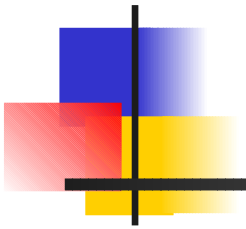
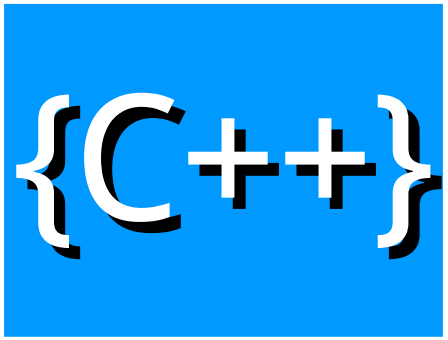


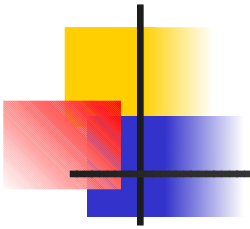
Functions

Week 4



Yang-Cheng Chang
Yuan-Ze University
yczhang@saturn.yzu.edu.tw

What is the use of function in C/C++



- Using function we can call the function in the program any where
- By using functions we can reduce the code redundancy

Reduce the code redundancy

- Original Code

```
// demostrate what the use of functions in c/c++
#include <iostream>
using namespace std;
int main(int argc, const char *argv[])
{
    int weight, height, age, YearOfBirth;
    cout << "input weight:" << endl;
    cin >> weight;
    cout << "input height:" << endl;
    cin >> height;
    cout << "input age:" << endl;
    cin >> age;
    cout << "input year of birth:" << endl;
    cin >> YearOfBirth;
    return 0;
}
```

Reduce the code redundancy

- Rewrite the code by using C++ functions

```
// demostrate what the use of functions in c/c++
#include <iostream>
#include <string>
using namespace std;

int prompt(string message)
{
    int inputValue = 0;
    cout << message << endl;
    cin >> inputValue;
    return inputValue;
}

int main(int argc, const char *argv[])
{
    int weight, height, age, YearOfBirth;
    weight = prompt("input weight:");
    height = prompt("input height:");
    age = prompt("input age:");
    YearOfBirth = prompt("input year of birth:");
    return 0;
}
```

Type of return value

The parameter of function

Return value

Function Definitions

- Format for function definition

```
return-value-type  function-name ( parameter-list )
{
   declarations and statements
}
```

- Parameter list
  - Comma separated list of arguments
    - Data type needed for each argument
  - If no arguments, use **`void`** or leave blank
- Return-value-type
  - Data type of result returned (use **`void`** if nothing returned)

Function Prototypes

- Function prototype contains
  - Function name
  - Parameters (number and data type)
  - Return type (void if returns nothing)
  - Only needed if function definition after function call
- Prototype must match function definition
  - Function prototype

    ```
    double maximum( double, double, double );
    ```

  - Definition

    ```
    double maximum( double x, double y, double z )
    {
    ...
    }
    ```

Most frequency mistakes in C++ function

- You can download the example program (mistake.cpp) from Virtual Classroom
- The return value type in a function definition doest not match the actual return value in a function statement
  - Line 9,12 in mistake.cpp
- Wrong parameter declarations in a function
  - Line 15 in mistake.cpp
- Undefined functions
  - Line 25, 27 in mistake.cpp

Week 4 Assignment

- A parking garage charges a $3.00 minimum fee to park for up to two hours.
- The garage charges an additional $0.60 per hour for each hour or part thereof in excess of two hours.
- The maximum charge for any given 24-hour period is $15.00.
- Assume that no car parks for longer than 24 hours at a time.

Week 4 Assignment

- Write a program that calculates and prints the parking charges for each of three customers who parked their cars in this garage yesterday.
  - You should enter the hours parked for each customer.
  - The program should use the function *calculateCharges* to determine the charge for each customer. The function has one parameter which is the hours parked for each customer.
  - Your program should print the results in a neat tabular format and should calculate and print the total of yesterday's receipts.

Week 4 Assignment

- Your outputs should appear in the following format

Car	Hours	Charge
1	1.5	3.00
2	4.0	4.20
3	24.0	15.00
TOTAL	29.5	22.20

Tip: Format output

- Fixed decimal precision

```
#include <iomanip> // std::setprecision
#include <iostream> //std::cout,std::fixed
int main(int argc, const char *argv[])
{
    double f =1.1;
    std::cout << std::fixed;
    std::cout << std::setprecision(5) << f << std::endl;
    Return 0;
}
```

- Set field width and make field left or right justified

```
// modify adjustfield using manipulators
#include <iostream>     // std::cout, std::left, std::right
#include <iomanip> // std::setw

int main () {
  int n = -77;
  std::cout << std::setw(6) << std::left << n << '\n';
  std::cout << std::setw(6) << std::right << n << '\n';
  return 0;
}
```