

ECE 220: Computer Systems & Programming

Lecture 7: Functions in C Thomas Moon

February 8, 2024



Previous Lecture

- Control Structure in C
 - `if`
 - `if-else`
 - `switch`

 - `while`
 - `for`
 - `do-while`

- `break` vs `continue`

Functions in C

- Roughly equivalent to a subroutine in LC-3 assembly language
- A segment of code that implements well-defined function
- Using functions enables
 - Hiding low-level details
 - Giving high-level structure to the program
 - Efficiently reusing code

Anatomy of a C function

```
#include <stdio.h>
```

```
int Fact(int n);
```

Function Prototype (declaration)

1. Name of function

2. Type of return value

3. Types of parameters

$$f(n) = n! = n \cdot (n - 1) \dots \cdot 1$$

Purpose of function prototype?

→ *informs the compiler about the properties of function*

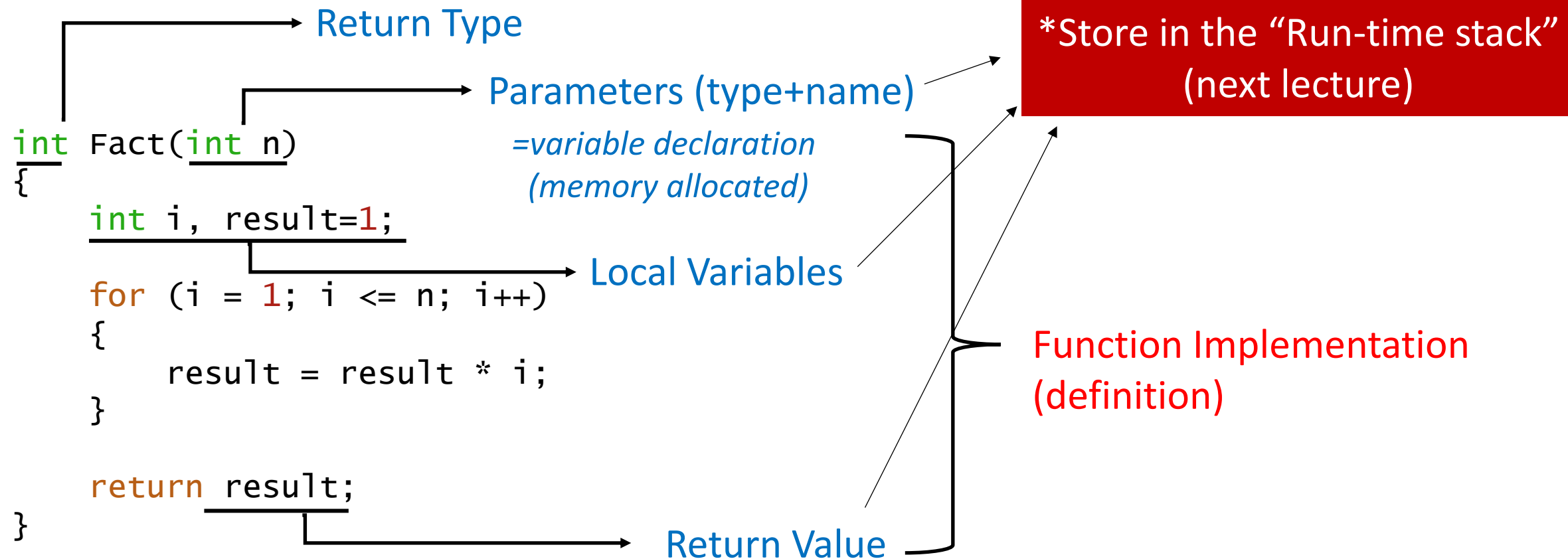
```
int main()  
{  
    int number;  
    int answer;  
    printf("Enter a number: ");  
    scanf("%d", &number);
```

```
    answer = Fact(number);  
    printf("factorial of %d is %d\n", number, answer);  
    return 0;  
}
```

Function Call

arguments (actual value of parameters)

Anatomy of a C function (continued)



**Is there other way to implement a factorial function?*

$$f(n) = n! = n \cdot (n - 1) \dots \cdot 1 = n \cdot f(n - 1)$$

***Recursion (Lec14,15)**

Number of Parameters & Return Value

- A function can have multiple parameters or none.
- A function can return at most ONE return value.

```
double add(double a, double b){  
    return a+b;  
}  
int zero(void){  
    return 0;  
}  
void printA(){  
    printf("A\n");  
}
```

→ function call should be zero();



```
double, double add(double a, double b){  
    return a+b, a-b;  
}
```

Separate Files

main.c

```
#include <stdio.h>
#include "mymath.h"

int main(){
    printf("%d", add(1,2));
}
```

mymath.h

```
int add(int, int);
int subtract(int, int);
```

mymath.c

```
#include "mymath.h"
int add(int a, int b){
    return a + b;
}
int subtract(int a, int b){
    return a - b;
}
```

To compile multiple files,

```
gcc main.c mymath.c
```

Some Useful Libraries

- `stdio.h`

- `printf`, `scanf`, `getchar`, `putchar`
- `fprintf`, `fscanf`

- `math.h`

- `cos`, `sin`
- `exp`, `log`, `log10`, `pow`
- `ceil`, `floor`, `round`
- `M_PI`, `M_E`

```
**To compile with math.h,  
gcc quad.c -lm
```

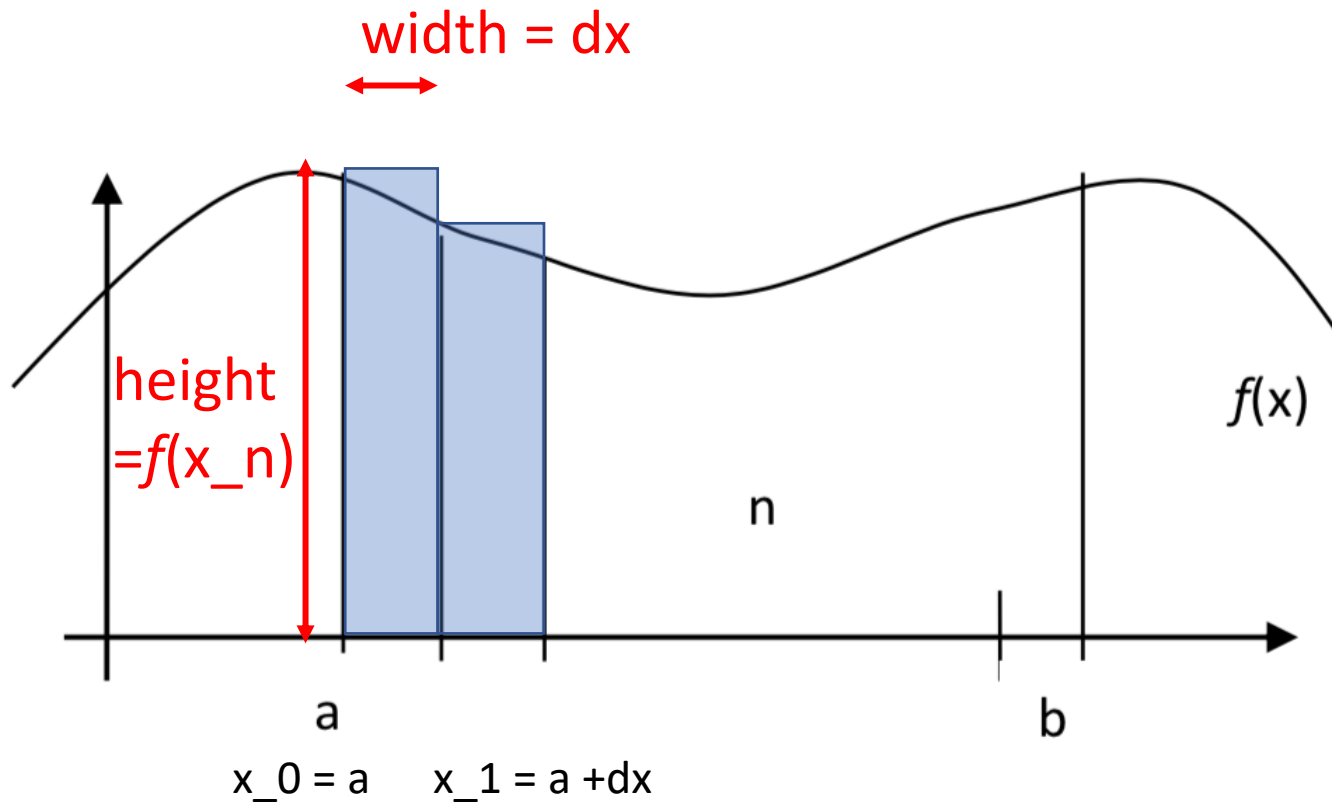
- `ctype.h`

- `isalpha`, `isupper`, `islower`, `toupper`, `tolower`

- `stdlib.h`

- `rand`

Example1: Riemann Integral



Given function $f(x)$,

- Input
 - interval $[a,b]$
 - n : number of sub-rectangular
- Output
 - computed integral value

```
#include <stdio.h>
#include <math.h>

double f(double x);

int main(){
    double a,b;
    int n;

    printf("Enter a,b,n: ");
    scanf("%lf %lf %d", &a, &b, &n);
    printf("Result is %lf\n", ??? );
}

double f(double x){
    return cos(x); // any function
    //return x;
}
```