A Little Review:
What is the Value of wicked, dude?

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

```vbnet
Dim wicked As Integer
Dim dude As Integer

dude = 5
wicked = 2

dude = wicked * 5  ' * means multiply

dude = dude + 3
wicked = dude + 1
```

Questions:
1. What values do dude and wicked contain at the end of this code?

Expressions

CONCEPT: Expressions are a means of performing the actual computation in a program. They are formulae made from variables and operators, e.g. calculator operations:

- `weeks = days / 7` divide value of days by 7
- `totalAfterTax = totalPrice * 1.087` multiply the two values

The Fundamental Rules of Assignment:
- The general form of an assignment statement is `<variable name> <assignment symbol> <expression>`
- The flow of information is always right to left
- The expression is evaluated before the assignment is made
  - `score = score + 3`

Operators:

CONCEPT: Operators are used to combine expressions. They are used in combination with values, or variables that contain values – both called operands when using operators to complete the expression formulae.

Operators in programming languages aren’t always what you find on 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 operator is concatenate, `&` in VB6, which connects two strings or variables together:

- `plural = "dog" & "s"`

When A Decision Must Be Made:

Conditionals

Computers can be programmed to make decisions – that is, to choose one path to follow from many alternatives. Conditionals are the programming tool that implements this concept.
The Reason to Have Conditionals:

- **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

Logical (Relational) Operators:

- **CONCEPT:** Operators are also used compare expressions (relational operators).

  - It’s like asking a true/false question
  - $A < B$ means “$A$ is less than $B$, true or false?”
  - $A > B$

  - 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 to
    - $a = b$ equal to
    - $a <> b$ not equal

Basic Conditional

- **Use conditionals to test to see if a condition holds:**
  - If $\text{temp} < 32$ Then
    state = “frozen”
    form = “ice”
  - End If

- **General form of basic conditional:**
  - If <T / F expression> Then
    <code statements>
  - End If

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

General Conditional Statement

- **CONCEPT:** When one set of statements must be performed for the true conditions and a different set of statements are needed for the false conditions, use the If-Then-Else statement

  - **General form**
    - If <T / F expression> Then
      <code statements>
    - Else
      <code statements>
    - End If

  - **Example:**
    If sky = “clear” AND temp > 70 Then
      clothing = “tank top”
    Else
      clothing = “sweats”
    End If
**Concept:** An advantage of the general conditional is that it can be imbedded within another conditional.

If sky = "clear" AND temp > 70 Then
  clothing = "tank top"
  If laundry = "clean" Then
    clothingColor = "purple"
  End If
Else
  clothing = "sweats"
  If ground = "muddy" Then
    shoes = "boots"
  End If
End If

**Exercise #1**

What does this print?

```vbscript
Dim x As Integer
x = 10
If x = 1 Then
  Print "octopus"
ElseIf x = 2 Then
  Print "squid"
Else
  Print "clam"
End If
Print "mollusk"
```

**Let's Move From Theory to Practice!**

We want to 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 if the input is positive or negative?
- What should we do with an input of zero?
- How should we output the "positive" or "negative" evaluation to the user?
  - Be Creative!
- How do we get started?