

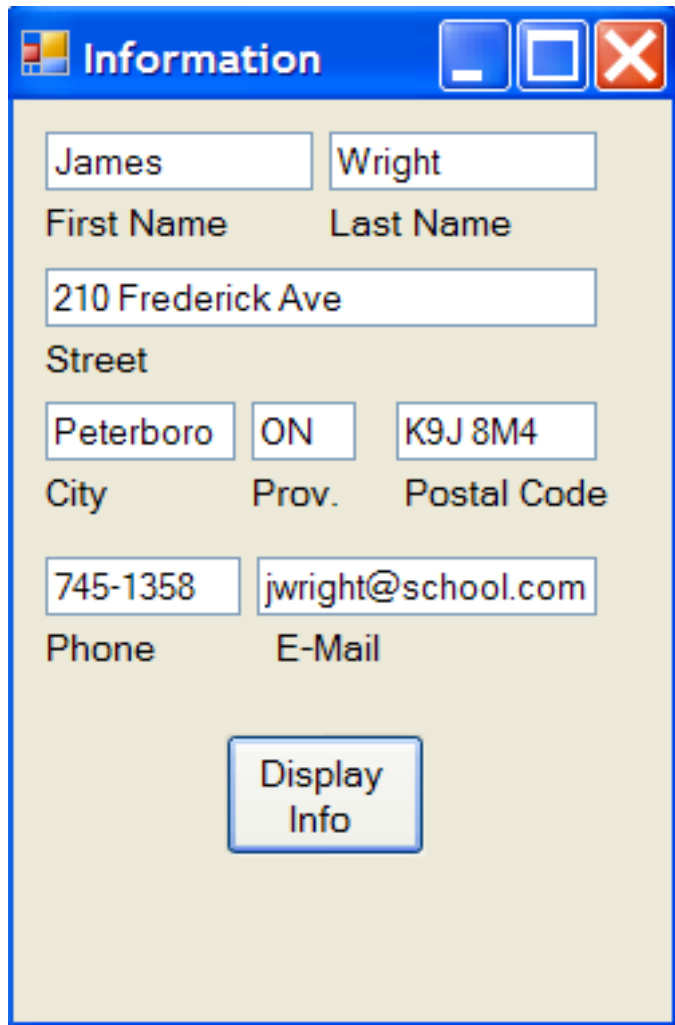
Creating A Personal Info Form

This project involves creating a form where the user inputs information. The key to a good program such as this is making a layout that is easy for the user use.

Most users are pretty familiar with filling out these forms. Generally they follow a pattern where the users enters their first name, last name, street, city, province and postal code.

It is no co-incidence that this is the standard means of addressing an envelope when sending it by mail.

The Form



The screenshot shows a window titled "Information" with a standard Windows title bar (minimize, maximize, close buttons). The form inside has a light beige background and contains the following elements:

- Two text boxes: "James" (First Name) and "Wright" (Last Name).
- Labels: "First Name" and "Last Name" positioned below their respective text boxes.
- A single wide text box: "210 Frederick Ave" (Street).
- Label: "Street" positioned below the wide text box.
- Three text boxes: "Peterboro" (City), "ON" (Prov.), and "K9J 8M4" (Postal Code).
- Labels: "City", "Prov.", and "Postal Code" positioned below their respective text boxes.
- Two text boxes: "745-1358" (Phone) and "jwright@school.com" (E-Mail).
- Labels: "Phone" and "E-Mail" positioned below their respective text boxes.
- A button: "Display Info" located at the bottom center of the form.

The text boxes are set out in the order a user would expect the information to be submitted.

Labels under each text box help the user to know where the info goes.

Text boxes and labels are neatly sized and arranged according to their function.

The text boxes are name in order:
txtFirstName, txtLastName, txtStreet,
txtCity,txtProvince,txtPostalCode,txtPhone,txtEMail.

The label names follow the same pattern except they have the lbl prefix instead of txt.

The button name is btnDisplayInfo

The naming of objects in a consistent and logical manner is important to creating code that is easy to read and maintain.

The Variables

Each one of these text boxes is going to contain information which will need to be assigned to a variable. Since this information consists of text, the variable types required will be String variables.

Declare a String variable for each of the required pieces of information.

Use logical names for each variables, names which hint at how the variable will be used.

Begin each variable name with a lower case letter.

Spaces are not allowed in variable names.

