FIT 100 Misinformation Exhibition

A brief celebration of our Web sites of Misinformation:

- Royal Tiger by Shawn Humphrey
  http://students.washington.edu/shawnph/FIT100/FIT100Projects/Project1/
- Jackalope by Christie Leff
  http://students.washington.edu/cleff/jackalope.htm
- Hail by Sokchea Kahnn
  http://students.washington.edu/skhann/Project1.html
- Mt. Rainer Resort by Mickey Pierce
  http://students.washington.edu/mep/FIT100/Project1b.htm

FIT 100 Brain Warm-up: What Value Does Zip Have?

Take out a piece of scratch paper. See if you can answer the questions below.

```
Dim zip As Integer
zip = 0
zip = zip + 1
zip = zip + 1
zip = zip + 1
```

Questions:

1. What value does the variable *zip* contain at the end of this code?
2. What is this code doing?
3. What would be a better variable name for *zip*?

FIT 100 Expressions

- CONCEPT: Expressions are formulae made from variables and operators, e.g. calculator operations:
  +, -, *, /, ^
  - `weeks = days / 7` *divide value of days by 7*
  - `grossPay = hours * rate` *multiply the two values*
- Fundamental rule of assignment
  The expression is evaluated before the assignment is made
  - `score = score + 3`

Computing is NOT *algebra*: Though = is used in assignment statements, it means "becomes" whereas in algebra it means equality. So, `score = score + 3` is essential to computing, but meaningless in algebra

Conditionals

FIT 100

Computers can be programmed to "make decisions" – that is, to choose which one path to follow among many alternatives. Conditionals are the programming construct that implements this concept.

FIT 100 General Idea of a Conditional

- CONCEPT: Computer programs execute all statements in the program in order unless the program is instructed to *only* execute *certain* statements under *certain* conditions

- For example:

```
If (something is true) Then
        [do this part of the program]
End If
```


FIT 100 Operators

- CONCEPT: Operators are used to *combine* expressions (logical operators) or to *compare* expressions (relational operators)

- Most programming languages have more logical operators than a pocket calculator
  - Operators like + taking 2 operands are called *binary*: `a + b`
  - Operators like - taking 1 operand are called *unary*: `- a`

- A very useful logical operator is *concatenate*, & in VB6, which connects two strings together:
  - `plural = "dog" & "s"`

FIT 100 Operators

- CONCEPT: Relational operators are often used in conditional statements to create expressions that evaluate to either “true” or “false”
- The relational operators in VB6 are:
  - a < b less than a > b greater than
  - a <= b less than or equal to a >= b greater than or equal
  - a = b equal to a <> b not equal

FIT 100 Basic Conditional

- Use conditionals to test to see if a condition holds:
  - ```
    If temp < 32 Then
      state = "frozen"
      form = "ice"
    End If
    ```
- General form of basic conditional:

```
If <T/F expression> Then
  <statement list>
End If
```

- What this means:
  - First, the <*T/F expression*> is *evaluated*
  - If the outcome is true, then the statements that follow are performed
  - If the outcome is false, then the statements that follow are skipped
```

FIT 100 General Conditional Statement

- Concept: When one set of statements must be performed for the true condition and a different set of statements are needed for the false false, use the If-Then-Else statement
- General form

```
If <T/F expression> Then
   <statement list>
Else
   <statement list>
End If
```

```
If sideUp = sideCalled Then
   coinTossWinner = hostTeam
   firstHalfOffense = hostTeam
   secondHalfOffense = visitorTeam
Else
   coinTossWinner = visitorTeam
   firstHalfOffense = visitorTeam
   secondHalfOffene = hostTeam
End If
```


FIT 100 "Nested" If-Then-Else

- CONCEPT: An advantage of the general conditional is that statement lists can contain other conditionals

```
If flip1 = guess1 Then
   If flip2 = guess2 Then
      score = "win win"
   Else
      score = "win lose"
   End If
Else
   If flip2 = guess2 Then
      score = "lose win"
   Else
      score = "lose lose"
   End If
End If
```

FIT 100 Let's Try Our Hand at VB6 Program

- Let's write a program that takes an integer as input and outputs whether or not the integer is a positive number.
  - How should we get the user's input?
  - How do we tell is the input is a positive or negative number?
  - What should we do with an input of "0"?
  - How should we output the "positive" or "negative" evaluation to the user?
  - How do we get started?