Software Design and Development

5 Selections

Terry Marris July 2009

In handout #4 we looked at the primitive data types. Now we start looking at program structures. We look at selections. But before that we look at the relational operators.

5.1 The Relational Operators

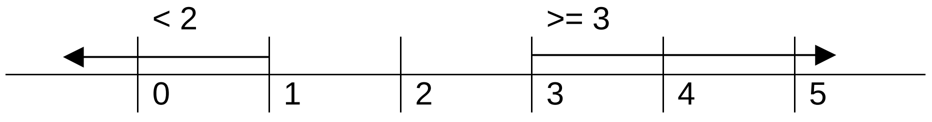
The relational operators include:

description	symbol	example	explanation
less than	<	2 < 3	2 is less than 3
more than	>	3 > 2	3 is more than 2
equal to	=	2 = 2	2 is equal to 2
less than or equal to	<=	2 <= 3	2 is less than **or** equal to 3
		2 <= 2	2 is les than **or** equal to 2
more than or equal to	>=	3 >= 2	3 is more than **or** equal to 2
		2 >= 2	2 is more than **or** equal to 2
not equal to	<>	2 <>3	2 is not equal to 3

Notice that **L**ess than has its point on the **L**eft, and mo**R**e than has its point on the **R**ight.

Notice the order: <=. The < symbol comes before the = symbol. And there is no space between the two symbols.

In picture of a number line:



5.2 If .. Else ...

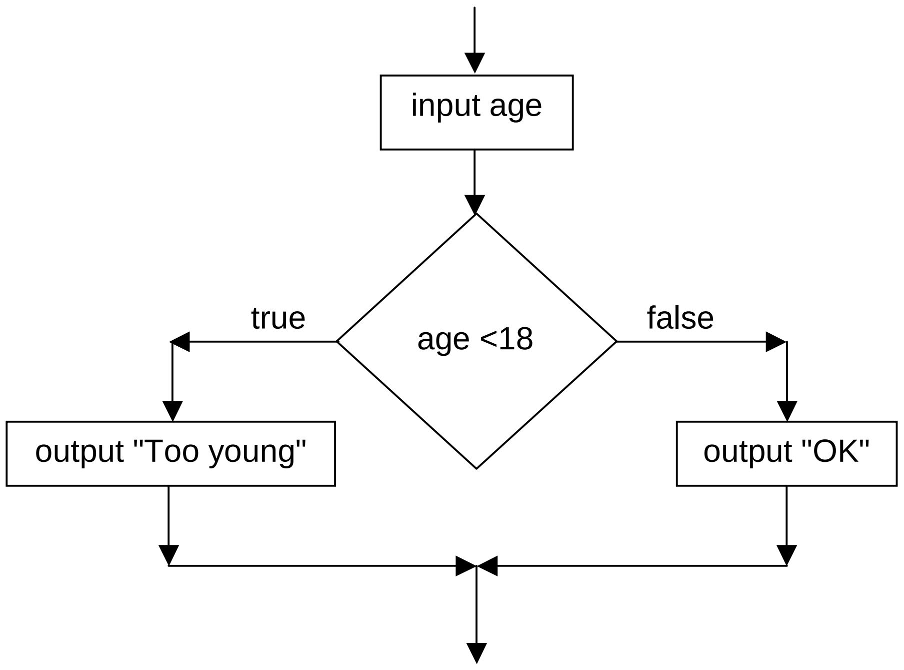
The design for our first program is:

Structured English

```
read in a person's age
if age < 18 then
   write out "Too young"      <——  done if the age is less than 18
else
   write out "OK"    <————————  done if the age is NOT less than 18
endif
```

Or:

Program Flowchart



You select to display either:

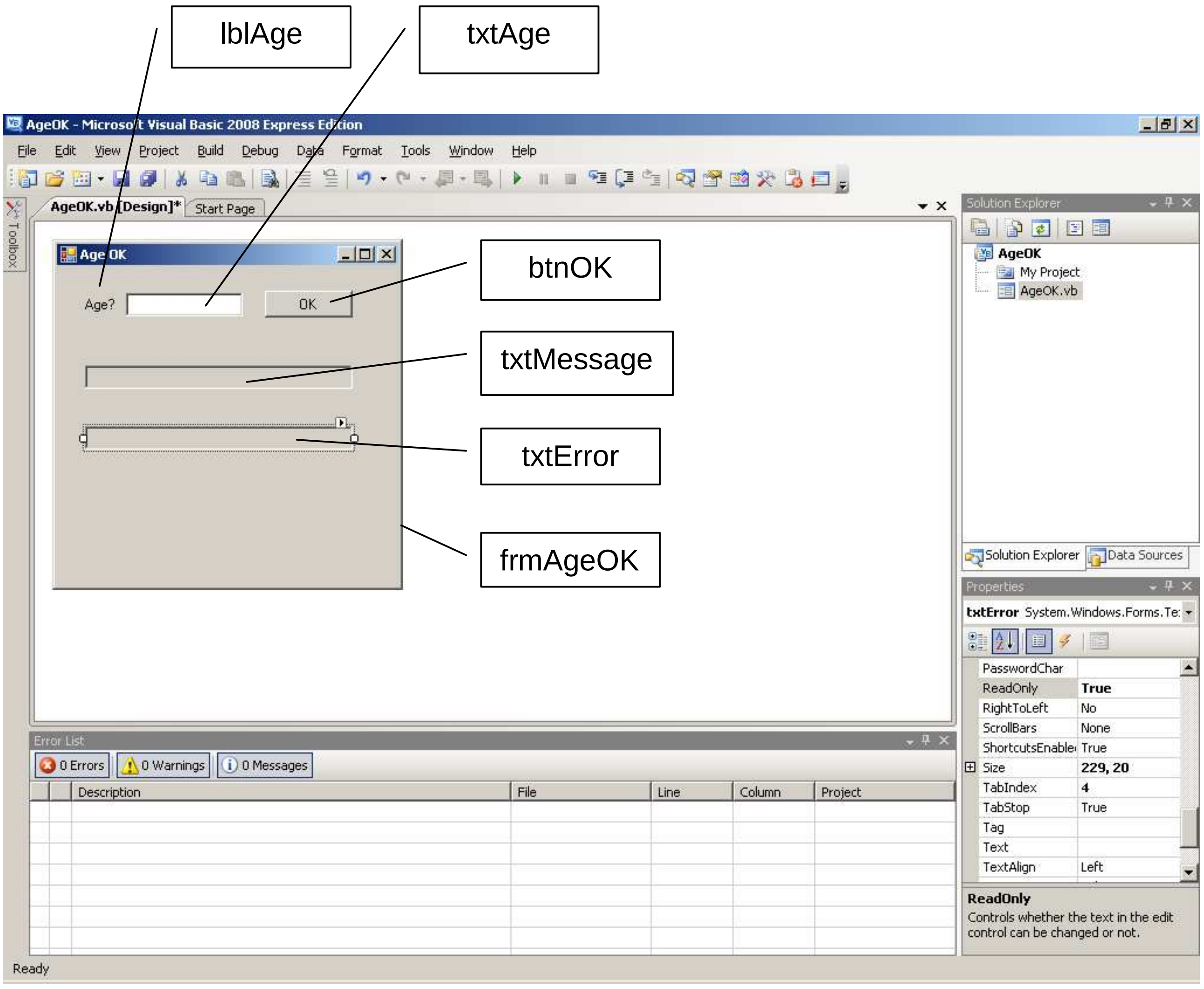
- *To young* if the age is less than 18

Or

- *OK* if the age is not less than 18

A selection is a bit like a T-junction: when you come to it you turn either one way or the other; you follow either one path or the other.

User Interface


lblAge
txtAge
btnOK
txtMessage
txtError
frmAgeOK
AgeOK - Microsoft Visual Basic 2008 Express Edition
File Edit View Project Build Debug Data Format Tools Window Help
Toolbox
AgeOK.vb [Design]* Start Page
Age OK
Age?
OK
Solution Explorer
AgeOK
My Project
AgeOK.vb
Solution Explorer Data Sources
Error List
0 Errors 0 Warnings 0 Messages
Description File Line Column Project
Ready

Properties

txtError System.Windows.Forms.Te:

PasswordChar	
ReadOnly	**True**
RightToLeft	No
ScrollBars	None
ShortcutsEnable	True
Size	**229, 20**
TabIndex	**4**
TabStop	True
Tag	
Text	
TextAlign	Left

ReadOnly
Controls whether the text in the edit control can be changed or not.

1. Start a new project and name it *AgeOK*
2. Set FileName = *AgeOK.vb*
3. Set Form properties: click on an empty part of the form
   a. Name = *frmAgeOK*
   b. Text = *Age OK*
4. Drag controls onto the form and set their properties as listed below:
   a. Label
      i. Name = *lblAge*
      ii. Text = *Age?*

   b. TextBox
      i. Name = *txtAge*

   c. Button
      i. Name = *btnOK*
      ii. Text = *OK*

- **d.** TextBox
    - **i.** Name = *txtMessage*
    - **ii.** ReadOnly = *True*

