

Software Design and Development

11 Standards

Chaos is life without standards.

Terry Marris April 2009

A standard is a uniform way of completing a task. We look at the programming standards expected of BTEC First Diploma students at Leicester College. We look at:

1. Identifiers
2. Comments
3. Layout
4. Simplicity

11.1 Descriptive Identifiers

An identifier is a name we chose for programming components such as files, forms, variables and functions. The identifiers must:

1. be descriptive e.g. *txtAge* not *textbox1* - helps the reader to understand the programming code
2. start with a three or four-letter code - helps the reader distinguish between GUI (Graphical User Interface) elements and variables
 - frm for form e.g. frmAge
 - lbl for label e.g. lblAge
 - txt for text box e.g. txtAgeInput
 - btn for button e.g. btnOK
 - int for integer e.g. intAge
 - dbl for double e.g. dblHeight
 - bool for boolean e.g. boolAgelsOK
 - str for string e.g. strName
 - chr for character e.g. chrGender

Usually, the identifier for a function describes the kind of value being returned.

3. have capital letters for the start of every word (except the first word) e.g. strFirstName - helps make identifiers easy to understand
4. not contain hyphens or underscores - they make programming code hard to read

11.2 Comments

Comments are written for the benefit of the person who has to read and understand your programming. That person could be yourself, your manager or your colleague. Comments describe WHAT the program does and WHAT each function and procedure does. Comments also describe WHAT tricky bits of coding do. The first line of a program must be a comment that explains what it does, who wrote it and when. For example:

1.

```
' This program inputs an age and outputs the corresponding category for a
' marathon
' Written by Terry Bull April 2009
```
2.

```
Function categoryFromAge(ByVal intAge As Integer) As String
' returns a runner's category from their given age
  If intAge < 16 Then
    Return "Reject - too young"
  ElseIf intAge >= 16 And intAge <= 35 Then
    Return "Regular"
  ElseIf intAge > 35 Then
    Return "Senior"
  End If
  Return "Unexpected error"
End Function
```
3.

```
' Calculates the number of days required to fill the surface of a 100 square
' metre pond with algae when the algae doubles in size every day
intDays = 0
intAlgae = 1
while intAlgae < 100
  intAlgae = intAlgae * 2
  intDays = intDays + 1
End While
```

11.3 Layout

The Visual Basic development environment does a good job of setting out your coding - usually. Sometimes, it needs your help.

1. insert two blank lines between functions (and procedures) - helps you to see where a function ends and the next one begins
2. indent between:
 - Function and End Function
 - If and Else
 - Else and End If
 - While and End While

See the examples shown in Section 11.2 above. Helps you understand the logic, especially when your programs become large and complicated

11.4 Simplicity

In everything you do ... KISS. Keep it small, sweet, simple and straightforward - makes your programs easy to read, write, understand and get right.

11.5 Standards

Standards - why bother?

- to minimise errors - this is the main aim of all programmers
- to maximise clarity - so that yourself and others can understand what you have written
- to train newcomers - so that they fit in with their programming community

Bibliography - None

Next - Nothing!