C. Escape Sequences and Format Items

Escape Sequences

The following escape sequences are defined in the C# programming language:

- `\'` – Single quote
- `\"` – Double quote
- `\\` – Backslash
- `\0` – Null
- `\a` – Alert (generates a simple beep noise)
- `\b` – Backspace
- `\f` – Form feed
- `\n` – New line
- `\r` – Carriage return
- `\t` – Horizontal tab
- `\v` – Vertical tab
- `{{` – An open brace character (`{`)
- `}}` – A closing brace character (`}`)

Note that for a new line character, you should generally avoid using `\n`. Instead, use `Environment.NewLine`.

Composite Formatting

Composite format strings are used regularly throughout C#/Microsoft.Net such as in the `Console.WriteLine()` function. Composite strings are built using one or more format items, e.g., `{0,15:c}`, which have the following syntax:

$$\{index[,alignment][:formatString]\}$$

The *index* refers to which parameter is used after the composite format string, starting from a zero base, i.e., the first parameter appearing after the composite format string has index `0`, the second has index `1`, and so on. The alignment allows a minimum field width to be specified, either right aligned (positive number) or left aligned (negative number), e.g., a field width specified as `10` is a right aligned field that is a minimum of 10 characters wide, whereas a field width specified as `-8` is a left aligned field that is a minimum of eight characters wide. Finally, the *formatString* changes depending on the data types:

For this unit, we only consider a limited number of formats for numeric types:

- `C` or `c` – formats a numeric type according to local currency rules, e.g., the value `5.5` formatted using `{0:C}` would result in `$5.50`
- `Dnum` or `dnum` – only for integral types, prefixes the value with zeroes up to *num* digits, e.g., the value `55` formatted using `{0:D5}` would result in `00055`
- `E` or `e` – formats a numeric type using scientific/exponential notation, e.g,. the value `12345` formatted using `{0:E}` would result in `1.234500e+004`, which is interpreted as

- `F`*num* or `f`*num* – formats a numeric type using a fixed number of decimal places, where *num* represents the number of decimal places, e.g., the value `5.5` formatted using `{0:F3}` would result in `5.500`
- `P`[*num*] or `p`[*num*] – formats a floating point value as a percentage value, and optionally specifies the number of decimal places (specified as *num*), e.g., `0.123` formatted using `{0:P3}` would result in `12.300 %`

More information regarding formatting numeric types can be found at:

http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx