

## Reasoning Through Applications



Algorithm design, programming and problem solving usually require thinking and creativity, and they benefit from experience. But it is possible to approach these tasks to simplify the process of finding a solution. Today we work through a case.

© Larry Snyder, 1999



## What's Making This Difficult?

- ❖ Algorithm design, programming, application development, etc. are intellectually tough ... why?
  - ❑ There is no "cookbook solution" ... each case has its own logic and requires its own reasoning
  - ❑ The solution must be *exactly right* in every detail
  - ❑ The language used to express the solution (Basic) is new, strange and unforgiving
  - ❑ The context -- Windows operating system, the VB6.0 development environment, the UW computing facilities -- is new and complicated
  - ❑ The instructors present examples that are "all prepared" so you do not see the actual programming, thinking, debugging and mistakes they make

Learn by example and analogy

© Larry Snyder, 1999



## The Latte Problem

- ❖ Ken's example from yesterday's lab provides the topic for this exercise

"Price a double, tall latte, please."

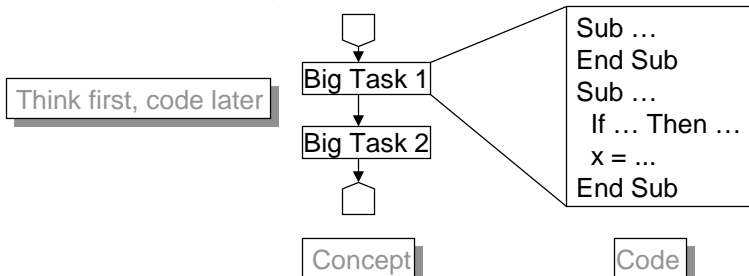
- ❖ Recall that the task is to figure the cost of a latte, given there is a base price for each size, extra cost for each shot of espresso and sales tax.

© Larry Snyder, 1999



## Thinking About The Solution ...

- ❖ The most common advice about programming is: "Reason the solution through first, code it later"
  - ❑ It is very easy to be distracted by the mechanism of writing a program and the IDE
  - ❑ Since the solution demands the reasoning effort no matter how you go about it -- reasoning before the coding or reasoning during the coding -- reasoning first allows you to think about the problem more abstractly



© Larry Snyder, 1999



## What's To Be Done?

- ❖ The very first step is to understand what is to be done
- ❖ Generally, this entails specifying the inputs, the outputs and the processing that gets from in to out

Task: Compute the price of a latte

The price is determined by three quantities:

- + Base price, given by the cup size
- + Number of shots of espresso
- + State sales tax
- ❖ Of these the coffee drinker specifies the first two, making them program inputs: size, shots
- ❖ The output is the price: price

© Larry Snyder, 1999



## Further specification ...

- ❖ The size input is further specified to have three values:
  - + Short with basePrice \$1.10
  - + Tall with basePrice \$1.50
  - + Grande with basePrice \$1.90
- ❖ The shots input is further specified to be any nonnegative number, i.e. 0 or more
  - + The per shot price is \$0.65
- ❖ The sales tax input (fixed for the problem) is 8.6%
- ❖ The price output is further specified to be
  - +  $\text{price} = \text{basePrice} + \text{shots} * 0.65 + \text{figureTax}(\text{subTotal})$

Notice, size is different from basePrice, implying basePrice is a necessary new variable

Summary: The coffee drinker needs to give desired size and shots, and will receive back the displayed price.

© Larry Snyder, 1999



## Finding A Solution

- ❖ With the task defined and the Inputs/Outputs specified ... Decide how the task will be solved?
- ❖ The best advice is “Formulate a ‘high level solution’ and then refine each operation into simpler form.”
- ❖ Solution considerations ...
  - ✦ There are two user inputs, size and shots, and they can be given in either order
  - ✦ The proper basePrice must be computed from size
  - ✦ The figureTax procedure is needed to compute the sales tax from the subTotal ... this is simply  $0.086 * \text{subTotal}$
  - ✦ The subTotal, which is  $\text{basePrice} + \text{shots} * 0.65$ , must be computed ... can be used in computing the price
  - ✦ The price must be computed ...  $\text{subTotal} + \text{figureTax}(\text{subTotal})$
  - ✦ The result must be displayed

