Data Structures

Dai Tian-Shyr
cameldai@mail.nctu.edu.tw

What The Course Is About

- Data structures is concerned with the representation and manipulation of data.
- All programs manipulate data.
- So, all programs represent data in some way.
- Data manipulation requires an algorithm.

What The Course Is About

- We shall study ways to represent data and algorithms to manipulate these representations.
- The study of data structures is fundamental to Computer Science & Engineering.

Course Information

- All course slides can be found on the website
- 助教
  - 周立軒 vic2211@gmail.com
  - 吳現任 y35246357468@gmail.com
  - 徐宇薇 yabe5413@gmail.com
  - 林俞君 sophia1715@hotmail.com.tw
  - 潘長華 cindy.dif99@nctu.edu.tw
- Textbook
  - Fundamental of Data Structures in C++
  - Horowitz, Sahni, Metha
  - 開發圖書魏錦玲 0939852332

Slides Presentations and Book Reading

- http://financelab.nctu.edu.tw/DataStructure.htm

Lecture	Content	Reading	Slides
1	Introduction, Insertion Sort	Section 7.2	Powerpoint
2	Algorithmic Complexity (Insertion Sort).	Section 1.7.	Powerpoint
3	Experimental Performance Measurement.	Section 1.7.	Powerpoint

評分標準

- Homework
  - Upload: http://dcpc.nctu.edu.tw/
  - Hand in on time
- 期中考
- 期末考
- 上課表現

Prerequisites
C++ Programming language

- C++ Enhancements as C programming language started with the addition of classes, followed by, among other features, encapsulation, inheritance and polymorphism.
- C++ is a object-oriented programming language.
- More information
  - http://www.cplusplus.com/

First C++ program

```
#include <iostream>    // 含括iostream標題檔
#include <cstdlib>              // 含括cstdlib標題檔
using namespace std;            // 使用std名稱空間
int main(void)
{
int a=12;
  float b=12.63F;
  char ch='T';
  cout << ch << "是字元"  << endl;     // 印出字元ch的內容,且換行
  cout << a << "是整數" << endl;       // 印出變數a的內容
  cout << b << "是浮點數" << endl;     // 印出變數b的內容

  system("pause");
  return 0;
}
```

Output:
T是字元
12是整數
12.63是浮點數

C++ Output format

#include <iomanip>

setw() //Set field width

setprecision()

//Set decimal precision(after the decimal point, drop numbers less then or equal to 4, and round up 1 for numbers higher/more than 4)

hex //Use hexadecimal base

oct //Use octal base

scientific / /Use scientific notation

cout example

```
#include <iostream>
#include <cstdlib>
#include <iomanip>
using namespace std;
int main(void)
{
int dec =10;
double flo=1.96789;
double do1= 3.14159 ,do2 = 1.0e-10;
cout<<setw(6)<<dec<<endl;
cout<<setprecision(1)<<flo<<endl;
cout<<setprecision(9)<<flo<<endl;
cout<<oct<<dec<<endl;
cout<<hex<<dec<<endl;
cout << scientific << do1<<endl;
cout << scientific << do2<<endl;
system("pause");
}
```

Output:
(四個空格)10
2
1.96789
12
a
3.141590000e+000
1.000000000e-010

C++ Basic if-else-if

```
if (condition) //若判斷condition成立，執行statement， 否則執行else if
    statement;
  else if (condition)
    statement;
     else
     statement;
```

```
statement →  {statement1;
              statement2;
               statement3;
               } //more than one statement
statement →statement //only one statement
```


if-else-if example

```
int score;
 cout<<"請輸入成績:"<<endl;
 cin>>score;          //讀取，輸入的資料

if (score>=80)
   cout<< score << "is A";                    /* 印出A */
 else if (score>=70)
       cout<< score << "is B";              /* 印出B */
     else if (score>=60)
           cout<< score << "is C";        /* 印出C */
         else
           cout<< "failed";       /* 印出字串"Failed!!" */
```

C++ Basic for-loop

for (設定迴圈初值;判斷條件;設定增減量)
 statement;

```
int sum=0;
for (int i=1;i<=9;i++)
    sum=sum+i;
```


Nested Loop

```
//九九乘法表的程式片段
for (int i=1;i<=9;i++)                  /* 外層迴圈 */
  {
    for (int j=1;j<=9;j++)  /* 內層迴圈 */
       cout<<i<<"*"<<j<<"="<<i*j<<" ";
    cout<<endl;
  }
```


C++ Basic while-loop

do
{
迴圈主體;
 	
}while (判斷條件); //符合才進入迴圈
	while (判斷條件) //符合才進入迴圈
	{
	迴圈主體;
	
	};