Use camel notation for multi-word names. eg. `dateOfBirth`

All code for this program including declarations, will go in the Button Sub.

Variable Declarations

These are the variables that will hold information from the various text boxes. Notice how their names provides hints as to what information they will hold.

Each variable begins with a lower case letter and when multi word names are used, the names are written in camel notation style.

```
Dim firstName As String
Dim lastName As String
Dim street As String
Dim city As String
Dim province As String
Dim postalCode As String
Dim email As String
Dim phone As String
```

It is important to remember that variables must be declared **before** they can be used!

Assigning Values to Variables

In order to get the information from the text boxes to the variables we must write some code.

We use the assignment operator(=) to assign the data on the right to the variable on the left.

The target or destination of the values/information/data(whatever you want to call it), goes on the left side of the assignment operator.

In these cases we are assigning the text values of each of the text boxes to an appropriate String variable.

Assignment Statements

```
firstName = txtFirstName.Text
```

```
lastName = txtLastName.Text
```

```
street = txtStreet.Text
```

```
city = txtCity.Text
```

```
province = txtProvince.Text
```

```
postalCode = txtPostalCode.Text
```

```
phone = txtPhone.Text
```

```
email = txtEMail.Text
```

Concatenating the Information

Once all the data is collected and assigned to variables, the task now, is to join or concatenate the variables and display the information.

This requires some thought and planning. If you try and join all the variables together, you are left with code that is awkward to read. eg.

```
MsgBox(firstName & lastName & address & city & province & postalCode & phone & email)
```

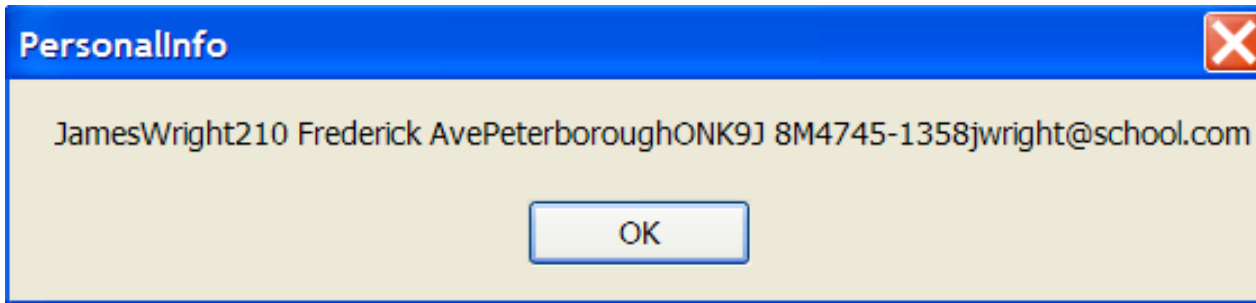
Code this long does not easily fit on a single line in the code editor.

To help this, it is possible to break long code into multiple lines by us an underscore line (_)

```
MsgBox(firstName & lastName & street _  
& city & province & postalCode & phone & email)
```

Oops

Not very professional
looking!



To fix this output we need to have:

- a space between the first and the last name
- a new line for the street
- a new line for the city, province and postal code
- spaces between the city, province and postal code
- a new line for the phone number and the email address
- spaces between the phone number and the email address

It's a wrap!

Having output move down a line is often referred to as wrapping text.

Lets introduce some line breaks in our output.

To do this we will create a String variable which we will call wrap.

We will then assign some special values to this variable which will force a line break.

```
Dim wrap As String  
wrap = Chr(13) & Chr(10)
```

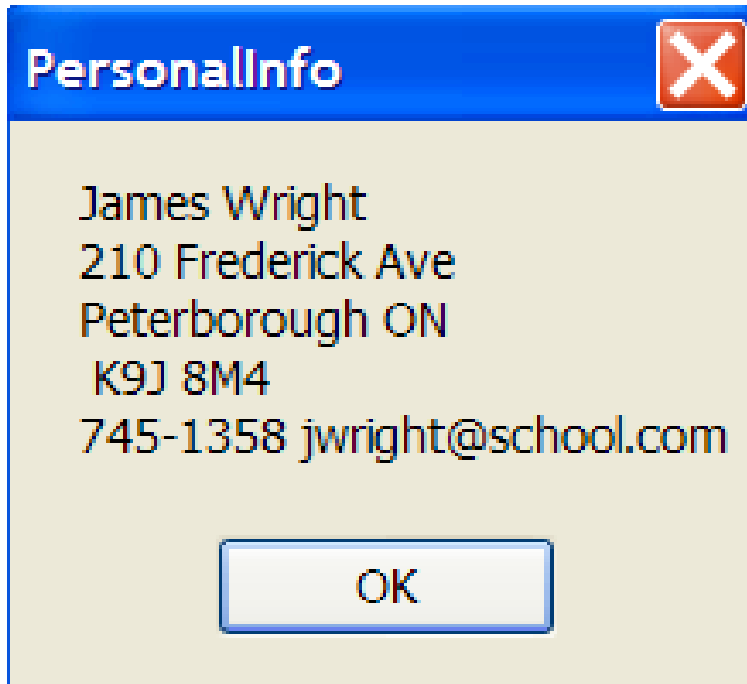
Now whenever we insert the variable wrap into a string, it will cause the text in the output to move down to the next line.