© Larry Snyder, 1999



## Consequences Of The Considerations

There are two user inputs, size and shots,  
and they can be given in either order

- ❖ When there are two (or more) inputs, it is necessary to postpone the computation until all inputs are available ... so to be notified when the input has been fully specified, use a button to command “calculate price now”
- ❖ The processing can be performed when the calcPrice button is clicked, so plan to put “processing” in the Sub calcPrice\_Click()

© Larry Snyder, 1999

**CSE  
100**

## Consequences Of Considerations II

The proper basePrice must be computed from size

- ❖ size has three values -- short, tall, grande -- and these could be specified with option buttons, e.g. optShort, optTall, optGrande; shots is textbox
- ❖ The basePrice can be set with each button, or the size can be "captured" for processing later
- ❖ Either way, need global variable

### Case 1: Set basePrice

```
Sub optShort_Click()  
    basePrice = 1.10  
End Sub
```

### Case 2: Capture data

```
Sub optShort_Click()  
    size = "short"  
End Sub  
  
...  
If size = "short" Then  
    basePrice = 1.10  
If size = "tall" Then  
    ...
```

© Larry Snyder, 1999

**CSE  
100**

## Consequence Of Considerations III

The figureTax procedure is needed to compute the sales tax from the subTotal ... this is simply  $0.086 * \text{subTotal}$

- ❖ The figureTax computation can be either a function or a procedure. Either way, it takes the subTotal as an actual parameter, and uses a formal parameter, say sT, in the definition
- ❖ If figureTax is a procedure it will also need a parameter, say tax, to pass the value back to the main program
- ❖ Or figure tax inline

```
Function figureTax (sT As Single) As Single  
    figureTax = sT * 0.086  
End Function
```

```
Sub figureTax (sT, tax As Single, tax)  
    tax = sT * 0.086  
End Sub
```

© Larry Snyder, 1999



## Consequences Of Considerations IV

The subTotal, which is  $\text{basePrice} + \text{shots} * 0.65$ , must be computed ... can be used in computing the price  
The price must be computed ...  $\text{subTotal} + \text{figureTax}(\text{subTotal})$

- ❖ The simple solution is just to compute the subTotal and then add that into the price

$\text{subTotal} = \text{basePrice} + \text{shots} * 0.65$

$\text{price} = \text{subTotal} + \text{subTotal} * 0.086$

The result must be displayed

- ❖ A label control, say lblPrice, must be introduced and assigned the value computed for price

© Larry Snyder, 1999



## Ready To Code

- ❖ At this point we know what controls are needed -- 3 option buttons, text box, command button, label
- ❖ The variables needed -- subTotal, ... -- are known
- ❖ The processing is known ...
  - + Capture data at the option buttons
  - + Figure result on command button
- ❖ The program can now be written, and tested interactively, as portions are finished
- ❖ Error checking and testing are needed, but they are topics for the future

© Larry Snyder, 1999



## One Solution ... I

---

Every program makes a specific set of choices

```
Option Explicit
Dim size As String

Private Sub btnCalculatePrice_Click()
    Dim shots As Integer
    Dim price As Single

    shots = txtShots.Text
    Call calcLattePrice(price, shots, size)
    Call addSalesTax(price)

    lblPrice.Caption = "$" & Round(price, 2)
End Sub
```

© Larry Snyder, 1999



## One Solution ... II

---

Every program makes a specific set of choices

```
Sub calcLattePrice(p As Single, sh As Integer,
                  sz As String)
    Dim basePrice, extraCharge As Single

    If sz = "short" Then
        basePrice = 1.1
    ElseIf sz = "tall" Then
        basePrice = 1.5
    ElseIf sz = "grande" Then
        basePrice = 1.9
    End If

    extraCharge = 0
    If sh > 1 Then
        extraCharge = (sh - 1) * 0.65
    End If

    p = basePrice + extraCharge
End Sub
```

© Larry Snyder, 1999



## One Solution ... III

---

Every program makes a specific set of choices

```
Sub addSalesTax(p As Single)
    p = p + p * 0.086
End Sub
```

```
Private Sub optGrande_Click()
    size = "grande"
End Sub
```

```
Private Sub optShort_Click()
    size = "short"
End Sub
```

```
Private Sub optTall_Click()
    size = "tall"
End Sub
```