Variable Length Argument Lists

Variable Length Argument Lists

- Must always have at least one argument of a fixed type
- May have 0 or more additional arguments of any type
- Fixed arguments must precede variable arguments

Variable Length Argument Lists

- Procedures must have an intrinsic way of determining the type of each argument, and finding the end of the argument list
- Prototypes use to represent variable portion of argument list
  - char *printf(char *format, ...);
  - char *sprintf(char *buff, char *format, ...);
  - void print_integers(int count, ...);

Variable Length Argument Lists

- Begin processing a variable argument list with the *va_start* macro
- Arguments are a state variable, and the last fixed parameter

```
void print_double_list( int count, ... )
{
    va_list argp;
    double  value = 0;
    va_start( argp, count );
```

- Continue processing a vararg list with the *va_arg* macro; the arguments are the state variable initialized by *va_start*, and the type of the next parameter

```
    while ( count-- > 0 )
    {
      value = va_arg( argp, double );
      printf( "%d\n", value );
    }
```

- When processing is complete, cleanup must be performed using the *va_end* macro

```
    va_end( argp );
}
```

Variable Length Argument Lists

- http://faculty.washington.edu/sproedp/advc/csamples/less23.c.html