As well we will put spaces between some of the output by concatenating a pair of quotations containing a blank space

“ “

The Result

```
MsgBox(firstName & " " & lastName & wrap & _  
street & wrap & city & " " & province & wrap _  
& " " & postalCode & wrap & phone & " " & email)
```



Well, the result is there, but it sure is a lot of code to stuff into a single Message Box.

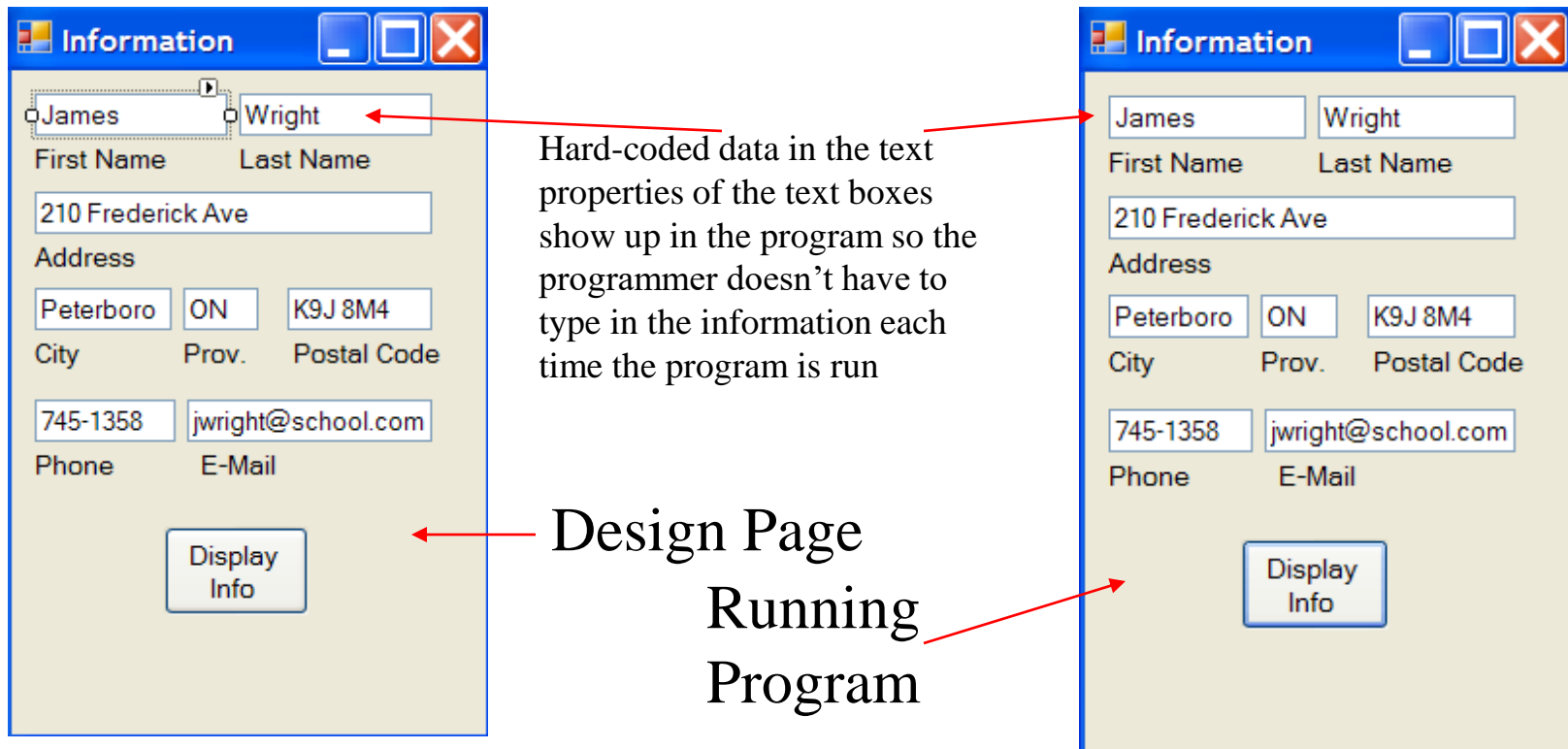
Pity the poor person who has to look at this code later should any debugging or modifications be required.

