មហាវិទ្យាល័យវិស្វកម្ម
FACULTY OF ENGINEERING

Data Structure & Algorithm I

Lecture 2
Function, Class in C++

Chhoeum Vantha, Ph.D.

Telecom & Electronic Engineering

Content

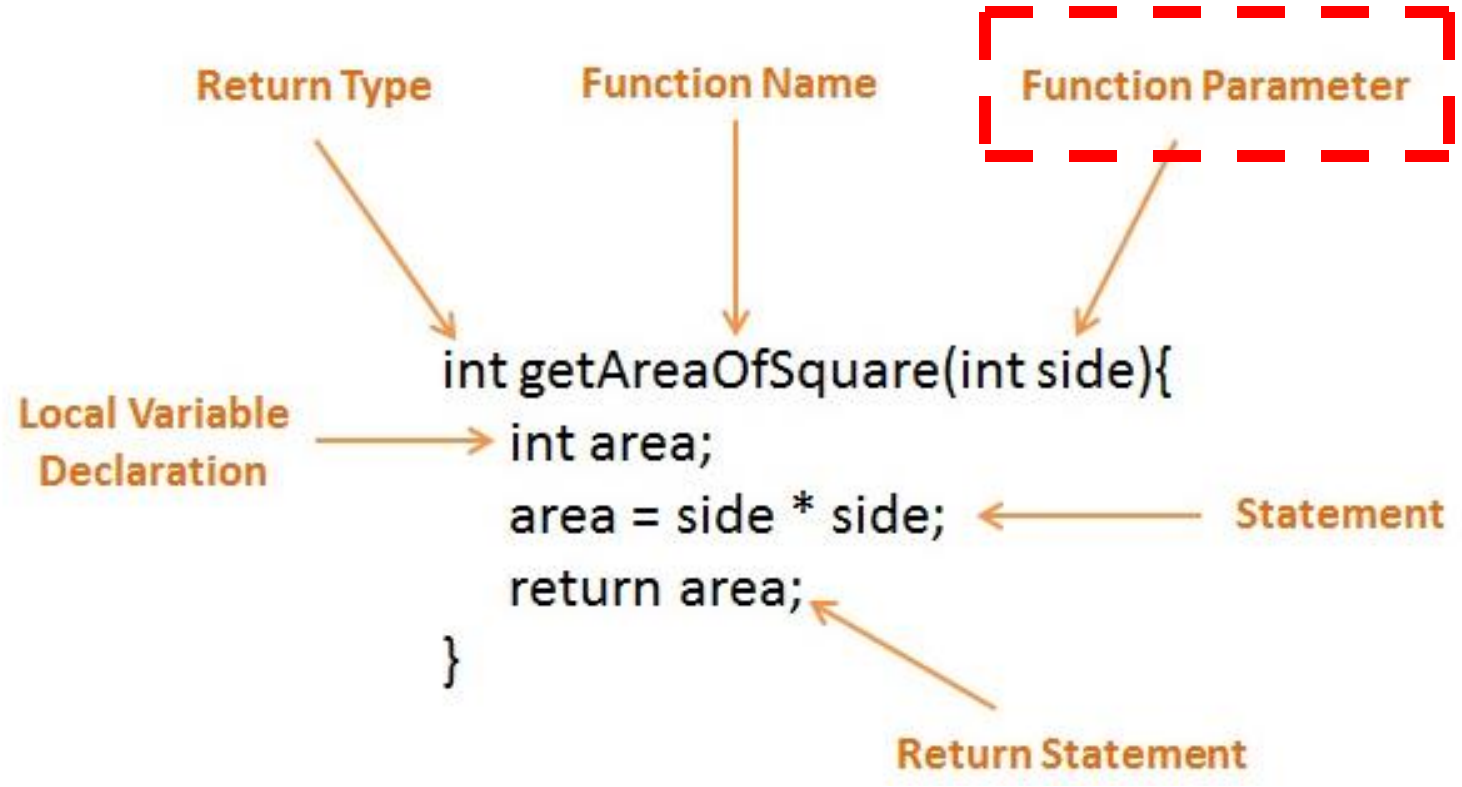
- Function in C++
- Class in C++

What is Function?

