Attendance record

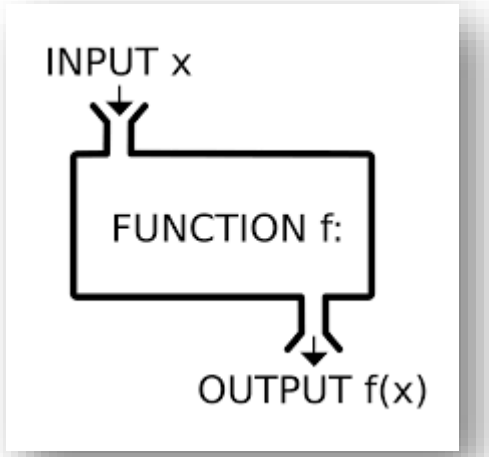
- To check what devices students are using to join class
- To check attendance

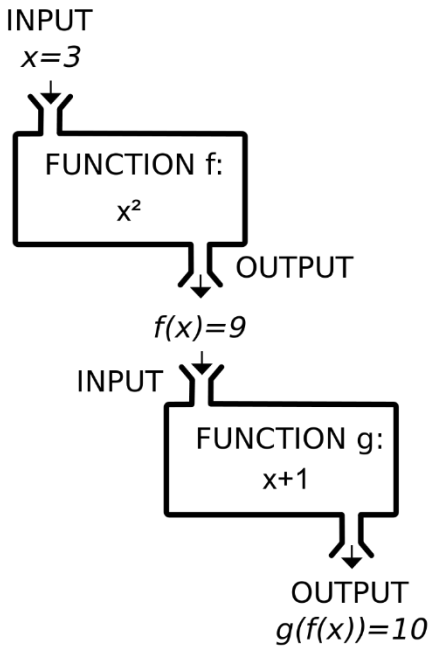
https://forms.gle/dBgZzTML3ZRQGS7Q9

Start class: **3:05pm**

DATA STRUCTURE AND PROGRAMMING

Chapter 5- **Sub-Program (Function)**





Prepared by: Mr. VALY Dona and Mr. NOU Sotheany

Updated and Lectured by: Mr. BOU Channa

Lecture overview

❑ Overall lectures

1. Introduction to algorithm
2. Basic data types and statements
3. Control structures and Loop
4. Array
5. ***Sub-program***
6. Data structure

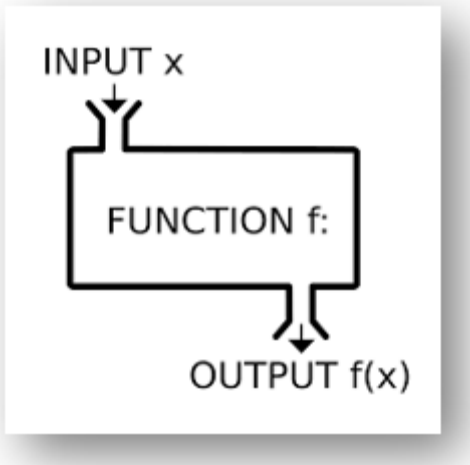
Objective

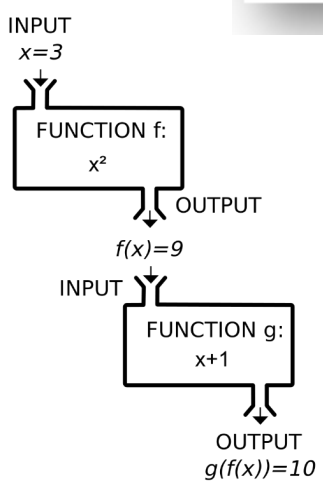
- Introduction to subprogram / function
- Advantages of using subprogram
- How to create your own function
- How to use your own function

Sub-program

❑ Introduction

- **Sub-program** is a part of feature/functionality in a program.
- Sub-program is a block of codes to perform particular task.
- When we start writing a larger program, it becomes
  - Difficult to have a global vision on its functionalities
  - Difficult to determine the errors
- **Solution**: Divide the problem to sub-problem
  - Solve each sub-problem
  - Put those sub-problems in sub-programs





