Iteration -- Once Is Not Enough

Recall the tripartite lecture strategy that begins today.

- Lectures
- Projects
- On-line work
- Concepts
- Questions on binary search
- Debugging tools
- Iteration, select
- Netherlands

Key Points Of Binary Search

- Summarizing binary search high points
  - Computing the probe from the end points
  - Correcting the probe for the month change
  - Where to call the Guess procedure
  - Updating the end point based on reply
  - Terminating when end points match
- If there are $2^n$ items in the interval initially, it takes $n$ probes to complete the search
- Since there are $32 = 2^5$ or fewer days in every sign the Day Finder will locate the answer in 5 guesses
More Control

- “Control statements” affect which statements in a program are executed
- So far, **If** statements are the only control statement studied, though procedure **calls** and **events** also change the normal sequencing of statements
- Programming concepts introduced today
  - Select Case … like **If** it picks among alternatives
  - For Next … allows operations to be repeated
  - Do While and Do Until variations

Select Case

- Syntax …
  ```vbnet
  Select Case <variable>
  Case <value>
     <statements>
  Case <value>
     <statements>
  ...
  End Select
  ```
- Only one alternative is chosen
- Use **else** if you need “for all other cases”
- Don’t forget the **End Select**
Repetition ...

- If all of the statements of a program were performed at most once, computers would not be very useful.
- Repeated execution of statements allows programs to Guess a birthday, check through a list, etc.
- The most common mechanism of repetition is the "iterative loop," which has several forms, two of which are available in Basic:
  - **For** ... allows a specific range to be covered
  - **Do** ... is open-ended, continuing until a condition occurs

### For ... Next

- **Syntax**
  - **For** <index variable> = <low bound> To <high bound>
  - **Next** <index variable>
- The index variable should be an integer

```plaintext
For i = 1 To 1000
    Form1.Print "I will not talk in class. -- Bart Simpson"
Next i
```

```plaintext
For i = 1 To 1000
    Form1.Print i & " will not talk in class. -- Lisa"
Next i
```
The For loop bounds can be constants (1000), variables (numEntries) or expressions (numEntries + 1).

- Add step either to skip
  ```v
  For indx = 2 To numAnimals Step 2
      Call boardArc (indx)
  Next indx
  ```

- Or to decrement
  ```v
  For indx = 99 To 0 Step -1
      Call bottlesOfBeer (indx)
  Next indx
  ```

Some repetitions, rather than counting, must continue until a certain condition occurs ... use while loops

- while means, “continue as long as condition true”

Syntax ...

```
Do
    <statements>
Loop While <condition>
```

```
Do While loDate <> hiDate
    Probe ( ... )
Loop
```
The Until Variation

- Until means, “continue as long as condition false”
- Syntax ...

```plaintext
Do
  <statements>
Loop Until <condition>
```

```plaintext
Do
  Probe ( ... )
Loop Until loDate = hiDate
```

```plaintext
Do Until loDate = hiDate
  Probe ( ... )
Loop
```

Debugging Tools

- Move to VB6.0
- Facilities exhibited ...
  - Debug Menu
  - Break points, setting and clearing
  - Probing values
  - Immediate window
  - Printing values