Table of Contents

Variable Number Function Parameters

- Specifying functions that accept an unknown number of arguments.
- Ellipsis " . . . " in a function prototype indicates the function accepts a arbitrary number of parameters.

```
int va_total ( int , ... ) ;
```

At least one named parameter must be specified.

The ellipsis must be placed at the end of the parameter list.

- The predefined types that support variable length parameter lists is defined in "stdarg.h". Contains one type definition and three macro functions.
- stdarg.h declarations:

```
va_list  Predefined type for storing the variable
         length argument list.

va_start Macro function called to obtain the variable
         length argument list.

va_arg   Macro function that returns a value in the
         argument list.
va_end   Macro function called to perform necessary
         (memory) cleanup before function exits.
```

Macro Function Arguments

Variable parameters macro function arguments:

- void va_start(va_list, int);
  - The first argument must be of type va_list to hold the variable number of parameters.
  - The second argument should be of type int and must be the last parameter in the variable-length function heading, (i.e., immediately preceding the ellipsis. When called it holds the count of the number of arguments.

- *type* va_arg(va_list, *type*);
  - The first argument must be of type va_list holding the variable number of parameters list previously returned by a call to va_start().
  - The second argument must be a standard C/C++ predefined language type.
  - The call returns the next argument in the list, converted to the passed type.

- void va_end (va_list);
  - The first argument must be of type va_list holding the variable number of parameters list previously returned by a call to va_start().

Variable Arguments Example

Total Function with Variable Arguments

```
#include <stdarg.h>

double va_total( int va_c, . . . )
{
  va_list  va_v;
  double sum=0.0;

  va_start(va_v, va_c);

  for (int i=0; i < va_c; i++)
        sum += va_arg(va_v, double);

  va_end(va_v);

  return( sum );
}
```

Variable Arguments Total Function calls:

```
double pi=3.14159, e=2.71828, total;
int args=2;

  total = va_total(args, pi, e);

  total = va_total(3, -1.0, -2.0, -3.0);
```