- A function is a block of code that only runs when it is called.
- You can pass data, known as parameters, into a function.
- Functions are used to perform certain actions, and they are important for reusing code:

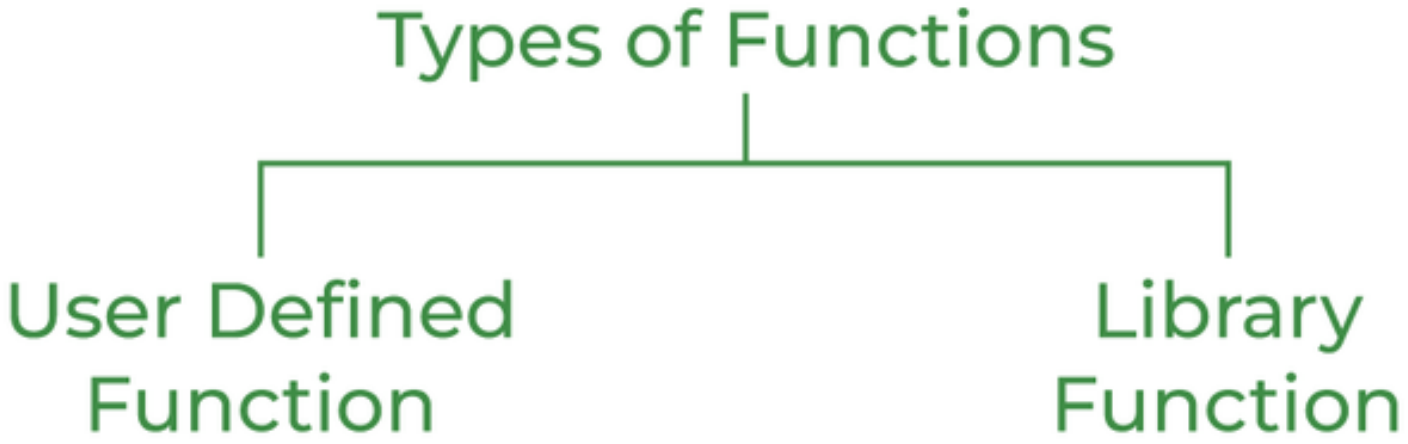
Functions - Parameters

- Local variables passed to function when called
- Provide outside information



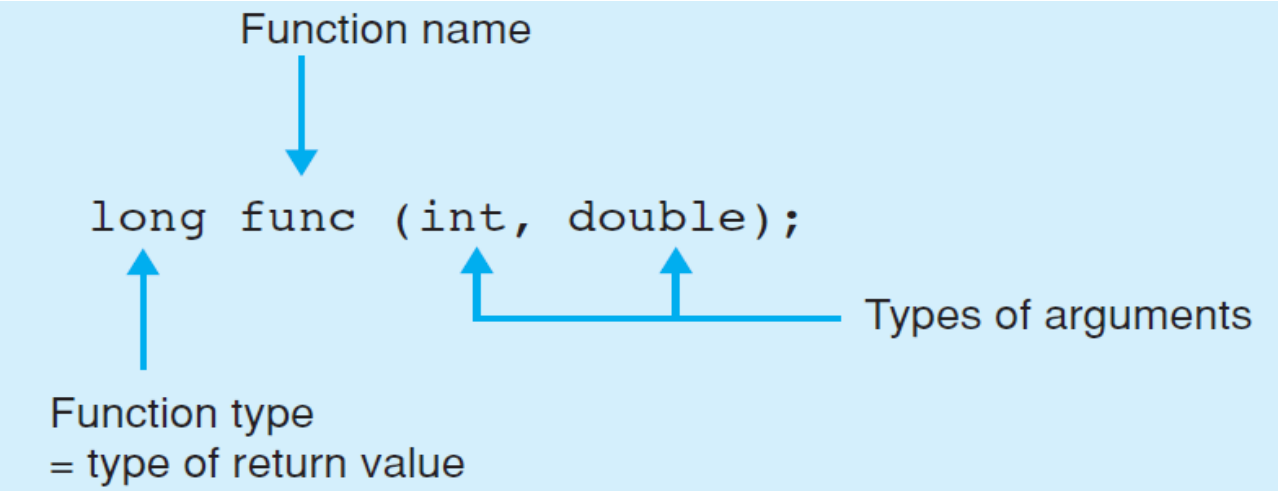
Functions - Types of functions

- There are two types of functions:
  - Standard Library Functions: Predefined in C++
  - User-defined Function: Created by users



Function - Declaration

The syntax to declare a function is:



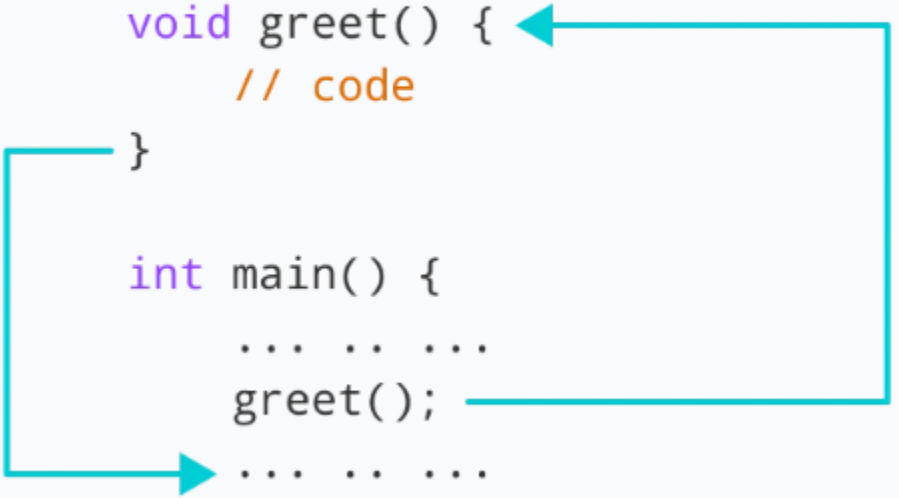

```
returnType functionName (parameter1, parameter2,...) {
    // function body
}
```

```
// function declaration
void greet() {
    cout << "Hello World";
}
```

Function – Calling

```
int main() {

    // calling a function
    greet();

}
```

```
#include<iostream>

void greet() {
    // code
}

int main() {
    ... .. ...
    greet();
    ... .. ...
}
```



User-defined Function Types

- Function with no argument and no return value

```
void print_name()
```

- Function with no argument but return value

```
int print_Number()
```

- Function with argument but no return value

```
void check_Number(int n)
```

- Function with argument and return value

```
int sum_xy(int x, int y)
```

Position of function in C++

- Above *int main ()* function

```
1   #include <iostream>
2   using namespace std;
3   int sum(int x, int y)
4   {
5       int a, b;
6       a = x;
7       b = y;
8       return x + y;
9   }
10  int main()
11  {
12      cout << sum(3,7);
13      return 0;
14  }
```