↓ 前測迴圈，至少執行一次	↓ 後測迴圈，至少執行0次

while-loop example

- /*sum from 1 to 10*/

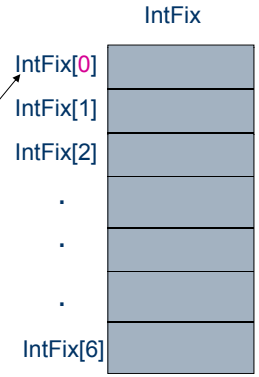
```
int i=1;
int tmp=0;
do
{
 tmp=tmp+i;//迴圈主體
 i++;  //增量
}while (i<=10);//判斷條件

cout<<"sum from 1 to 10 is "<< tmp;
```

C++ Basic Array

- 在記憶體中佔一塊連續的儲存空間
- 每一筆資料的資料型態都相同
- **`float IntFix[7];`**

資料型態　陣列名稱　陣列大小

從編號0至6,共7筆



取代宣告 `IntFix1, IntFix2,..`

陣列資料的使用

變數宣告:

```
float a;
float IntFix[7];
```

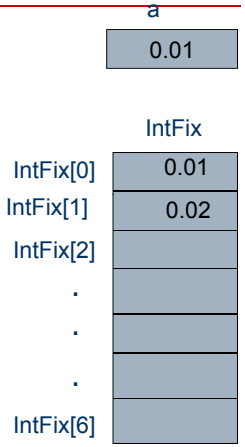
資料存取:

```
a=0.01;
IntFix[0]=a;
IntFix[1]=a+IntFix[0];
```

資料的輸入輸出

```
cin>>IntFix[2];
cout<<IntFix[2];
```

Don't write:　IntFix[7]=>不存在



2-dimensional arrays

```
int main (void)
{
int jimmy [3][5];
int n,m;
for (n=0;n<3;n++)
  {for (m=0;m<5;m++)
    {
      jimmy[n][m]=(n+1)*(m+1);
    }
  }
```

Jimmy[3][5]

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15

C++ Basic function

傳回值型態 函數名稱 (型態 引數1，型態 引數2,.....)

{

變數宣告;

敘述主體;

return 傳回值；//若有傳回值

}

```
int add(int num1, int num2)
{
   int a;
   a=num1+num2;
   return a;

}
```

```
void add()//無傳回值，及引數
{
  cout<<"a test function";

}
```

Example of function

```
int add(int num1, int num2)

{/* add()函數的定義 */
   int a;

   a=num1+num2;
   return a;

}
int main(void)
{
   int sum, a=5, b=3;
   sum=add(a,b);/* 呼叫add()函數
   cout<<sum;
}
```

若將function，置於main之後。

則需在main之前，宣告function原型

```
int add(int num1, int num2);//原型宣告
int main(void)
{
   int sum, a=5, b=3;
   sum=add(a,b);/* 呼叫add()函數
   cout<<sum;
}
int add(int num1, int num2)
{/* add()函數的定義 */
   int a;
   a=num1+num2;
   return a;
}
```


Sorting

- Rearrange a[0], a[1], ..., a[n-1] into ascending order. When done, a[0] <= a[1] <= ... <= a[n-1]
- 8, 6, 9, 4, 3 => 3, 4, 6, 8, 9

Sort Methods

- Insertion Sort :We focus on it in the following slide.
- Bubble Sort
- Selection Sort
- Count Sort
- Shaker Sort
- Shell Sort
- Heap Sort
- Merge Sort
- Quick Sort

Consider Inserting an Element First

- Given a sorted list/sequence, insert a new element
- Given 3, 6, 9, 14
- Insert 5
- Result 3, 5, 6, 9, 14

Insert an Element

- 3, 6, 9, 14 insert 5
- Compare new element (5) and last one (14)
- Shift 14 right to get 3, 6, 9, , 14
- Shift 9 right to get 3, 6, , 9, 14
- Shift 6 right to get 3, , 6, 9, 14
- Insert 5 to get 3, 5, 6, 9, 14

Implement “Inserting an Element”

```
// insert t into a[0:i-1]
int j;
for (j = i - 1; j >= 0 && t < a[j]; j--)
   a[j + 1] = a[j];
a[j + 1] = t;
```


Insertion Sort

- Start with a sequence of size 1
- Repeatedly insert remaining elements

Insertion Sort

- Sort 7, 3, 5, 6, 1
- Start with 7 and insert 3 => 3, 7
- Insert 5 => 3, 5, 7
- Insert 6 => 3, 5, 6, 7
- Insert 1 => 1, 3, 5, 6, 7

Insertion Sort

```
for (int i = 1; i < a.length; i++)
{// insert a[i] into a[0:i-1]
   // code to insert comes here
}
```


Insertion Sort

```
for (int i = 1; i < a.length; i++)
{// insert a[i] into a[0:i-1]
  int t = a[i];
  int j;
  for (j = i - 1; j >= 0 && t < a[j]; j--)
     a[j + 1] = a[j];
  a[j + 1] = t;
}
```


Homework

- Understand all the example codes in sec. 1.4
- Write a C++ function that returns the sum or the product of the first n numbers in an array.
  - If n<=5 , output the product of the first n numbers, otherwise output sum of the first n numbers (Test data ,n=-1, 5 and 10)
  - Your function should throw an exception in case n<0.