CSE 113 Structured Programming Language

Function

Tasnim Zahan
Assistant Professor
Dept. of CSE
NEUB

Function introduction

A function is a set of statements that take inputs, do some specific computation and produces output

Syntax:

```
Return_type Function_name (argument/parameters)
{
    Local variable declaration;
    statement1;
    statement2;
    . . .
    return (value)
}
```

Example

```c
#include <stdio.h>
#define PI 3.1416

float area_of_circle(float r)
{
    float area = PI*r*r;
    return area;
}

int main(void)
{
    float rad, area;

    printf("Enter radius: ");
    scanf("%f", &rad);
    area = area_of_circle(rad);

    printf("area = %0.2f", area);

    return 0;
}
```

Example

```c
#include <stdio.h>

int max(int x, int y)
{
    if (x > y)
      return x;

    else
      return y;
}

int main(void)
{
    int a = 10, b = 20;

    int m = max(a, b);

    printf("max value %d", m);

    return 0;
}
```

Function Parameters

- Call by value: In this method, values of actual parameters are copied to function’s formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of caller.

- Call by reference: Both actual and formal parameters refer to same locations, so any changes made inside the function are actually reflected in actual parameters of caller

The parameters passed to function are called ***actual parameters***

The parameters received by function are called ***formal parameters***

Example

```c
#include <stdio.h>

void modify(int v) {
    printf("modify 1: %d\n", v);
    v = 42;
    printf("modify 2: %d\n", v);
}

int main() {
    int v = 0;

    printf("main 1: %d\n", v);
    modify(v);
    printf("main 2: %d\n", v);

    return 0;
}
```

Examples (Call by value)

```
long factorial(long n) {
    long i, fact=1;
    for(i=1; i<=n; i++){
        fact *= i;
    }
    return fact;
}
int main(){
    long n, result;

    printf("Enter a number: ");
    scanf("%ld", &n);

    result = factorial(n);
    printf("factorial of %ld is %ld", n, result);

    return 0;
}
```

Array to function (Call by reference)

```c
#include<stdio.h>
int arr_max(int data[], int sz);
int arr_min(int data[], int sz);

int main(){
    int arr[100], n, x, i, min, max;
    printf("Enter array size: ");
    scanf("%d", &n);

    printf("Enter %d values:\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);

    max = arr_max(arr, n);
    min = arr_min(arr, n);
    printf("max: %d, min: %d", max, min);
    return 0;
}
```

```c
int arr_max(int data[], int sz){
    int i, max = data[0];
    for(i=1; i<sz; i++){
        if(data[i]>max)
            max = data[i];
    }
    return max;
}

int arr_min(int data[], int sz){
    int i, min = data[0];
    for(i=1; i<sz; i++){
        if(data[i]< min)
            min = data[i];
    }
    return min;
}
```