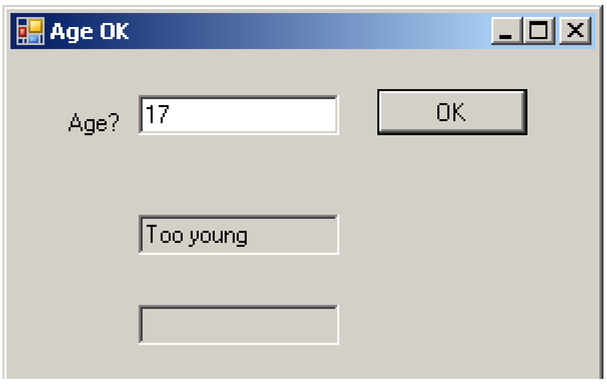
- **e.** TextBox
    - **i.** Name = *txtError*
    - **ii.** ReadOny = *True*

Programmers Code

5. The VB code is placed under the OK button.

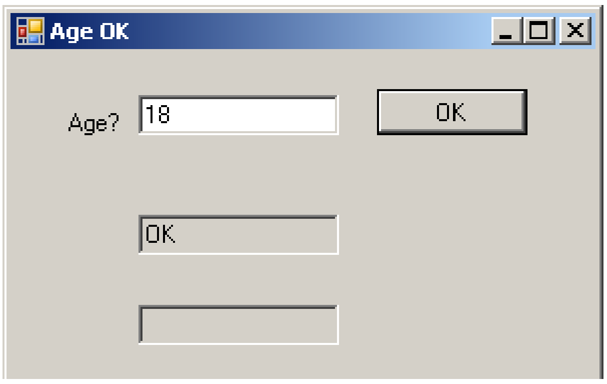
```
Try
    Dim intAge As Integer = Convert.ToInt32(txtAge.Text)
    If intAge < 18 Then
        txtMessage.Text = "Too young"
    Else
        txtMessage.Text = "OK"
    End If
Catch ex As FormatException
    txtError.Text = "Error: not a number"
Catch ex As OverflowException
    txtError.Text = "Error: number too big"
End Try
```

Examples of program runs:


Age OK

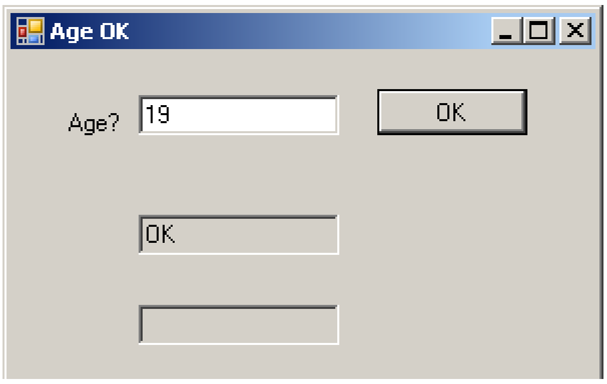
Age? 17 OK

Too young


Age OK

Age? 18 OK

OK


Age OK

Age? 19 OK

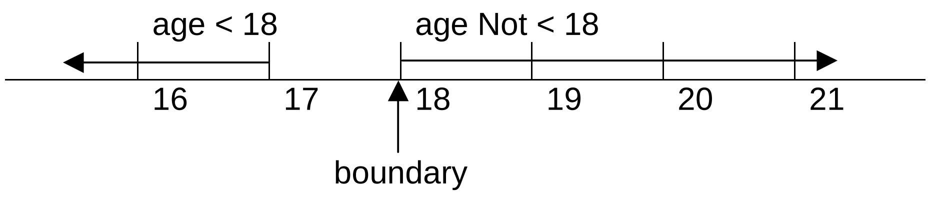
OK

5.3 Boundary Testing

Look at

```
If intAge < 18 Then
   txtMessage.Text = "Too young"
Else
   txtMessage.Text = "OK"
End If
```

intAge < 18 is an example of a *boundary*. If *intAge* had the value 17, then *Too Young* would be displayed. If *intAge* had the value 18, then *OK* would be displayed. A boundary occurs wherever a small change in value (e.g. from 17 to 18) causes a big change in effect (e.g. from *Too Young* to *OK*).



You always test your programs with values close to the boundary points because programmers (people like you and me) are sometimes human and humans sometimes make mistakes. Errors in programming logic are often found close to boundary points.

5.4 Test Plans

The purpose of testing is to find errors. If you had a job as a tester, you would be looking for errors in programs. No matter how many errors you find, you can never guarantee to find every error. (Why do programs sometimes freeze? Why do e-mails sometimes "get lost"? Why do aircraft sometimes crash? Check out http://catless.ncl.ac.uk/risks for some horror stories.)

A test plan always has four columns:

Test #	Test Data	Reason	Expected Result
1	intAge = 17	boundary between Too young and OK	Too young
2	intAge = 18		OK
3	intAge = 19		OK

When you run your program you compare your actual results with the ones you were expecting. Sometimes you are surprised. The result you actually get is not the one you expected. A mistake has been made somewhere. What do you do? You **either** fix the error (the error could be in your test plan or in your program or ...) **or** you write: *There is an error. I don't know why. I will fix it later*. The best programmers are up front about their errors. Better that you tell me about your errors than it is for me to find them for you. Better you find your errors before your customers do!

