



# Wollo University, Kombolcha

## Institute of Technology

### College of Informatics

#### Department of Information System

By Daniel G.

Mar/2021 G.C


A vertical column of hexagonal icons on the left side of the slide. From top to bottom, the icons are: a lightbulb, a thumbs up, a network of nodes, a smartphone, a magnifying glass, a gear, and a speech bubble. The central hexagon containing the text 'INSY2031' is a larger, solid cyan color.

INSY2031

# Fundamentals of Programming II



# Contents

- ◇ Programing
  - ◇ Basic concept and need of function
  - ◇ Declaring, Calling and defining a function
  - ◇ Types and components Function
  - ◇ Calling /invoking function by value and referece
  - ◇ Functions Overloading and Recursion
- 

A decorative pattern of hexagons in various shades of blue and teal. Some hexagons contain icons: a lightbulb, a thumbs up, a network of nodes, a smartphone, a magnifying glass, a gear, and a speech bubble. A large teal hexagon in the center-left contains the number '1'.

# 1

# What is Programing

Programming is the process of creating a set of instructions that tell a computer how to perform a task. Programming can be done using a variety of computer programming languages, such as Php, Python, and C++

A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center with a white number '2'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and several smaller nodes connected by lines.

2

## About Function



A **function** is block of code which is used to perform a particular task.

or

Simply it is a **group of statements** that together perform a task.

It is a block of code which only runs when it is called



# Why Function

- ◇ Important for reusing code, Define the code once, and use it many times.
- ◇ Make your code simple, readable and reusable
- ◇ Function can Modularization a block code in to segments





# Syntax

## Syntax of Function

```
return_type function_name (parameter_lists ... )  
{  
    //C++ Statements  
}
```



A decorative pattern of hexagons in various shades of blue and cyan on the left side of the slide. Some hexagons contain icons: a lightbulb, a thumbs up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble.

3

Lets start coding with  
Function..

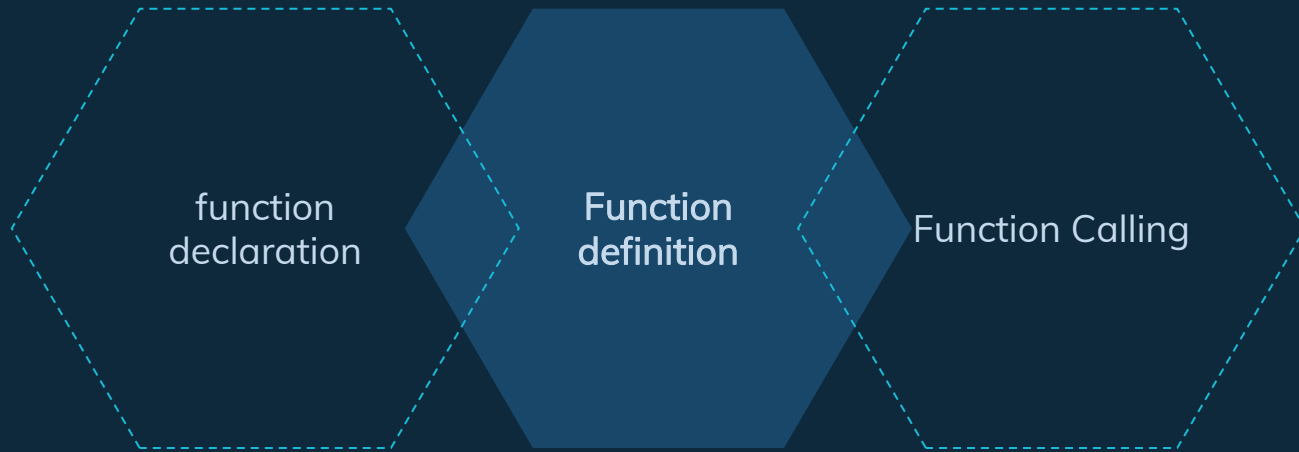


Waittt...

First things First  
THREE BIG CONCEPTS



# Basic parts of function





A Function declaration tells the compiler about a function's name, return type, and parameters followed by ; .

### Syntax

```
return_type function_name(parameter_list);
```



A Function definition provides the actual body of the function followed.

### Syntax

```
return_type function_name(parameter_list) {  
    //Statements inside function  
}
```



A **Function calling** invokes a function to be executed or telling the computer to execute set of actions.

**NB.** Declared or Define functions are not executed immediately. They are "saved for later use", and will be executed later, when they are called.

Syntax

```
function_name(parameters);
```



## Exercise

1. Write a program that will ask the user to input three integer values from the keyboard. Then it will print the smallest and largest of those numbers using function.

.

.

.

A decorative pattern of hexagons in various shades of blue and cyan. Some hexagons contain icons: a lightbulb, a thumbs up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble. The number '4' is centered in a large cyan hexagon.

4

# Types and components Function





# Types Function

- 1, Built-in functions
- 2, User-defined functions



Built-in functions are also known as library functions. We need not to declare and define these functions as they are already written in the C++ libraries such as `iostream`, `cmath` etc. We can directly call them when we need.

Example : `pow(x,y)`, `sqrt(x)`.....



The functions that we declare and write in our programs are **user-defined functions**

Example:

```
int sum(int,int){  
    return x +y;  
}  
  
int main(){  
    sum(4,7);  
    return 0;  
}
```



5

Calling /invoking function  
by value and referece



# Calling Function by value

In call by value, original value is not modified.

“

In call by value, value being passed to the function is locally stored by the function parameter in stack memory location.



# Calling Function by Referece

In call by reference, original value is modified because we pass reference (address).



# The Difference between call by value and reference

No.	Call by value	Call by reference
1	A copy of value is passed to the function	An address of value is passed to the function
2	Changes made inside the function is not reflected on other functions	Changes made inside the function is reflected outside the function also
3	Actual and formal arguments will be created in different memory location	Actual and formal arguments will be created in same memory location



## Example :

```
#include <iostream>
using namespace std;
void print(int& a){
    a = a + 1;
    cout << "print function " << a << "\n";
}
int main(){
    int x = 5;
    cout << "original " << x << "\n";
    print(x);
    cout << "after print " << x << "\n";
    return 0;
}
```



A decorative cluster of hexagonal icons in various shades of blue and cyan is located on the left side of the slide. The icons include a lightbulb, a thumbs-up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble.

6

# Functions Overloading and Recursion



Recursion is a process in which a function calls itself and the corresponding function is called the recursive function.

# Example :

```
#include <iostream>
using namespace std;
```

```
int f(int n){
    if (n <= 1)
        return 1;
    else
        return n*f(n-1);
}
```

```
int main(){
    int num;
    cout<<"Enter a number: ";
    cin>>num;
    cout<<"Factorial of entered number: "<<f(num);
    return 0;
}
```

“



Function overloading is a feature in C++ where two or more functions can have the same name but different parameters. In Function Overloading “Function” name should be the same and the arguments should be different.

Example:

```
int myFunction(int x)
float myFunction(float x)
double myFunction(double x, double y)
```



## Exercise :

1, Write a cpp program that works on each arithmetic operation and the program accepts an input `op` , `x` and `y` where `op` is the arithmetic operation to compute , `x` and `y` are numbers to compute.

the arithmetic operations must be listed first so that it can be selected using the number assigned to them and that number is the input `op` using function



# 50% DONE!

The remaining 50% is lab practice





# Thanks!

## Any questions?

Feel free!

