

Common Defects for C/C++

Instructor: <Name of Instructor>

© FPT Software

1

Latest updated by: HanhTT1

Objectives

The course helps attendees understand:

- Common defects for C/C++ and how to prevent it.

Agenda

- Common Defect For C/C++
- Examples

Common Defects for C/C++

1. Coding Standard
2. Hard Code
3. Logic
4. Performance

Coding Standard

– Not follow coding rule

```
// DlgRule.cpp : implementation file
//

#include "stdafx.h"
#include "CCR.h"
#include "DlgRule.h"
#include "afxdialogex.h"
#include "Rule.h"
#include "GlobalVariable.h"
#include "DlgCreateNewARule.h"

// CDlgRule dialog

IMPLEMENT_DYNAMIC(CDlgRule, CDialogEx)

CDlgRule::CDlgRule(CWnd* pParent /*=NULL*/)
    : CDialogEx(CDlgRule::IDD, pParent)
    , m_tbxSearchValue(_T(""))
    , m_rtbxRegExpValue(_T(""))
    , m_rtbxDescriptionValue(_T(""))
    , m_rtbxWrongCaseValue(_T(""))
    , m_rtbxRightCaseValue(_T(""))
    , m_cbxTypeValue(_T(""))
{
}

// include<stdio.h>
// include<conio.h>
// include<string.h>
// include<malloc.h>
// include<stdlib.h>
#define nmax 100

static int count;

typedef struct
{
    char Address[40];
    char Name[30];
    int BirthDay;
    float Math;
    float Physic;
    float Chemical;
}student;

//khai tao danh sach
void create(student a[])
{
    count=0;
}
```

© FPT Software

5

Coding Standard(cont)

– Not understanding coding convention of C.

```
/*
In logic expression, set constant in the right.
For example:
*/
/* Not good: */
if ( a == 5 )
{
// TODO: add code here
}
/* Good: */
if ( 5 == a )
{
// TODO: add code here
}
```

Coding Standard(cont)

– Not check NULL

```
...  
 pInfoA->pPrinterName = (LPSTR) ((LPBYTE)pInfoA + ((LPBYTE)pInfoW->pPrinterName - (LPBYTE)pInfoW));  
 ...
```

Coding Standard(cont)

- Did not remove comma after last element of ENUM

```
typedef enum{
    /*interface_name = LIO */
    LIO_Motor_Direction = 0,
    LIO_Motor_Error,
    LIO_Motor_Position,
    LIO_Motor_Speed,
    LIO_Motor_Temp,
} l_signal_handle;
```

Coding Standard(cont)

- Hard to read the code

```
bool CCounter::IsQuotation(CString str,int idx)
{
    int len =  str.GetLength();
    if(str[idx]==""&&idx>=0 && idx<len && str[idx-1]!= 39 &&str[idx-1]!= 92)
        return true;
    return false;
}
```

Coding Standard(cont)

– Missing default in switch statement

```

void PStatUBasicFunction(tParameter* pParam,
                         char*      pName,
                         uint8_t     nProperty,
{
    pAPI->WTestTraceInfo(WTEXT("State 0x%02X"), pParam->State);
    switch (pParam->State)
    {
        case 0:
            /* Ask user to input a name */
            pAPI->WTraceInfo(WTEXT("Ask user to input a name"));
            pParam->nExpectedResult = MB_OK;
            pAPI->WPresentText(
                pName,
                VALUE_00,
                pParam);
            break;
        case 1:
            /* Register a name */
            pAPI->WTraceInfo(WTEXT("Register a name "));
            aPropertyArray[0] = nNameProperty;
            pParam->nError = pAPI->WReaderDetection(
                ADR_PReaderDetectionComplete,
                aPropertyArray,
                &pParam->hEventRegistry);
            if (pParam->nError != W_SUCCESS)
            {
                pAPI->WTestTraceError(WTEXT("WReaderDetection: Cannot register a name "));
                EndFunction(pParam, RESULT_FAILED);
            }
            break;
    }
    pParam->State++;
}

```

© FPT Software

10

Hard Code

– Use Hard code

```
// Used to find out if a given section exists
BOOL CCCRIniFile::SectionExists(CString lpszSection)
{
    CHAR ac_Result[100];
    CString csAux;
    // Get the info from the .ini file
    DWORD lRetVal = GetPrivateProfileString(lpszSection,NULL,
                                              _T(""),
                                              ac_Result, 90, m_lpszFileName);
    // Return if we could retrieve any info from that section
    return (lRetVal > 0);
}
```

Logic

- Các biến count trong các vòng lặp lồng nhau thì cần phải để là i, j, k

- Free memory of a pointer

```
int *p = new int[256];  
...  
delete p; /* Wrong */
```

Logic(cont)

– technique skill chưa tốt

```
/* Check the input character is special character */
bool CCounter::IsSpecialSymbol(WCHAR ch)
{
    if(ch=='=' || ch==',' || ch=='>' || ch=='<'
       || ch=='&' || ch=='|' || ch=='+' || ch=='-' || ch=='%'
       || ch=='^' || ch=='*' || ch=='?' || ch=='!')
        return true;
    return false;
}
```

Performance

– Variables are declared but it is not used

```
* [in]cbSize : UINT chua kich thuoc can cap phat
* Return values :
* LPITEMIDLIST : con tro den ITEMIDLIST da duoc cap phat
* Mod. History : 13/08/2012 - LongTT4
*
* --
*****
```

```
LPITEMIDLIST CSourceExplorerTreeCtrl::Create (UINT cbSize)
{
    LPMALLOC lpMalloc;
    HRESULT hr;
    LPITEMIDLIST pidl = NULL;
    if (FAILED(SHGetMalloc(&lpMalloc)))
    {
        return 0;
    }

    pidl=(LPITEMIDLIST) lpMalloc->Alloc(cbSize);

    if (pidl)
    {
        memset(pidl, 0, cbSize);
    }

    if (lpMalloc)
    {
        lpMalloc->Release();
    }

    return pidl;
}
```

© FPT Software

14

Performance(cont)

- Warning is still remaining when source code is compiled.

The screenshot shows a software interface with a code editor and an output window. The code editor contains C++ code for a function named `CSourceExplorerTreeCtrl::Create`. The output window displays several warning messages from the compiler:

```
source code\ccr\ccr\colorizer.cpp(122): warning C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy_s  
5) : see declaration of 'strcpy'  
source code\ccr\ccr\colorizer.cpp(233): warning C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy_s  
5) : see declaration of 'strcpy'  
source code\ccr\ccr\colorizer.cpp(626): warning C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy_s  
5) : see declaration of 'strcpy'  
source code\ccr\ccr\colorizer.cpp(628): warning C4996: 'strtok': This function or variable may be unsafe. Consider using strtok_s  
7) : see declaration of 'strtok'
```

© FPT Software

15

Apply tools

- [CPPCheck](#)
- [PVS-Studio](#)



© FPT Software

17