Reading:
1. Deidel & Deidel, Chapter 2 and Chapter 4

Objectives:
1. Modifying Java programs
2. Basic Java mathematical functions
3. Decision making skills and practices
4. Introduction to control structures
5. Exploring pseudocode
6. If/else statements

Concepts:
1. Discuss the use of arithmetic operators
2. Use decision making models to find solutions to problems
3. Discuss control structures
4. Examine the use of pseudocode in finding solutions to problems
5. Discuss if/else statements
6. Explore the while statement
1. Discuss Memory Concepts
   • Variable names and storage locations
   • Variable names and conversions

2. Modifying Java programs
   • Comment cues
   • Understanding syntax

2. Java Arithmetic
   • Adding integers
   • Arithmetic expressions in straight line form
   • Arithmetic operators
   • Rules of operator precedence
   • Show examples

3. Decision Making Skills
   • Examining problem statements
   • Case Study 2.9

4. Control Structure Introduction
   • Algorithms
   • Pseudocode
   • Java keywords

5. The if statement
   • If pseudocode
   • Allows for output or computation only on a true statements
   • Show examples

6. The if/else structure
   • Allows for different actions if statement is false
   • Allows for more than one condition (if/else if/else)
   • Show examples

7. The while repetition structure
   • While pseudocode
   • Allows for action until a parameter is reached
   • Show examples
Java Keywords:

abstract    float    short  
catch       instanceof static 
do         new       throw
final        public    void 
implements  byte       const 
long         continue    super    
private      extends    throws 
boolean      for        volatile 
char         int        goto    
double       null       switch 
finally      return     transient 
import       case       while      
native       default    synchronized 
protected    false      true    
break        if         this    
class        interface  try 
else         package    

The if Selection Structure:

If the number of dogs in the house is greater than or equal to 3
print “Too many dogs in the house!”

Relational Operators in Java:

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

Pseudocode:
if numberofdogs >= 3  
print “Too many dogs in the house!”

Java code:
if( numdogs >= 3) 
    System.out.println( “Too many dogs in the house!” );
The **if/else** Selection Structure:

If the number of dogs in the house is greater than or equal to 3
print “Too many dogs in the house!” else print “The number of dogs in the house is just right”

**Pseudocode:**

if numberofdogs \( \geq 3 \)
print “Too many dogs in the house!”
else print “The number of dogs in the house is just right”

**Java code:**

```java
if (numdogs \( \geq 3 \))
    System.out.println(“Too many dogs in the house!”);
else
    System.out.println(“The number of dogs in the house is just right”);
```

The **if/else if/else** Selection Structure:

If the number of dogs in the house is greater than 3 print “Too many dogs in the house!” else if the number of dogs in the house is equal to 3, print “The number of dogs in the house is just right” else print “There are too few dogs in the house!”

**Pseudocode:**

if numberofdogs > 3
print “Too many dogs in the house!”
else if numberofdogs = 3
print “The number of dogs in the house is just right”
else print “There are too few dogs in the house!”

**Java code:**

```java
if (numdogs > 3)
    System.out.println(“Too many dogs in the house!”);
else if (numdogs == 3)
    System.out.println(“The number of dogs in the house is just right”);
else
    System.out.println(“There are too few dogs in the house!”);
```

The **while** structure:

This repetition structure allows the programmer to specify that an action is to be repeated while a condition remains true

While the number of people on the bus is less than 20 admit another person on the bus and increment the number of people on the bus by one

**Pseudocode:**

if numberOfPeopleOnBus > 20
print “Passenger admitted!”
numberofpeopleonbus = numberofpeopleonbus + 1

Java code:

int numpass = 0;

while ( numpass < 20 )
    System.out.println( “Passenger Admitted” );
    numpass = numpass + 1;

Increment and Decrement Operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>++</td>
<td>preincrement</td>
</tr>
<tr>
<td>++</td>
<td>postincrement</td>
</tr>
<tr>
<td>--</td>
<td>predecrement</td>
</tr>
<tr>
<td>--</td>
<td>postdecrement</td>
</tr>
</tbody>
</table>

a = a + 1;
same as a++;
Objectives:
1. To explore pseudocode
2. To explore if/else structures
3. Debugging and Correcting Code

Students Activities:
1. Become familiar with pseudocode as a tool to structure programs
2. Complete exercises that work with if/else statements

//Lab 2
//If/else exercises

4.21
a) x = 9  x=11
   y=11  y=9
   Output:   Output:
             ****
             none
             $$$$$$
b) x = 9  x=11
   y=11  y=9
   Output:   Output:
             *****
             ######
             $$$$$$

4.22
a) if ( y == 8 ){
    if ( x == 5 )
     System.out.println( "@@@" );
    else
     System.out.println( "###" );
    System.out.println( "$$$" );
    System.out.println( "&&&" );
  }

b) if ( y == 8 ){
    if ( x == 5 )
     System.out.println( "@@@" );
    else{
     System.out.println( "###" );
     System.out.println( "$$$" );
     System.out.println( "&&&" );
  }
c) \[ \text{if ( y == 8 )} \{ \text{if ( x == 5 )} \text{System.out.println("@@@\@")}; \text{else} \text{System.out.println("####")}; \text{System.out.println("$$\$$")}; \} \text{System.out.println("&\&&\&")}; \]


d) \[ \text{if ( y == 8 )} \{ \text{if ( x == 5 )} \text{System.out.println("@@@\@")}; \text{else} \text{System.out.println("####")}; \text{System.out.println("$$\$$")}; \text{System.out.println("&\&&\&")}; \}

\]

4.26
\[
\text{count = 1} \\
\text{while count <= 8} \{ \text{if (count % 2 == 1)} \text{print "* * * * * * * "} \text{else print "* * * * * * "} \text{count = count + 1; } \}
\]