A Hint for Testing

As you are creating and testing this project you will quickly discover what a nuisance it is to have to type in the personal information in each text box every time.

To help save time when testing, you should 'hard-code' some test info into the program by writing data in the the text box properties. This way when you run the program, the info is already there.

When th project is finished and looking the way you want it to, ou can remove the hard-coded text so that users can put in their own information.



Consolidating the Variables by Concatenation

As said before, this is a lot of code for one message box.

```
MsgBox(firstName & " " & lastName & wrap & _  
street & wrap & city & " " & province & wrap _  
& " " & postalCode & wrap & phone & " " & email)
```

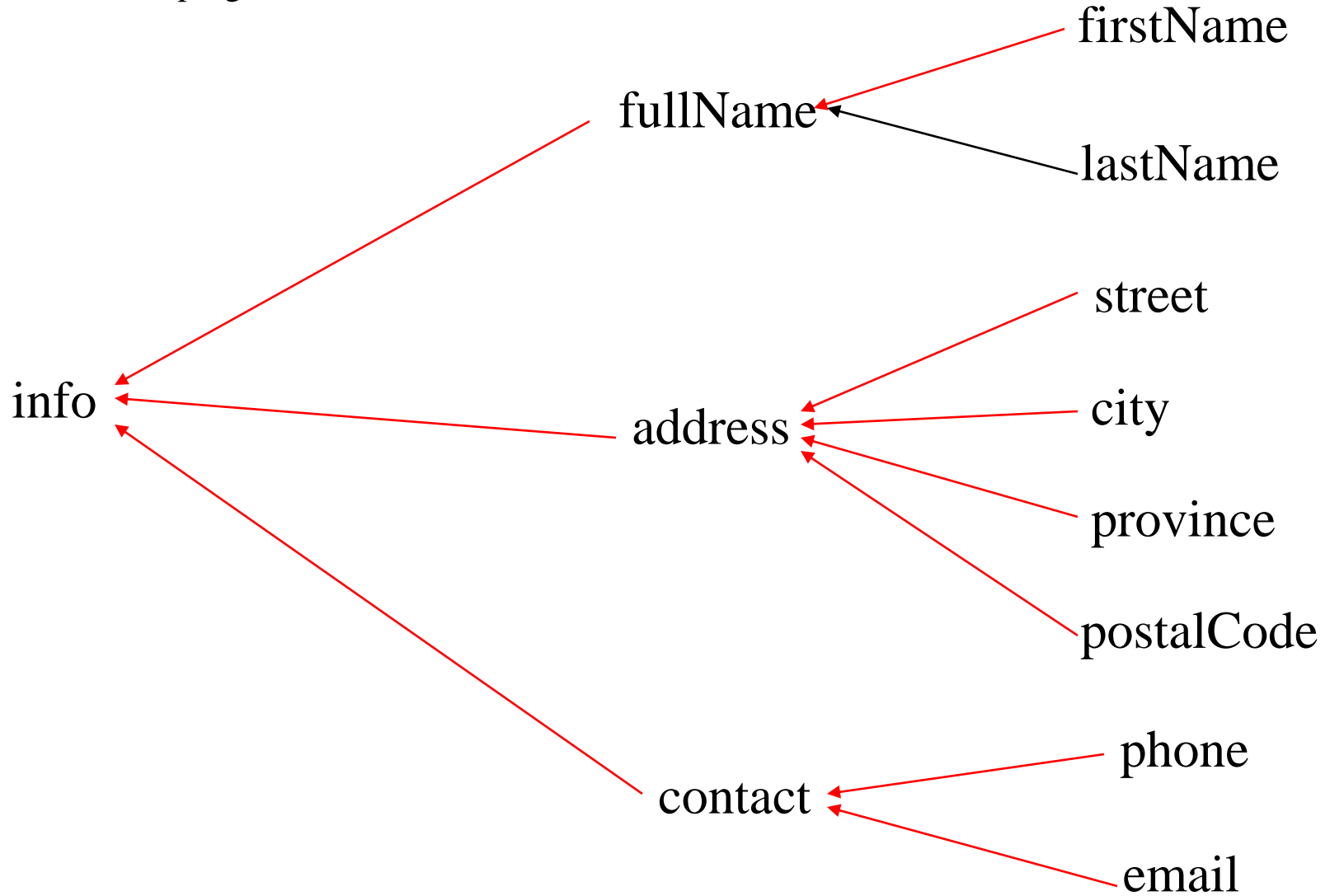
We can simplify this by concatenating the variables `firstName` and `lastName` along with the space between them and then assigning them to a third variable which we will call `fullName`. (remember to declare `fullName` first).

```
Dim fullName As String  
fullName = firstName & " " & lastName
```