Advantages

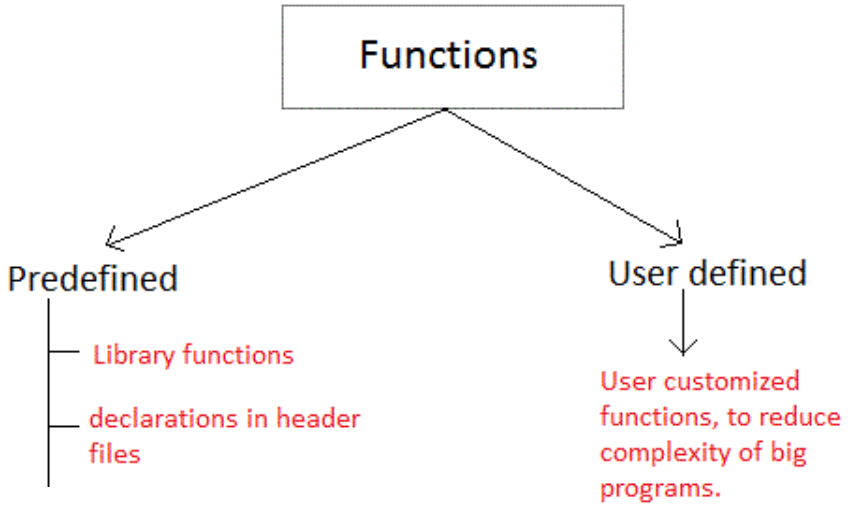
❑ Advantages of using sub-program

- Clear and readable code
- **Reusability**
  - A sub-program can be used many time with the same instruction without rewriting it by just calling its name once defined.
- Easy to test and find error
- Helpful for team work
  - E.g: Each team member is assigned to work on a specific feature which can be implemented as a sub-program

Predefined subprogram

❑ Predefined sub-program (existing/built-in sub-program)

- There are predefined sub-programs
  - E.g: `strcmp(), strlen(), ……`



- Those predefined sub-program **are not enough**
  - so we need to define our own sub-program to solve our problem based on our needs

Sub-program

❑ Type of sub-program

- **Sub-program** is a small program that can be executed in the other program
- There are two types of sub-programs:
  1. Function : used to calculate something and **has return value**
  2. Procedure : a set of commands executed in order and it **has no return value**

Sub-program

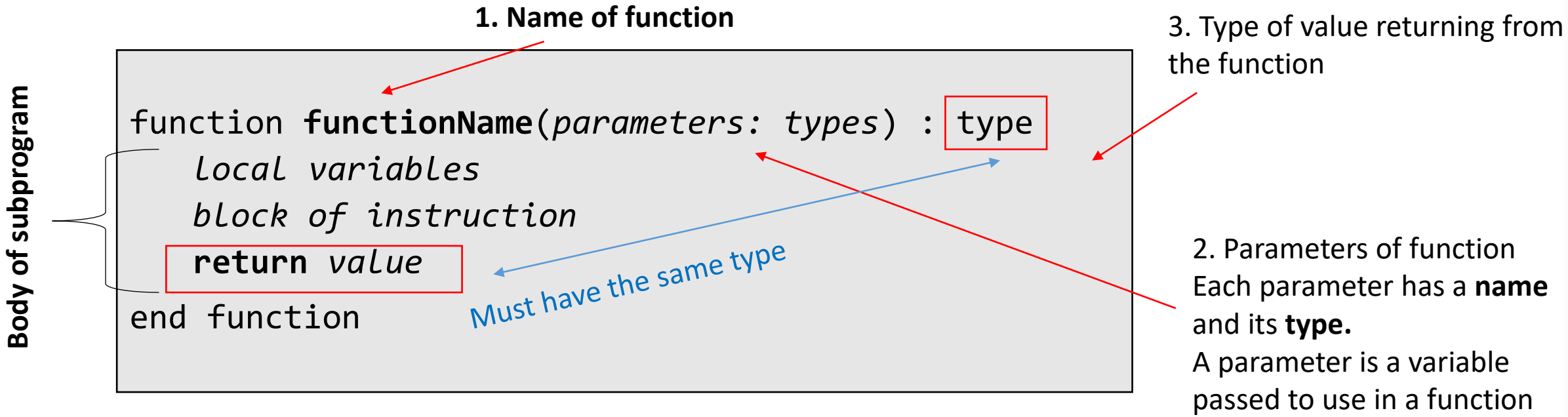
❑ Function: DEFINTION

- A function is a set of instructions grouped under a name, that is called a sub-program and will return a value.
- A function is a sub-program that:
  - ✓ Has a name
  - ✓ Have none, one or many parameters (arguments)
  - ✓ **Return a value** in a certain type
  - ✓ Can have variables inside functions

  (that variables are called *local variable*)
  - ✓ Composed of instructions/codes (body of function)