Exercise 5.1

1. Design, write, comment and test VB programs to:

   a. read a person's age. If the age is 18 or more then write *ok*, otherwise write *too young*.

   b. read the thermometer. If the temperature is less than (or equal to) zero write *freezing*, otherwise write *not too cold*.

   c. read the clinical thermometer. If the temperature is more than 37.4 write *patient has a fever*, otherwise write *patient does not have a fever*.

   d. read the number of people who have passed through a turnstile. If it is more than 50000 write *stadium full*, otherwise write *seats available*.

5.5 If ... Else If ...

The next example is the familiar exam mark problem.

```
read examMark
if examMark >= 0 and examMark <= 39 then
    write Fail          ◄———   done if the examMark is between 0 and 39 inclusive
else if examMark >= 40 and examMark <= 100 then
   write Pass           ◄———   done if the examMark is between 40 and 100 inclusive
else
    write Error         ◄———   done if none of the above cases are true
endif
```

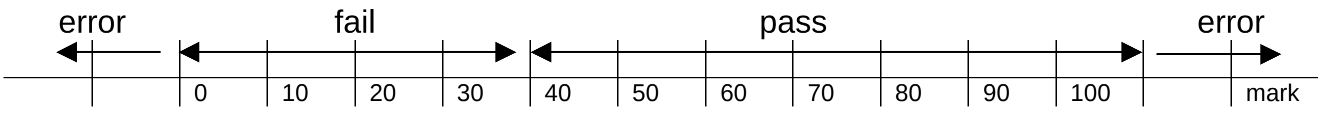
Case? An if statement that is either true or false.

We write Fail if the exam mark is between 0 and 39 inclusive.

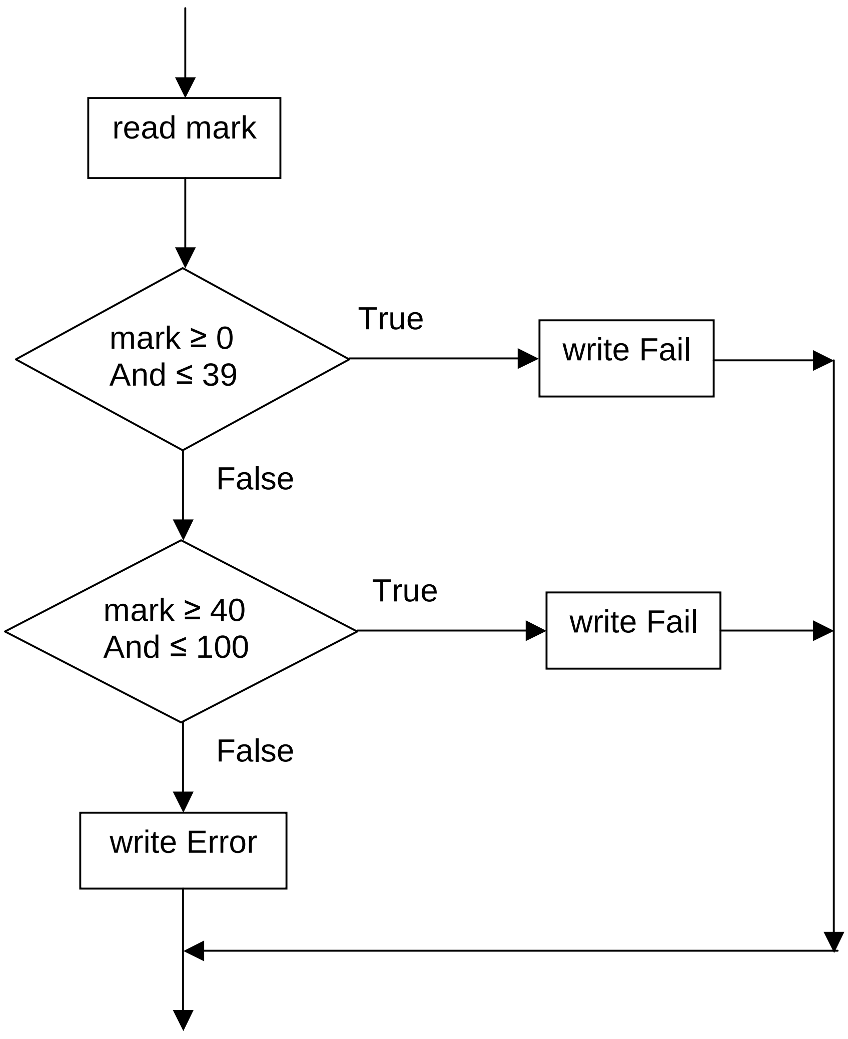
We write Pass if the exam mark is between 40 and 100 inclusive.

