for chapters 4 and 5)***



**if Statements**

if (boolean condition)

statement 1;

else

 statement 2;

if (boolean condition)

{multiple statements}

else

 {multiple statements}

if (boolean condition 1)

 if (boolean condition 2)

 if (boolean condition 3)

statement 1;

else statement 2;

 else statement 3;

else statement 4;

Relational Operators: = = ! = > < > = < =

Logical Operators: && | |

Write code fragment to assign into biggest the largest value of a, b, c.



How to Assign the largest value of a, b, and c into the variable biggest using the conditional expression operator

biggest = (a>b) ? ((a>c) ? a : c) : ((b>c) ? b : c);

**while Statement**

while (condition)

{

 body of the loop

}

// Demonstrate the while loop.

class While {

 public static void main( String args[ ]) {

 int n = 10;

 while (n > 0) {

 System.out.println(“tick ” + n);

 n--;

 }

 System.out.println(“Lift Off!”);

 }

}

**Output:**

 tick 10

 tick 9

 tick 8

 tick 7

 tick 6

 tick 5

 tick 4

 tick 3

 tick 2

 tick 1

 Lift off!

// The target of a loop can be empty.

class NoBody {

 public static void main(String args[ ]) {

 int i = 100, j = 200;

 // find midpoint between i and j

 while (++i < --j) ; // no body in this loop

 System.out.println(“Midpoint is ” + i);

**Output:**

 Midpoint is 150

**do-while Statement**

do {

 // body of loop

} while (*condition*);

// Demonstrate the do-while loop.

Class DoWhile {

 public static void main (String args[]) {

 int n=10;

 do {

 System.out.println(“tick ” + n);

 n--;

 } while (n > 0);

 }

}

A 2nd Technique

 do {

 System.out.println(“tick” + n--);

 } while (n>0);**for Statement**

4

2

1

for (initialization; condition; iteration) {

3

 // body of loop

}

for loops begin with initialization (1)

they then circle around, evaluating the condition (2)

 executing the body of the loop (3)

 executing the iteration section (4)

When the condition becomes false, the loop is exited.

**Examples of the for loop**

// Demonstrate the for loop

class ForTick {

 public static void main(String args[ ]) {

 int n;

 for (n=10; n>0; n--)

 System.out.println(“tick ” + n);

 }

}

A 2nd Method

 for (int n=10; n>0; n--)

 System.out.println(“tick ” + n);

A 3rd Method

for (int n=10; n>0; System.out.println(“tick ” + n--));**Nested for loops**

// Loops may be nested

class Nested {

 public static void main(String args[]) {

 int i, j;

 for (i=0; i<10; i++) {

 for (j=i; j<10; j++)

 System.out.print(“.”);

 System.out.println();

 }

 }

}

**Output Produced:**

……….

………

……..

…….

……

…..

….

…

..

.Using **break**

The **break** statement will force the immediate termination of a loop, bypassing the conditional expression and any remaining code in the body of the loop. When a **break** statement is encountered inside a loop, the loop is terminated and program control resumes at the next statement following the loop. Here is a simple example:

// Using break to exit a loop.

Class BreakLoop {

 public static void main(String args[]) {

 for (int i=0; i<100; i++) {

 if (i = = 10)  **break**; // terminate if i is 10

 System.out.println( “i: ” + i);

 }

 System.out.println(“Loop Complete.”);

 }

}

**Output Produced:**

i: 0

i: 1

i: 2

i: 3

i: 4

i: 5

i: 6

i: 7

i: 8

i: 9

Loop Complete.

// Using break to exit from nested loops

class BreakLoop4 {

 public static void main(String args[]) {

 outer: for (int x=0; x<3; x++) {

 System.out.print(“Pass ” + x + “: ”);

 for (int y=0; y<100; y++) {

 if (y = = 10) break outer; // exit both loops

 System.out.print(y + “ ”);

 }

 System.out.println(“This will not print”);

 }

 System.out.println(“Loops Complete.”);

 }

}

**Output Produced:**

Pass 0: 0 1 2 3 4 5 6 7 8 9 Loops Complete.Using **continue**

Sometimes it is useful to force an early iteration of a loop. That is, you might want to continue running the loop, but stop processing the remainder of the code in its body for this particular iteration. This is, in effect, a goto just past the body of the loop, to the loop’s end. The **continue** statement performs such an action. In **while** and **do-while** loops, a **continue** statement causes control to be transferred directly to the conditional expression that controls the loop. In a **for** loop, control goes first to the iteration portion of the for statement and then to the conditional expression. For all three loops, any intermediate code is bypassed.

// Demonstrate continue

class Continue {

 public static void main(String args[]) {

 for (int x=0; x<10; x++) {

 System.out.print (x + “ ”);

 if (x % 2 = = 0) **continue**;

 System.out.println (“”);

 }

 }

}

**Output Produced:**

0 1

2 3

4 5

6 7

8 9

**Switch Statement**

switch (expression) {

case *value1*:

// statement sequence 1

break;

case *value2*:

// statement sequence 2

break;

.

.

.

case *valueN*:

// statement sequence N

default:

// default statement sequence

}

// A simple example of the switch

class SampleSwitch {

 public static void main(String args[]) {

 for (int x=0; x<5; x++)

 switch(x) {

 case 0:

 System.out.println(“x is zero.”);

 break;

 case 1:

 System.out.println(“x is one.”);

 break;

 case 2:

 System.out.println(“x is two.”);

 break;

 default:

 System.out.println(“x is greater than two.”);

 }

 }

**Output Produced:**

x is zero.

x is one.

x is two.

x is greater than two.

x is greater than two.class Switch {

 public static void main(String args[]) {

 int month = 4;

 String season;

 switch (month) {

 case 12:

 case 1:

 case 2:

 season = “Winter”;

 break;

 case 3:

 case 4:

 case 5:

 season = “Spring”;

 break;

 case 6:

 case 7:

 case 8:

 season = “Summer”;

 break;

 case 9:

 case 10:

 case 11:

 season = “Autumn”;

 break;

 default:

 season = “Bogus Month”;

 }

 System.out.println(“April is in the ” + season + “.”);

 }

}

Notes: “A” is not equal to ‘A’

 case (3<= month<=5) will not work

 case (3<=month) && (month<=5) will not work

**Programming Assignment 5**

A mail order house sells five different products whose retail prices are:

product 1 - $2.98,

product 2 - $4.50,

product 3 - $9.98,

product 4 - $4.49,

product 5 - $6.87.

Write an applet that accepts pairs of numbers:

 1. Product number

 2. Quantity sold

Each pair of numbers represents a purchase. Your program must use a **switch** statement to determine the retail price for a purchase. Your program should calculate and display the total retail value of the purchase, and a grand total of all purchases.