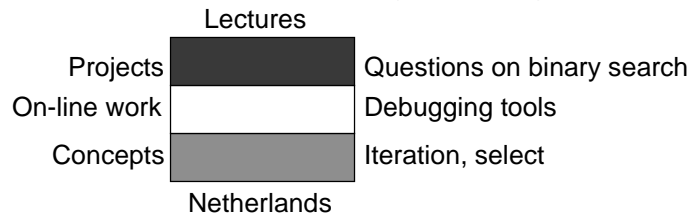


Iteration -- Once Is Not Enough



Recall the tripartite lecture strategy that begins today.



© Larry Snyder, 1999



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



© Larry Snyder, 1999

- ❖ “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



- ❖ Syntax ...

```
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**

```
Select Case size
Case "short"
  basePrice = 1.10
Case "tall"
  basePrice = 1.50
Case "grande"
  basePrice = 1.90
Case Else
  MsgBox "What size?"
End Select
```



CSE 100 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



© Larry Snyder, 1999

CSE 100 For ... Next

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

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

```
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
I will not talk in class. -- Bart Simpson
```

```
For i = 1 To 1000
  Form1.Print i & " will not talk in class. -- Lisa"
Next i
```

```
1 will not talk in class. -- Lisa
2 will not talk in class. -- Lisa
3 will not talk in class. -- Lisa
4 will not talk in class. -- Lisa
5 will not talk in class. -- Lisa
6 will not talk in class. -- Lisa
7 will not talk in class. -- Lisa
```

© Larry Snyder, 1999

**CSE
100**

For ... Next (Continued)

- ❖ The **For** loop bounds can be constants (1000), variables (**numEntries**) or expressions (**numEntries - 1**)

- ❖ Add **step** either to skip

```
For indx = 2 To numAnimals Step 2
  Call boardArc (indx)
Next indx
```

- ❖ Or to decrement

```
For indx = 99 To 0 Step -1
  Call bottlesOfBeer (indx)
Next indx
```



**CSE
100**

While Loops

- ❖ 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	Do While <condition>
<statements>	<statements>
Loop While <condition>	Loop

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

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



❖ **Until** means, “continue as long as condition false”

❖ Syntax ...

Do	Do Until <condition>
<statements>	<statements>
Loop Until <condition>	Loop

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

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



❖ Move to VB6.0

❖ Facilities exhibited ...

- + Debug Menu
- + Break points, setting and clearing
- + Probing values
- + Immediate window
- + Printing values