We write Fail is none of the above is true.

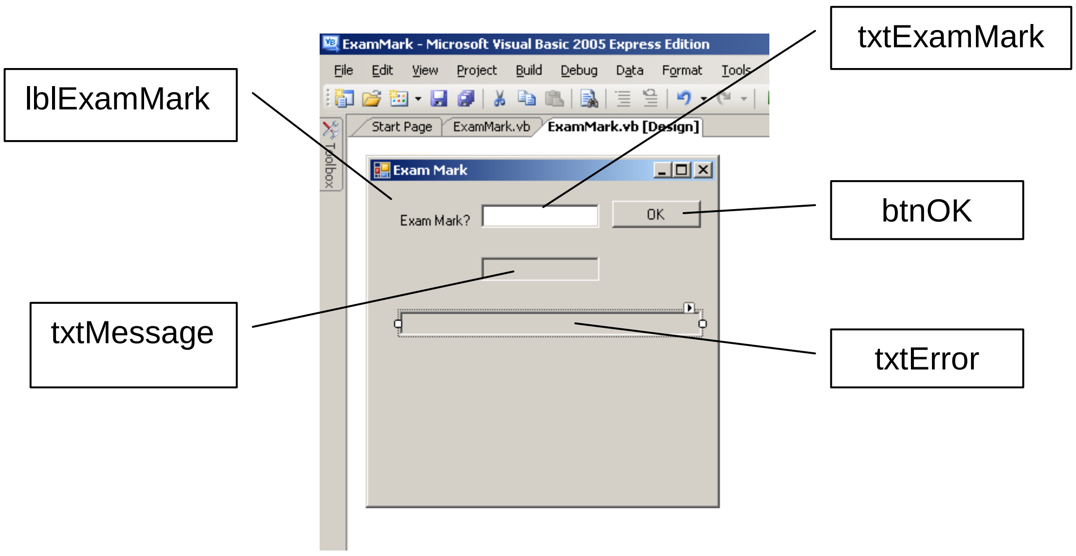
On a number line:



In a program flowchart:



Here is the user interface.



User Interface

1. Start a new project and name it *ExamMark*

2. Set FileName = *ExamMark.vb*

3. Set Form properties: click on an empty part of the form
   a. Name = *frmExamMark*
   b. Text = *Exam Mark*

4. Drag controls onto the form and set their properties as listed below:
   a. Label
      i. Name = *lblExamMark*
      ii. Text = *Exam Mark?*

   c. TextBox
      i. Name = *txtExamMark*

   d. Button
      i. Name = *btnOK*
      ii. Text = *OK*

   e. TextBox
      i. Name = *txtMessage*
      ii. ReadOnly = *True*

   f. TextBox
      i. Name = *txtError*
      ii. ReadOny = *True*

Programming Code

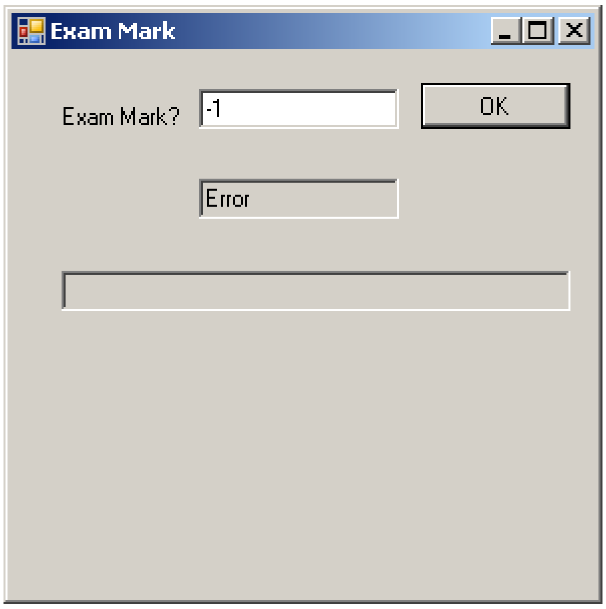
5. Under the OK button:

```
Try
    Dim intExamMark As Integer = Convert.ToInt32(txtExamMark.Text)
    If intExamMark >= 0 And intExamMark <= 39 Then
        txtMessage.Text = "Fail"
    ElseIf intExamMark >= 40 And intExamMark <= 100 Then
        txtMessage.Text = "Pass"
    Else
        txtMessage.Text = "Error"
    End If
    Catch ex As FormatException
        txtErrorMessage.Text = "Number format error"
    Catch ex As OverflowException
        txtErrorMessage.Text = "Number too large error"
 End Try
```