Sub-program

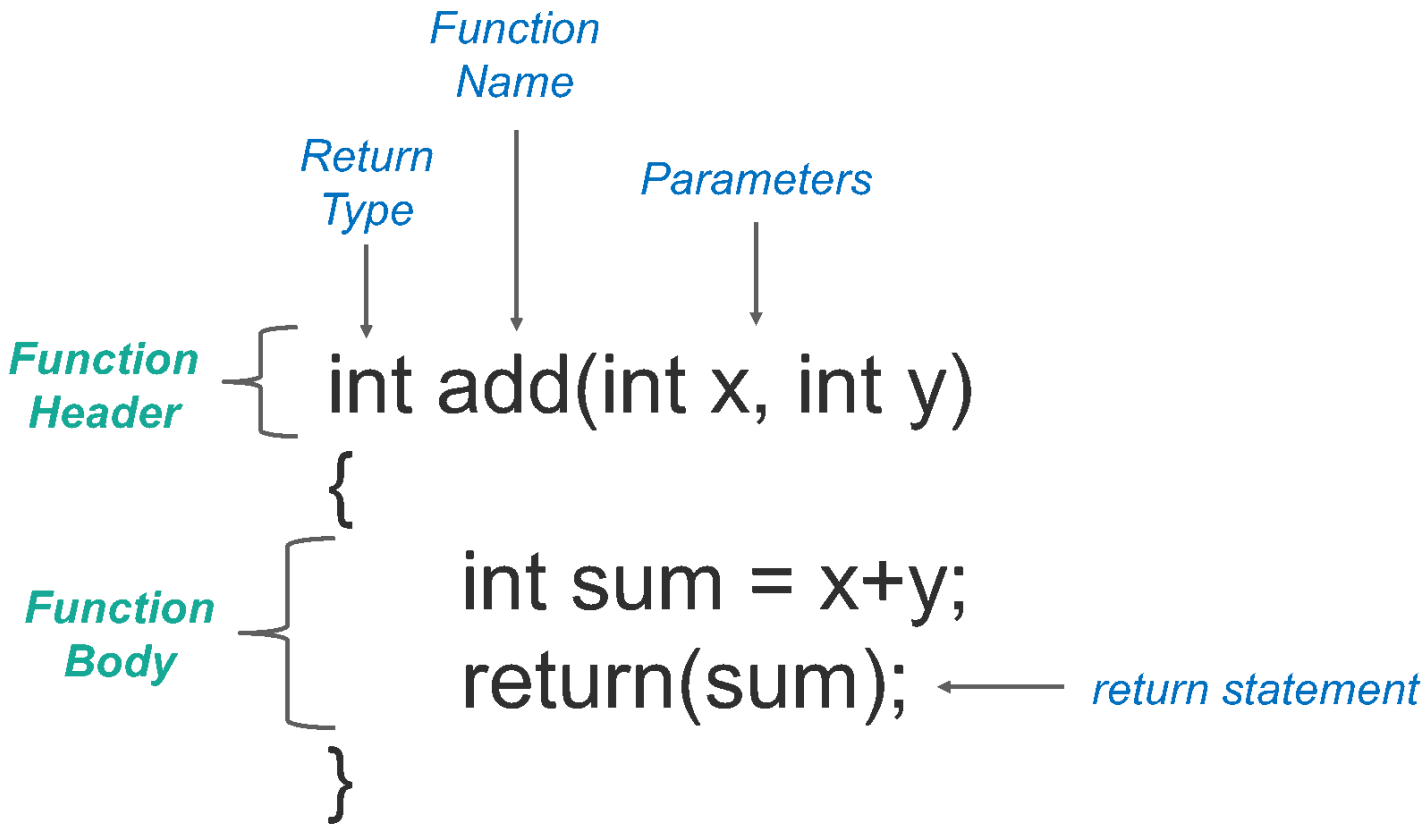
❑ Function: SYNTAX



- Parameters: declaration parameters of function
  - Order of parameter is important
  - A function can have no parameter, one or many parameters (if more than one, separate them by comma)
- Type of return value must be the same with type of function
- ***return*** is an instruction for sending a value from function to where this function is called.

