

GE3 COMPUTER SCIENCE

C AND C ++ LECTURE SERIES *FOR*

B.SC 3RD SEMESTER *BY*

SUBHADIP MUKHERJEE

DEPARTMENT OF COMPUTER SCIENCE

KHARAGPUR COLLEGE

LECTURE 8



FUNCTIONS

DEFINITION

A **function** is a self-contained program segment that carries out some specific, well-defined task.

```
data-type name(type 1 arg 1, type 2 arg 2, . . . , type n arg n)
```

Formal Argument

Actual Argument

```
char lower_to_upper(char c1)      /* programmer-defined conversion function */  
{  
    char c2;  
  
    c2 = (c1 >= 'a' && c1 <= 'z') ? ('A' + c1 - 'a') : c1;  
    return(c2);  
}
```

FUNCTIONS

Lowercase to Uppercase Conversion

```
#include <stdio.h>

char lower_to_upper(char c1)      /* function definition */
{
    char c2;

    c2 = (c1 >= 'a' && c1 <= 'z') ? ('A' + c1 - 'a') : c1;
    return(c2);
}

main()
{
    char lower, upper;

    printf("Please enter a lowercase character: ");
    scanf("%c", &lower);
    upper = lower_to_upper(lower);
    printf("\nThe uppercase equivalent is %c\n\n", upper);
}
```

FUNCTIONS

FACTORIAL

The *factorial* of a positive integer quantity, n , is defined as $n! = 1 \times 2 \times 3 \times \cdots \times n$.

$$4! = 1 \times 2 \times 3 \times 4 = 24;$$

```
long int factorial(int n)      /* calculate the factorial of n */
{
    int i;
    long int prod = 1;

    if (n > 1)
        for (i = 2; i <= n; ++i)
            prod *= i;
    return(prod);
}
```

FUNCTIONS

MINIMUM OF TWO NUMBERS

```
void maximum(x, y)
{
    int z;

    z = (x >= y) ? x : y;
    printf("\n\nMaximum value = %d", z);
    return;
}
```

FUNCTIONS

ACCESSING A FUNCTION

If the function returns a value, the function access is often written as an assignment statement; e.g.,

```
y = polynomial(x);
```

On the other hand, if the function does not return anything, the function access appears by itself; e.g.,

```
display(a, b, c);
```

FUNCTIONS

FUNCTION PROTOTYPE

Function prototypes are usually written at the beginning of a program, ahead of any programmer-defined functions (including **main**).

```
data-type name(type 1 arg 1, type 2 arg 2, . . ., type n arg n);
```

FUNCTIONS

FUNCTION PROTOTYPE (Cont.)

```
#include <stdio.h>

long int factorial(int n)
/* calculate the factorial of n */
{
    int i;
    long int prod = 1;
    if (n > 1)
        for (i = 2; i <= n; ++i)
            prod *= i;
    return(prod);
}
```

```
main()
{
    int n;
    /* read in the integer quantity */
    printf("\nn = ");
    scanf("%d", &n);
    /* calculate and display the factorial */
    printf("\nn! = %ld", factorial(n));
}
```


COMPILE AND RUN A C CODE

Thank You

End of Lecture 8

Subhadip Mukherjee

Department of Computer Science

Kharagpur College

Kharagpur, India