Position of function in C++

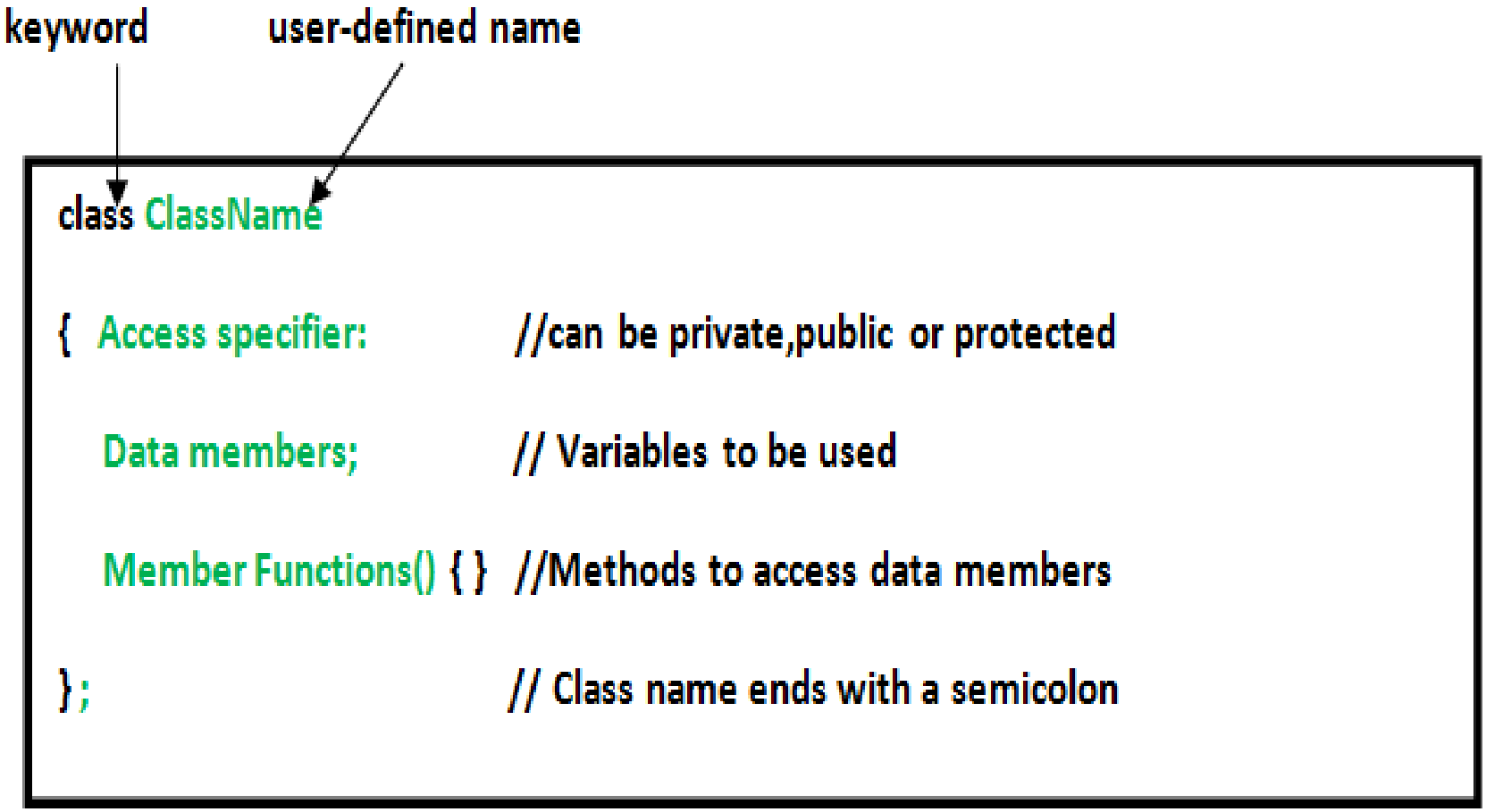
- Below *int main ()* function

```
1   #include <iostream>
2   using namespace std;
3   int sum(int n1, int n2);
4   int main()
5   {
6       cout << sum(3,7);
7       return 0;
8   }
9   int sum(int x, int y)
10  {
11      int a, b;
12      a = x;
13      b = y;
14      return x + y;
15  }
```

What is *class* in C++?

- A class is a user-defined data type that we can use in our program
- It works as an object constructor

Structure of Class in C++



Structure of Class in C++

```cpp
// C++ program to demonstrate
// accessing of data members

#include <bits/stdc++.h>
using namespace std;
class myFriend
{
    // Access specifier
    public:

    // Data Members
    string name;

    // Member Functions()
    void printname()
    {
       cout << "My Friend's name is " << name
    }
};
```

```cpp
int main() {

    // Declare an object of class myFriend
    myFriend fri1;

    // accessing data member
    fri1.name = "Dara";

    // accessing member function
    fri1.printname();
    return 0;
}
```

Object and Class in C++ Programming – Example

```cpp
// Program to illustrate the working of
// objects and class in C++ Programming
#include <iostream>
using namespace std;
// create a class
class Room
{
   public:
    double length;
    double breadth;
    double height;

    double calculateArea()
    {
        return length * breadth;
    }

    double calculateVolume()
    {
        return length * breadth * height;
    }
};
```

Object and Class in C++ Programming – Example

```
23  int main() {
24      // create object of Room class
25      Room room1;
26      // assign values to data members
27      room1.length = 42.5;
28      room1.breadth = 30.8;
29      room1.height = 19.2;
30      // calculate and display the area and volume of the room
31      cout << "Area of Room =  " << room1.calculateArea() << endl;
32      cout << "Volume of Room =  " << room1.calculateVolume() << endl;
33      return 0;
34  }
```

C++ Constructors with parameters - Example

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

class Car           // The class
{
  public:           // Access specifier
    string brand;   // Attribute
    string model;   // Attribute
    int year;       // Attribute
    Car(string x, string y, int z) // Constructor with parameters
    {
      brand = x;
      model = y;
      year = z;
    }
};
```

C++ Constructors with parameters - Example