While this extra step might seem fussy and unnecessary, this process help make the code more easily read and managed.

Extending the Consolidation of Variables

This chart should help illustrate the steps involved in consolidating and concatenating the various variables in this program.



The Code Consolidated

We will be using 4 new variables so we need to declare them first.

```
Dim fullName As String
Dim address As String
Dim contact As String
Dim info As String
```

Now we can group the information from our text boxes into larger categories.

```
fullName = firstName & " " & lastName
```

```
address = street & wrap & city _
& " " & province & " " & postalCode
```

```
contact = phone & " " & email
```

```
info = fullName & wrap & address & wrap & contact
```

“info” ... the Chief of the Variables

Now it is time to group or consolidate the 3 higher level variables into one overall variable.

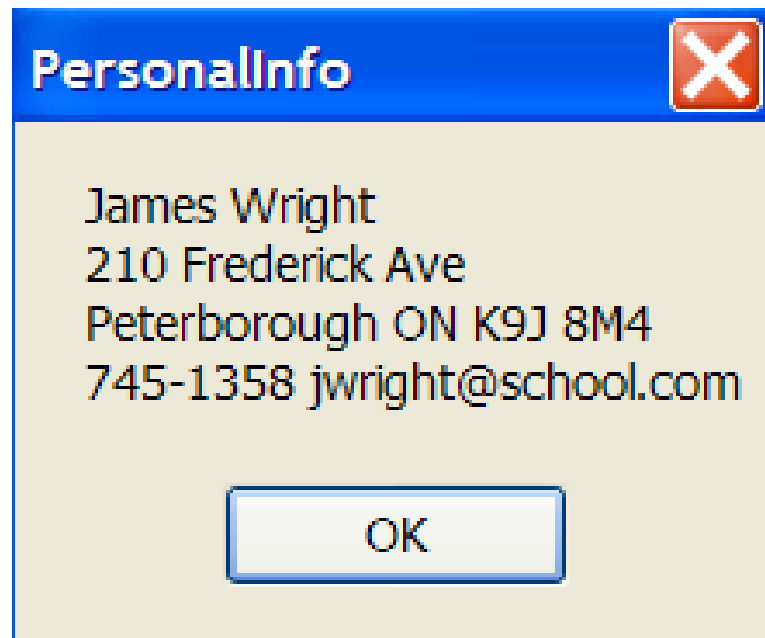
```
info = fullName & wrap & address & wrap & contact
```

‘info’ now contains the sum of all the other data that we collected plus the formatting spaces and new lines.

