Raymond Walters College

| OATN 261-Beginning Java Gregory D. KummerRm 328 Muntz Hall - Mon 6:20-9:00Visual Basic to Java - Spring 2002 |
| --- |

| [Gregory D. Kummer](http://192.0.0.1/ws1/kummerg/default.html), Instructor. E-mail: [kummerg@mason.k12.oh.us](mailto:kummerg@mason.k12.oh.us)  Office phone: 513-794-9140 Office hours: 5:00-6:20 M |
| --- |

### Variable Usage

Variables are the short-term memory of computer programs. This is how your program remembers information while it is running.

| Figure 1: Variable Usage | |
| --- | --- |
| Java | Visual Basic |
| int i;// ints can store integer numbers double d;// doubles can have decimal places String s;// strings can hold characters  i = 10; d = 3.14; s = "Java"; | Dim d As Double ' A double can hold decimal places Dim i As Integer ' An integer can hold integers Dim s As String ' A string can hold characters  d = 3.14 i = 10 s = "Visual Basic" |

### If Statements

Nearly every programming language has a way to make decisions. If statements are the usual means.

| Operators Used with IF Statements | | |
| --- | --- | --- |
| Name | VB | Java |
| Equal | = | == |
| Not Equal | <> | != |
| Greater Than | > | > |
| Less Than | < | < |
| Greater Than or Equal | >= | >= |
| Less Than or Equal | < | < |

| Figure 2: Simple If Statements | |
| --- | --- |
| Java | Visual Basic |
| int a,b;  a=1; b=2;  if(a==b) {  System.out.println("A and B are equal"); } else {  System.out.println("A and B are not equal"); } | Dim a as integer Dim b as integer  a=1 b=2  if a=b then  MsgBox "A and B are equal" else  MsgBox "A and B are not equal" end if |

### Compound If Statements

Compound if statements make decisions based on several things. This is done by using Boolean operators such as AND and OR.

| Figure 3: Compound Simple If Statements | |
| --- | --- |
| Java | Visual Basic |
| int a,b;  a=1; b=2;  if( (a==b) || (a==2) ) {  System.out.println("A and B are equal, or a equals 2"); } else {  System.out.println("A and B are not equal or a is not 2"); }  if( (a==b) && (a==2) ) {  System.out.println("A and B are equal, and a equals 2"); } else {  System.out.println("A and B are not 2"); } | Dim a As Integer Dim b As Integer  a = 1 b = 2  If (a = b) Or (a = 2) Then  MsgBox "A and B are equal, or a equals 2" Else  MsgBox "A and B are not equal or a is not 2" End If  If (a = b) And (a = 2) Then  MsgBox "A and B are equal, and a equals 2" Else  MsgBox "A and B are not 2" End If |

### While Loops

The ability to execute a section of code over and over is very important to a programming language. This is called a loop. Java has three types of loops: the while, do/while and for.

| Figure 4: While Loops | |
| --- | --- |
| Java | Visual Basic |
| int a;  a=1;  while( a<=10 ) {  System.out.println("Counting: " + a + "..." );  a = a + 1;// could also use the shorthand a++; } | Dim a As Integer   a = 1  While (a <= 10)  MsgBox "Counting: " & a & "..."  a = a + 1 ' could also use the shorthand a++; Wend |

### Do/While Loops

A do while loop is very much like a while loop, except that the decision to execute is made at the end of the loop. This means that a do/while loop must execute at least once.

| Figure 5: Do/While Loops | |
| --- | --- |
| Java | Visual Basic |
| int a;  a=1;  do {  System.out.println("Counting: " + a + "..." );  a = a + 1;// could also use the shorthand a++; } while( a<=10 ); | a = 1  Do  MsgBox "Counting: " & a & "..."  a = a + 1 ' could also use the shorthand a++; Loop While (a <= 10) |

### For Loops

A for loop is similar to a while loop. Except that much more is built into the loop command.

| Figure 6: For Loops | |
| --- | --- |
| Java | Visual Basic |
| int a;  // count from 1 to 10  for(a=1;a<=10;a++) {  System.out.println("Counting: " + a + "..." ); }  // count from 10 to 1  for(a=10;a>=1;a--) {  System.out.println("Counting: " + a + "..." ); }  // count from 2 to 10 by twos  for(a=2;a<=10;a+=2) {  System.out.println("Counting: " + a + "..." ); } | a = 1  Dim a As Integer  ' count from 1 to 10  For a = 1 To 10  MsgBox "Counting:" & a & "..." Next a  ' count from 10 to 1  For a = 10 To 1 Step -1  MsgBox "Counting:" & a & "..." Next a  ' count from 2 to 10 by twos  For a = 2 To 10 Step 2  MsgBox "Counting:" & a & "..." Next a |

### Methods:Functions and Subs

Functions and subs allow you to create a block of code that can be executed from many different locations. Java makes no distinction between functions and subs. They are all called methods.

| Figure 7: Functions and Subs | |
| --- | --- |
| Java | Visual Basic |
| class HelloWorld {  public static void ExampleSub()  {  System.out.println("Calling the sub");  }   public static int ExampleFunction()  {  System.out.println("Calling the function");  return 5;  }   public static int AddIt(int x,int y)  {  int result;   System.out.println("Calling AddIt");  result = x + y;   return result;  }  public static void main(String args[])  {  ExampleSub();  ExampleFunction();  int t = AddIt(4,3);  System.out.println("Output: " + t);  } } | Private Sub Form\_Load()  Sub ExampleSub()  MsgBox "Calling the sub" End Sub  Sub ExampleFunction()  MsgBox "Calling the function" End Sub  Function AddIt(x As Integer, y As Integer) As Integer  Dim result As Integer   MsgBox "Calling AddIt"  result = x + y  AddIt = result End Function  Sub main()  ExampleSub  ExampleFunction  Dim t As Integer  t = AddIt(4, 3)  MsgBox "Output: " & t End Sub |

SYLLABUS  |  LESSON 1  | LESSON 2  | LESSON 3 |  LESSON 4  |  LESSON 5  |  LESSON 6  |  LESSON 7   |  LESSON 8