Programming in C/C++

Patrick Ho
peiqistar@gmail.com

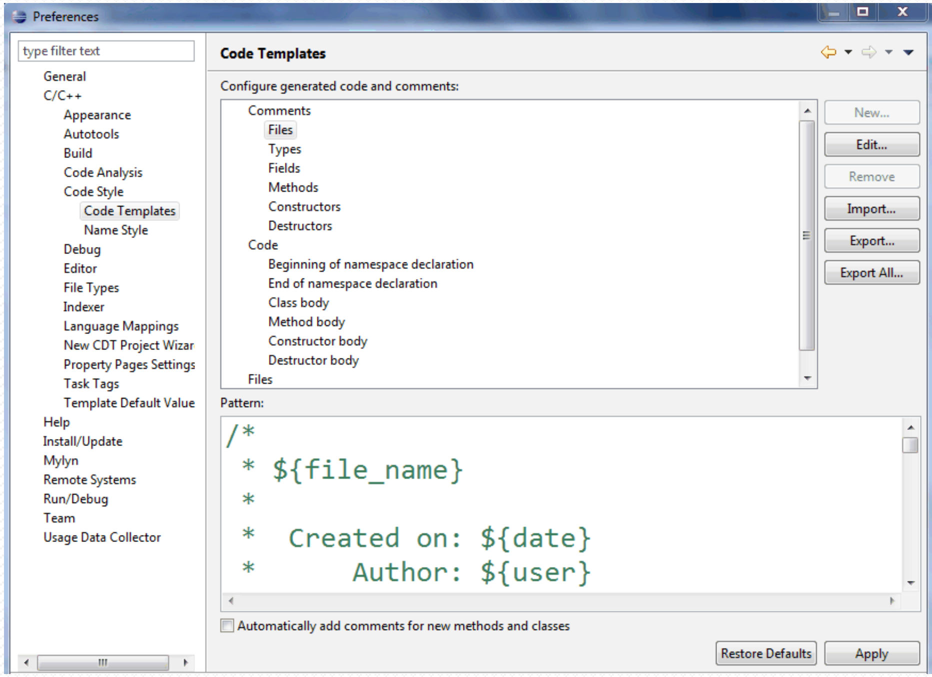
(Day3)

Eclipse C/C++ Template Setup

- **Help to quick coding**
  Time limit is KEY
- **Avoid typo**
- **Reliable coding style**

USACO Time Limit is KEY

•Leave more time for testing (make more test cases, especially corner case)



```
/*
 * ${file_name}
 *
 *  Created on: ${date}
 *      Author: ${user}
```

□ Automatically add comments for new methods and classes

USACO File Input/Output

- USACO (www.usaco.org)
  - Problem with one input file and one output file
  - USACO Input File:
    - Problem_Name.in
  - USACO Output File:
    - Problem_Name.out

- Trick:

  1. use freopen(…) function to change standard input pointers (stdin, stdout).
  2. use standard input/output functions to access data. (scanf, printf, <<, >>, etc.)

  This way can be used in any standard input/output problem by comment out freopen statement

- Code Example
  - freopen(“problem1.in", "r", stdin);   =>   Rd(“problem1.in”);
  - freopen(“problem1.out", "w", stdout); =>   Wt(“problem1.out”);

C++ Standard Input/Output

- **Advantages**
  - Short code characters
  - Easy for mixed data type
  - Easy for string
  - Easy for debug variable

- **Disadvantages**
  - Performance is slow, do not use is large loop case
  - Hard for complex output format , printf is easy and accurate

- **Header files**

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

- **Input item**
  - cin >> *variable_name*;

    e.g. int iData;   cin >> iData;

- **Output item**
  - cout << *variable_name*;

    e.g. int iData = 100; cout << iData;

Data Structure

- A particular way of storing and organizing data in a computer, so that it can be used efficiently

- A key to designing efficient algorithm

- Common types:
  - Array
  - Link list
  - Stack
  - Queue
  - Set
  - Map
  - Graph
  - Tree

Array

1. A collection of same type elements
   a. Total number of elements (Array size) is know

2. Each element can be identified by position number (index)
   a. Position number start from 0

3. Stored in memory with same size

4. One dimension or multi-dimention

5. Dynamic array in STL(Standard Template Library), use vector

Array (Static)

1. Declaring

   a. Type ArrayName[TotalSize]

   e.g. int c[12];

2. Access

   a. ArrayName[Index]

   e.g. c[0] = 10;
   int c10 = c[10];

3. Initial

```
a. int a[5] = {0};                    // all set to 0
b. int a[5] = {10,11,12,13,14};       // set one by one
c. char str1[ ] = “string array”;     // auto set size
```

Problem Solving

Generate a reversed string from a given sized string (max size is 1000)

Input: ABCDE
Output: EDCBA

Input: Aa
Output: aA

1. Solving:

```
/*
 * lesson4.cpp
 *
 *  Created on: Oct 15, 2011
 *      Author: PatrickHo
 */

…

int main()
{
    char sInputStr[1024] = {0};

    scanf(“%s”, sInputStr);
    //cout <<sInputStr << endl;
    int iNum = strlen(sInputStr);

    for (int i=iNum-1; i>=0; i--)
    {
        printf("%c",sInputStr[i]);
    }
    printf("\n");

    return 0;
}
```

String

- string, is a basic data type for char array(dynamic) (belong to STL)

- To define a string variable, use string variable_name

  e.g. string myName;
  string day[2] = {“Mon”, “Tue”};

- To access each char, use array [ ], or at(int i) function

- To assign value, use = sign, or “+” sign to append string, or cin/cout

  e.g. myName = “Patrick Ho ”;
  myName += “is a teacher.”;
  cin >> myName;

- To compare, use == operator

  e.g. if (myName == day[1])

- Use common STL build-in function (begin(), end(), size(), length(), etc)
- Print out: printf(“%s\n”, myName.c_str()); ***
  cout << myName << “\n”;

Problem Solving

Generate a reversed string from a given sized string

Input: ABCDE

Output: EDCBA

Input: Aa

Output: aA

1. Solving:

```
/*
 * lesson4.cpp
 *
 *  Created on: Oct 15, 2011
 *      Author: PatrickHo
 */

…

int main()
{
    string  sInputStr;
    cin >> sInputStr;
    // cout << sInputStr << endl;

    int iNum = sInputStr.size();

    //printf("input=%s\n", sInputStr);
    for (int i=iNum-1; i>=0; i--)
    {
        printf("%c",sInputStr[i]);
    }
    printf("\n");

    return 0;
}
```