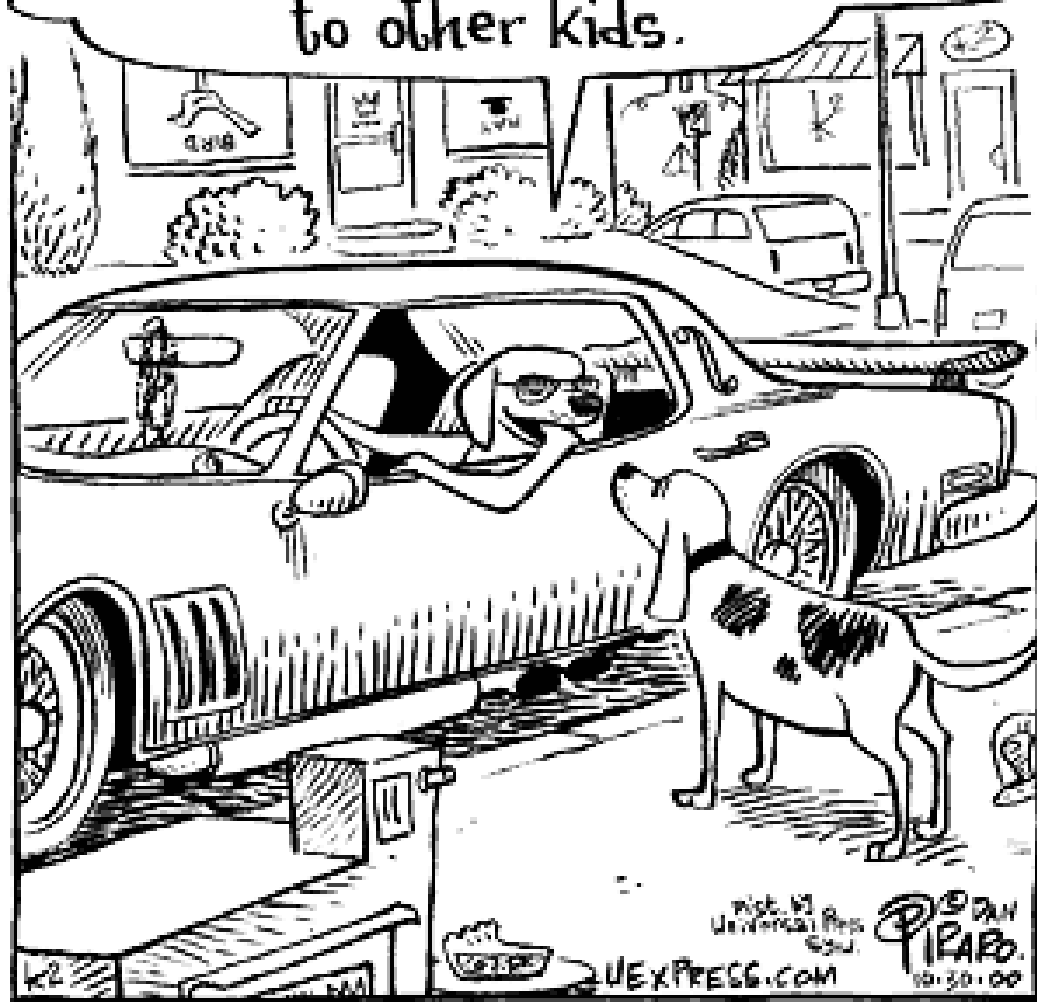
Now simply by putting the variable info inside the message box.....

The Power of a Single Variable

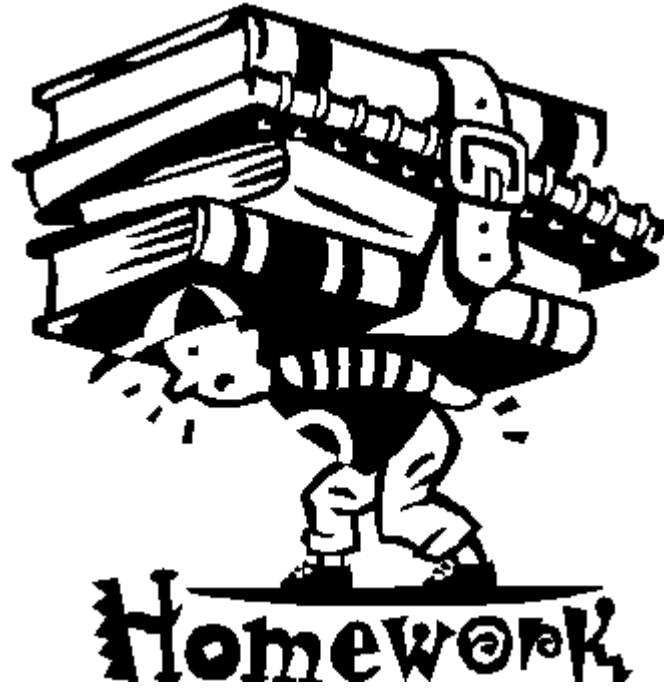
MsgBox (info)



I finally wised up — instead of EATING his homework, I started SELLING it to other kids.



Walt Disney
Universal Press
©2001
DAN
PIRARD
10.30.00
UEXPRESS.COM



Finish PersonalInfo and hand in for Monday.

Bonus marks for creating a drop down box for province.