

## Fmtflt for C++

```
void fmtflt( double value_in , int ipad_in )
//      ASTEM97 IAPWS INDUSTRIAL FORMULATION
//      FOR WATER AND STEAM IF-97
//      Edward D. Throm (C) 2002
//      E-mail : mister-ed@cox.net
//      http://members.tripod.com/~Mister_Ed
// For I/O streams and output format
// Detemines type of output fixed or scientific and precision
// Incoming I/O stream is saved and restored after output
// Input
//      value_in floating point value to convert to formatted output
//      ipad_in if = 1 then set output length to 13 characters Sm.nnnnnESxxx
//      else format value_in to 6-digits
// Output  cout << value_in;      //in selected format/precision
{
    int ians,iwidth;
    double vcheck;

// Start Save I/O settings
    int current_settings;
    int current_precision;
    current_settings = cout.flags();
    current_precision = cout.precision();
// End   Save I/O settings

// Get absolute value on value_in (avoid need for #include "math" )
    vcheck = value_in;
    if ( vcheck < 0 )
        vcheck = -vcheck;

// Special case for zero,one or routine return value of -1.0
    if ( vcheck == 0.0 || vcheck == 1.0 )
        ians = 1;

// Scientific, set precision to 5
    else if ( ( vcheck < 1.0e-3 ) || ( vcheck > 1.0e4 ) )
        ians = 0;

// Else, fixed with precision
    else if ( vcheck < 1.0 )
        ians = 6;
    else if ( vcheck < 10.0 )
        ians = 5;
    else if ( vcheck < 100.0 )
        ians = 4;
    else if ( vcheck < 1000.0 )
        ians = 3;
    else
        ians = 2;
```

```

// Determine output width
if (ipad_in != 1)
    switch( ians )
    {
        case 0:
            iwidth = 13;
            break;
        case 1:
            iwidth = 4;
            break;
        case 2:
        case 3:
        case 4:
        case 5:
            iwidth = 8;
            break;
        case 6:
            iwidth = 9;
            break;
    }
    else
        iwidth = 13;      // Supports max length Sm.nnnnnESxxx

// Select format style, scientific or fixed
if (ians == 0)
    cout.setf(ios::scientific, ios::floatfield);
else
    cout.setf(ios::fixed      , ios::floatfield);
if (ians == 0 ) ians = 5;

// cout << valin

    cout << setprecision ( ians ) << setw( iwidth );
    cout << value_in;

// Reset I/O settings
cout.flags(current_settings);
cout.precision(current_precision);
//
}

```

## fmtflt for Delphi

```
Function fmtflt( value_in: Double ; ipad_in: Integer ): String;
```

```
{
    ASTEM97 IAPWS INDUSTRIAL FORMULATION
    FOR WATER AND STEAM IF-97
    Edward D. Throm (C) 2002
    E-mail : mister-ed@cox.net
    http://members.tripod.com/~Mister_Ed
```

```
-----
For I/O streams and output format
Determines type of output fixed or scientific and precision
Incoming I/O stream is saved and restored after output
Input
    value_in floating point value to convert to formatted output
    ipad_in if = 1 then set output length to 12 characters Sm.nnnnnESxx
            else format value_in to 6-digits
Output string fmtflt //in selected format/precision
```

```
-----
}
```

```
var len,kct,k: Integer; { used to equalize output line }
var vcheck: double; { absolute value of value_in }
var str_tmp1: string; { holds value_in in string text }
var str_tmp2: string;
begin
```

```
str_tmp1 := '';
{ Get absolute value on value_in }
vcheck := value_in;
If ( vcheck < 0.0 ) Then
    vcheck := -vcheck;

{ Special case for zero,one or routine return value of -1.0 }
If ( (vcheck = 0.0) Or (vcheck = 1.0) ) Then
str_tmp2 := formatfloat('0.0',value_in)
Else
```

```
{ Scientific, sets precision to 5 }
If ( vcheck <= 1.0e-3 ) Then
str_tmp2 := formatfloat('0.00000E+00',value_in)
Else
```

```
{ Else, fixed with precision }
If ( vcheck < 1.0 ) Then
str_tmp2 := formatfloat('0.000000',value_in)
Else
If( vcheck < 10. ) Then
str_tmp2 := formatfloat('0.00000',value_in)
Else
If( vcheck < 100. ) Then
str_tmp2 := formatfloat('0.0000',value_in)
Else
If( vcheck < 1000. ) Then
str_tmp2 := formatfloat('0.000',value_in)
Else
If( vcheck < 10000. ) Then
str_tmp2 := formatfloat('0.00',value_in)
```

```
Else
str_tmp2 := Formatfloat('0.00000E+00',value_in);

{ make value width 12 characters if ipad_in is 1 (length Sm.nnnnnESee)  }
If ( ipad_in = 1 ) Then
begin
len := 12;
kct := len - StrLen(PChar(str_tmp2));
  for k := 1 to kct do
    begin
      str_tmp1 := str_tmp1 + ' ';
    end;
  end;

str_tmp1 := str_tmp1 + str_tmp2;
fmtflt := str_tmp1;
end;
```