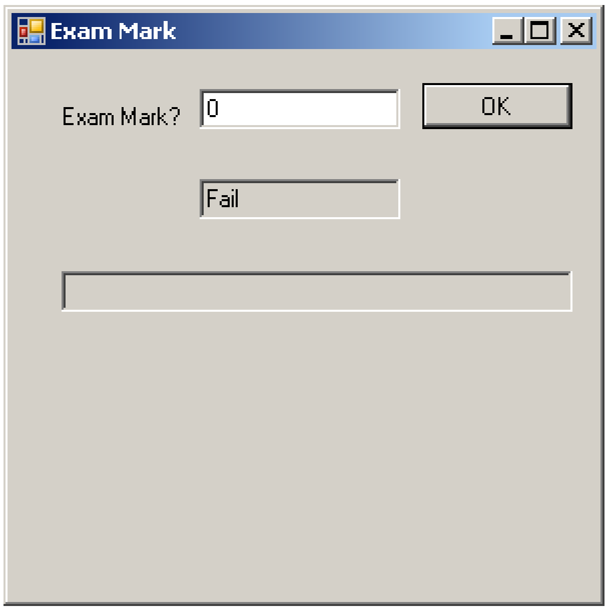
Notice that, in VB, ElseIf is written as one word.

Testing

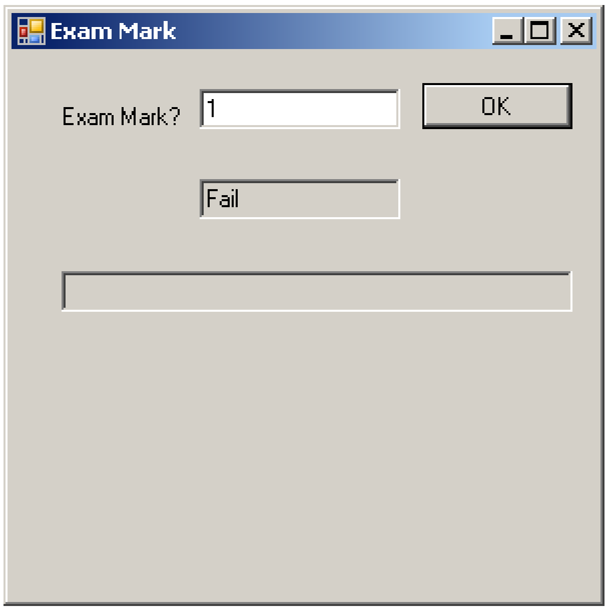
Notice that each boundary and the two exceptions have been tested.



-1
Error



0
Fail

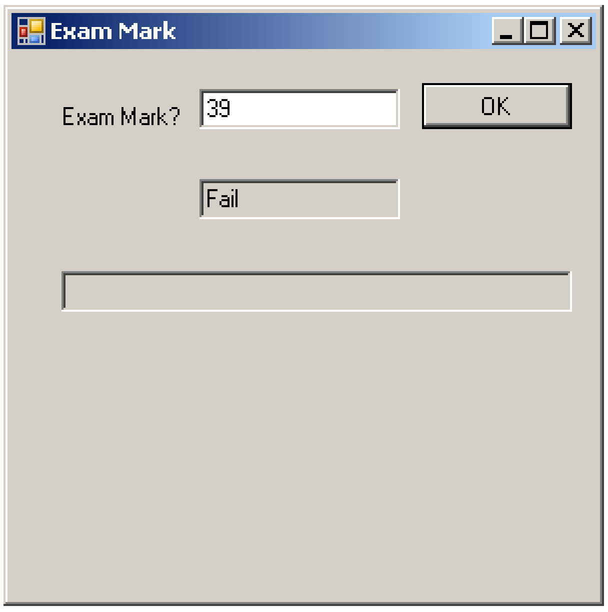


Exam Mark

Exam Mark? 1

OK

Fail

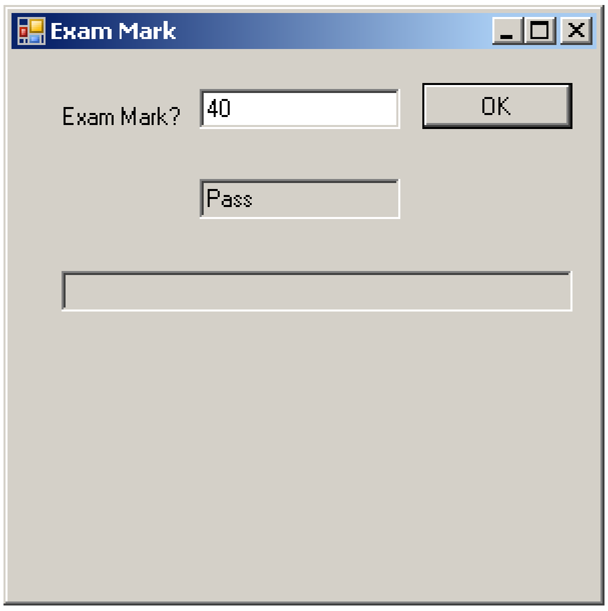


Exam Mark

Exam Mark? 39

OK

Fail

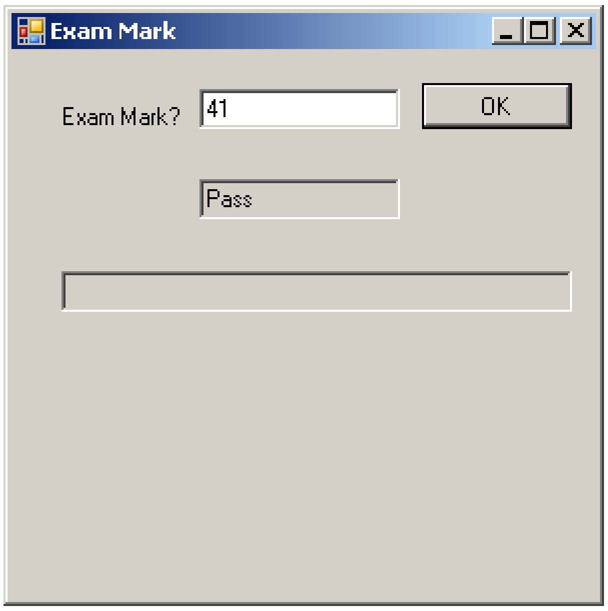


Exam Mark

Exam Mark? 40

OK

Pass

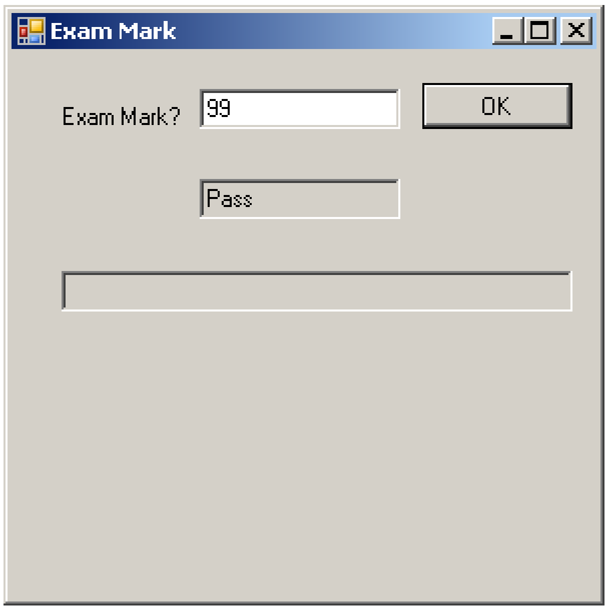


Exam Mark

Exam Mark? 41

OK

Pass

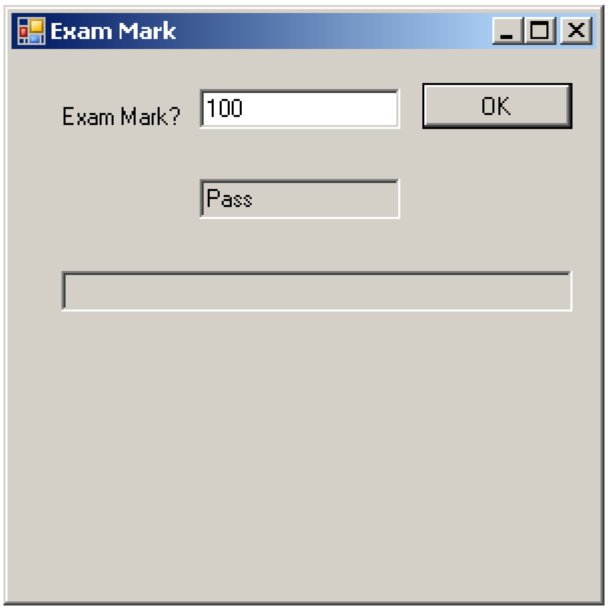


Exam Mark

Exam Mark? 99

OK

Pass

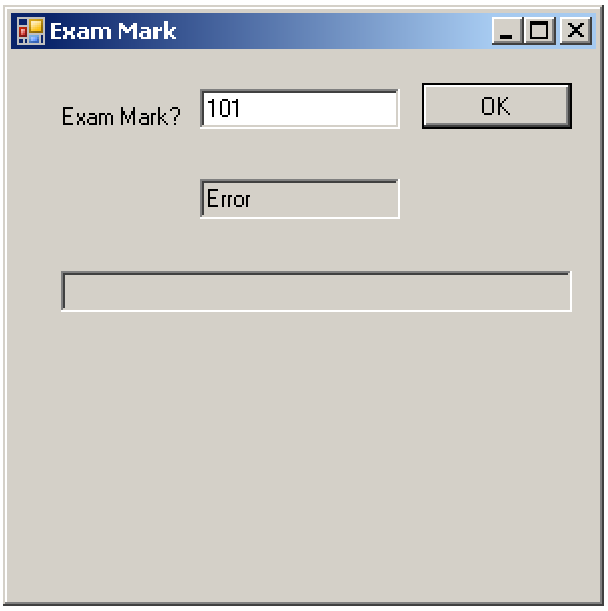


Exam Mark

Exam Mark? 100

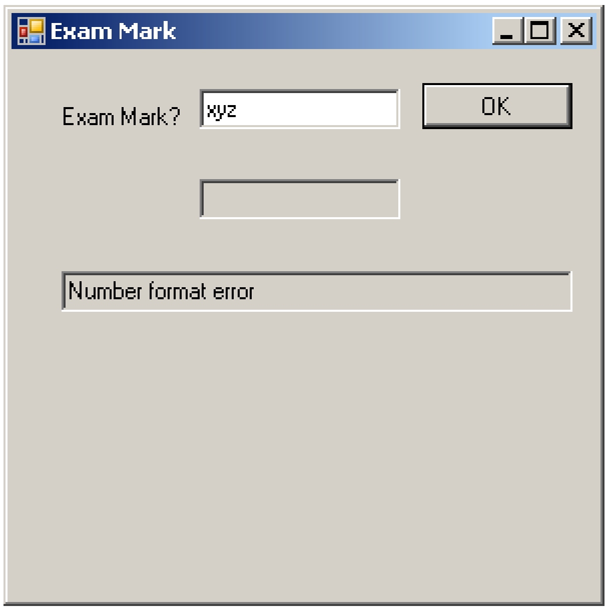
OK

Pass



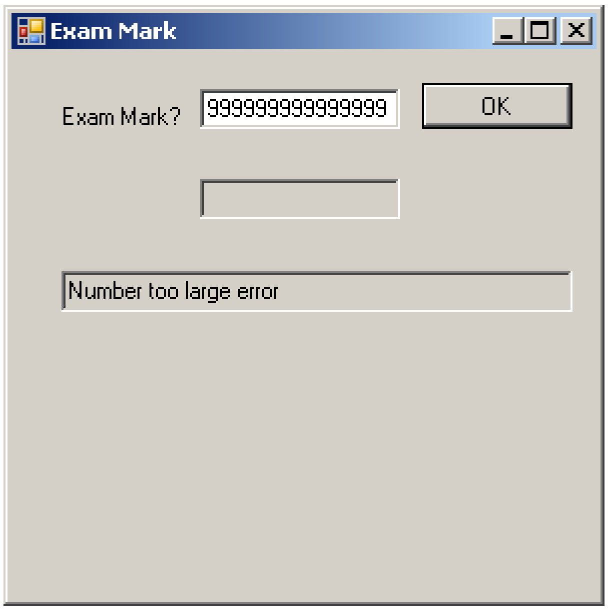
101

Error



xyz

Number format error



999999999999999

Number too large error

Exercise 5.2

1. Try out the Exam Mark program shown above. Remember to include appropriate comments.

2. Design, write, comment and test VB programs to:

   a. read the pollen count. If the pollen count is less than zero, write *error*. If the pollen count is between 0 and 29 inclusive, write *low*. If the pollen count is between 30 and 69 inclusive, write *medium*. If the pollen count is between 70 and 100 inclusive, write *high*. If the pollen count is more than 100, write *very high.*

   b. read the bill. If the bill is less than zero, write error. If the bill is between 0 and 50 inclusive write *no discount*. If the bill is over 50 write *discount is 10%*.

   c. read the bill. If the bill is less than £400 write *delivery charge is £25*. If the bill is more than £400 write *no delivery charge*. If the bill is less than zero *write error*.

   d. read the time taken to cover a ¼ mile from a standing start. If the time is less than 4 seconds write *quick*. If the time is between 4 and 8 seconds inclusive, write *average*. If the time is more than 8 seconds write *slow*. If the time is negative write *Error*.

   e. read the number of steps taken by a marathon runner in one minute. If the number of steps is less than 175 write *speed up*. If the number of steps is more than 225 write *slow down*. If the number of steps is between 175 and 225 inclusive write *write on*. If the number of steps is less than zero or more than 500 write *Error*.

   f. Normal body temperature is 35.7 ± 2.5 i.e. between 33.2 and 38.2 inclusive. Read body temperature. If it is less than 33.2 write *hypothermic*. If it is more than 38.2 write *hyperthermic*. If it is between 33.2 and 38.2 inclusive, write *normal*.

Bibliography

http://catless.ncl.ac.uk/risks accessed 18Sep2008

Next we look at loops.