```
17  int main()
18  {
19    // Create Car objects and call the constructor with different values
20    Car carObj1("BMW", "X5", 1999);
21    Car carObj2("Ford", "Mustang", 1969);
22
23    // Print values
24    cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";
25    cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";
26    return 0;
27  }
```

W2 – Lab

Ex 1

- Write a program that asks for two numbers, compares them, and shows the maximum.
- Declare a function called *max_two* that compares the numbers and returns the maximum.

Ex 2

Write a program to print the factorial of a number by defining a function named 'Factorial'.

Factorial Formula

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 1$$

$$1! = 1$$
$$2! = 2 \times 1 = 2$$
$$3! = 3 \times 2 \times 1 = 6$$
$$4! = 4 \times 3 \times 2 \times 1 = 24$$
$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

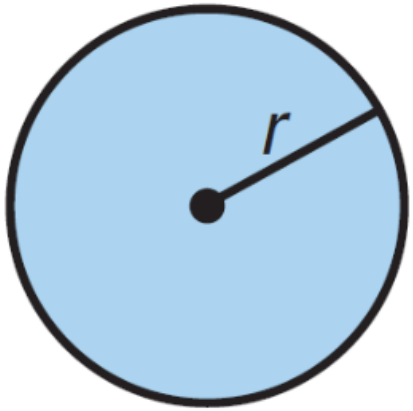
Ex 3.

- Write a C++ program to implement a class called ***Circle*** that has private member variables for radius.
- Include member **functions** to calculate the **circle's area** and **circumference**.

radius r

$C = 2\pi r$

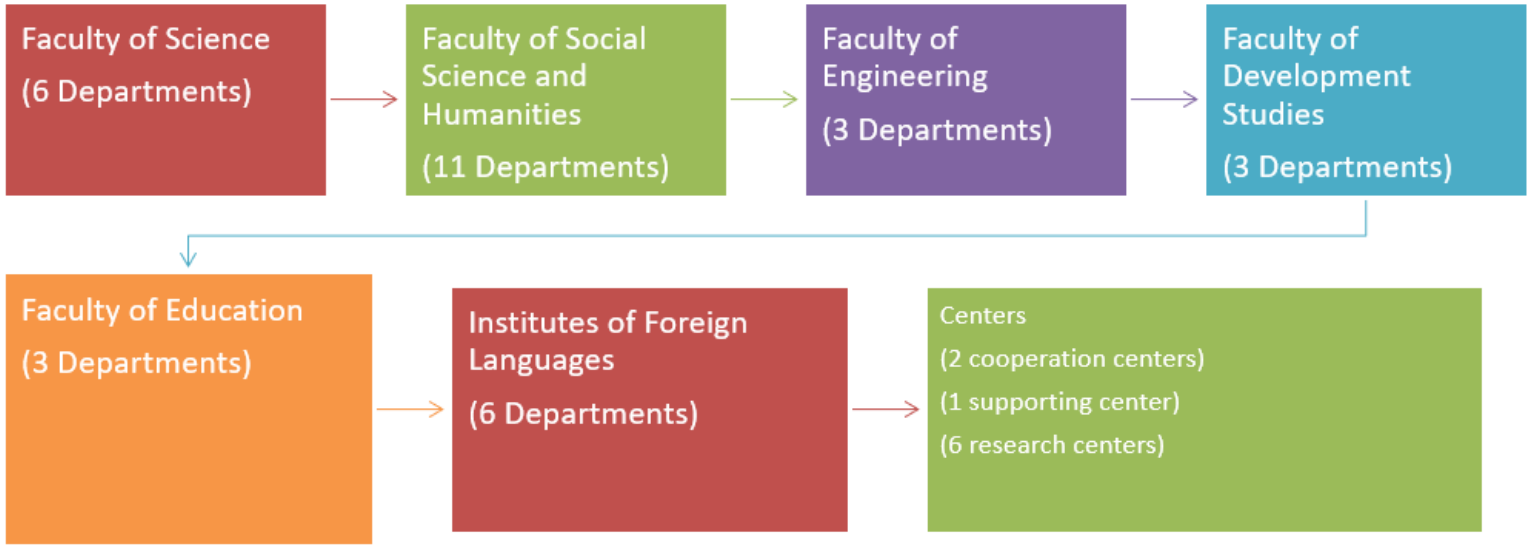
$A = \pi r^2$



Ex 4

Design and Write a class name as *infomRUPP* that contains a few pieces of information such as:

- Number of Faculty in RUPP
- Name of Faculty in RUPP
- Name of the department of FE
- Display all information-related classes of *infomRUPP*



Thanks!