## fmtflt for Visual Basic

```
Function fmtflt(ByVal value_in As Double, ByVal ipad_in As Integer) As String
' ASTEM97 IAPWS INDUSTRIAL FORMULATION
'   FOR WATER AND STEAM IF-97
'   Edward D. Throm (C) 2002
' For I/O streams and output format
' Detemines type of output fixed or scientific and precision
' Incoming I/O stream is saved and restored after output
' Input
'   value_in floating point value to convert to formatted output
'   ipad_in if = 1 then set output length to 12 characters Sm.nnnnnESxx
'   else format valin to 6-digits
' Output string fmtflt //in selected format/precision
'
Dim ken As Integer ' used to equalize output line
Dim kct As Integer
Dim k As Integer
Dim vcheck As Double ' absolute value of value_in
Dim str_tmp1 As String ' holds value_in in string text
Dim str_tmp2 As String

str_tmp1 = ""
' Get absolute value on value_in
vcheck = value_in
If (vcheck < 0#) Then vcheck = -vcheck

' Special case for zero,one or routine return value of -1.0
If (vcheck = 0# Or vcheck = 1) Then
    str_tmp2 = Format$(value_in, "0.0" )

' Scientific, sets precision to 5
ElseIf vcheck <= 0.001 Then
    str_tmp2 = Format$(value_in, "0.00000E+00")

' Else, fixed with precision
ElseIf vcheck < 1# Then
    str_tmp2 = Format$(value_in, "0.000000" )
ElseIf vcheck < 10# Then
    str_tmp2 = Format$(value_in, "0.00000" )
ElseIf vcheck < 100# Then
    str_tmp2 = Format$(value_in, "0.0000" )
ElseIf vcheck < 1000# Then
    str_tmp2 = Format$(value_in, "0.000" )
ElseIf vcheck < 10000# Then
    str_tmp2 = Format$(value_in, "0.00" )
Else: str_tmp2 = Format$(value_in, "0.00000E+00")
End If

' make value width 12 characters if ipad_in is 1 (length Sm.nnnnnESxx)
If ipad_in = 1 Then
ken = 12
kct = ken - Len(str_tmp2)
    For k = 1 To kct
        str_tmp1 = str_tmp1 + " "
    Next
End If

str_tmp1 = str_tmp1 + str_tmp2
fmtflt = str_tmp1
End Function
```

## Examples

```
Print text    "The value of v is "  
Print value   spvl  
Print text    " m^3/kg."
```

### **Fmtflt for C++**

Without fmtflt

```
cout << "The value of v is " << spvl << cout << " m^3/kg." << endl;  
displays: The value of v is 0.00106111 in m^3/kg.
```

With fmtflt

```
cout << "The value of v is "; fmtflt(spvl,0); cout << " m^3/kg." << endl;  
displays: The value of v is 0.001061 m^3/kg.  
Example:
```

### **fmtflt for Delphi**

Without fmtflt

```
strval := 'The value of v is ' + FloatToStr(spvl) + ' m^3/kg.';  
displays: The value of v is 0.00106110666753808 m^3/kg.
```

With fmtflt

```
strval := 'The value of v is ' + fmtflt(spvl,0) + ' m^3/kg.';  
displays: The value of v is 0.001061 m^3/kg.
```

### **fmtflt for Visual Basic**

Without fmtflt

```
strval = 'The value of v is ' + Str(spvl) + ' m^3/kg.'  
displays: The value of v is 1.06110666753808E-03 m^3/kg.
```

With fmtflt

```
strval = 'The value of v is ' + fmtflt(spvl,0) + ' m^3/kg.'  
displays: The value of v is 0.001061 m^3/kg.
```