Function

❑ Syntax in C programming



Sub-program

❑ Procedure: SYNTAX

```
Procedure procedureName(parameters: types)
    … local variables …
    … block of instruction …
end procedure
```

Example 1

- A program to get max btw two numbers **without using subprogram**

```
Var a1, b1, a2, b2, c1, c2 : Integer
Var maxa, maxb, maxc : Integer
Begin
    read(a1, b1)
    read(a2, b2)
    read(c1, c2)

    if (a1>a2) then
        maxa ← a1
    else
        maxa ← a2
    end if
    if (b1>b2) then
        maxb ← b1
    else
        maxb ← b2
    end if
    if (c1>c2) then
        maxc ← c1
    else
        maxc ← c2
    end if
    Write(maxa, maxb, maxc)
End
```

Main program without sub-program

Example 1

❑ Using subprogram

Name of sub-program

Sub-program

```
function max(x: integer, y: integer): integer
    var res : integer
    if (x>y) then
        res ← x
    else
        res ← y
    end if
    return res
end function
```

```
Var a1, b1, a2, b2, c1, c2 : integer
Var maxa, maxb, maxc : Integer
Begin
    read(a1,b1)
    read(a2,b2)
    read(c1,c2)
    maxa ← max(a1, a2)
    maxb ← max(b1, b2)
    maxc ← max(c1, c2)
    write(maxa, maxb, maxc)
End
```

Main program

Sub-program

❑ Example 2

Create a function to calculate addition of two integer

```
function add(a:integer, b:integer) : integer
    Var r: integer
    r ← (a + b)*2
    return r
end function
```

Use the created function in a main program

```
var x, y, z: integer
begin
    z ← add(3,5)
    write(z)   16

    read(x,y)          2,3
    z ← add(x,y)       ?? 10
    z ← z - add(x,y)   ?? 0
    z ← add(1, add(2,3))  ?? 22
    z ← add(1, z-1))  ??? 44
end
```

Advantages

❑ Advantages of using sub-program

- Clear and readable
- **Reusability**
  - A sub-program can be used many time with the same the instruction without rewriting it
- Easy to test and find error
- Helpful for team work
  - E.g: Each team member is assigned to work on a specific feature which can be implemented as a sub-program

Predefined subprogram

❑ Predefined sub-program (existing/built-in sub-program)

- There are predefined sub-programs
  - E.g: `strcmp(), strlen(), ……`
- Those predefined sub-program **are not enough**
  - so we need to define our own sub-program to solve our problem based on our needs

C Programming

Function syntax

C program

Defining a Function

The general form of a function definition in C programming language is as follows

```
return_type function_name( parameter list ) {
   body of the function
}
```

Components of function:

- ✓ Has a name
- ✓ Have none, one or many parameters (arguments)
- ✓ **Return a value** in a certain type
- ✓ Can have variables inside functions

(that variables are called *local variable*)

- ✓ Composed of instructions/codes (body of function)

```
/* function returning the max between two numbers */
int max(int num1, int num2) {

   /* local variable declaration */
   int result;

   if (num1 > num2)
      result = num1;
   else
      result = num2;

   return result;
}
```

An example of function in C programming

C program

❑ Examples

```c
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    ... .. ...
}
```

Create function as prototype

```c
#include<stdio.h>

//returnType functionName(parameters){
//}
int add(int a, int b){
    //return (a+b)*2;
    int res;
    res=(a+b)*2;

    return res;
}

int main(){
    int z;
    int x,y;

    z=add(3,5);
    printf("%d\n",z);

    printf("Enter x and y separated by a space: ");
    scanf("%d %d", &x, &y);

    z = add(x,y);
    printf("%d\n",z);
    z =  z - add(x,y);
    printf("%d\n",z);
    z =  add(1, add(2,3));
    printf("%d\n",z);
}
```

Create a function and call it in main

C program

❑ Examples

```c
#include <stdio.h>

/* function declaration */
int max(int num1, int num2);

int main () {

   /* local variable definition */
   int a = 100;
   int b = 200;
   int ret;

   /* calling a function to get max value */
   ret = max(a, b);

   printf( "Max value is : %d\n", ret );

   return 0;
}

/* function returning the max between two numbers */
int max(int num1, int num2) {

   /* local variable declaration */
   int result;

   if (num1 > num2)
      result = num1;
   else
      result = num2;

   return result;
}
```

C program

- Example

```c
#include <stdio.h>
int addNumbers(int a, int b);          // function prototype

int main()
{
    int n1,n2,sum;

    printf("Enters two numbers: ");
    scanf("%d %d",&n1,&n2);

    sum = addNumbers(n1, n2);         // function call
    printf("sum = %d",sum);

    return 0;
}

int addNumbers(int a, int b)          // function definition
{
    int result;
    result = a+b;
    return result;                   // return statement
}
```

Practices

❑ Exercises

1. Write a function to calculate this formula $y=3x^2-2x$ where x is the parameter of the function. The function returns the value of y.

2. Write a function to display whether a person is allowed to vote or not. This function have one parameter which is the age of a person.
   - A person is allowed to vote when his/her age is greater than or equal 18.
   - Display the message either “You are allowed to vote” or “You are not allowed to vote”

Q&A