

Question 1:

Suppose we are going to create a new file named “**vuSample.txt**” on a hard disk using the Windows **CreateFile()** API. Write the signature of this API by setting the parameters as:

1. The file is created for read/write both.

FILE_ATTRIBUTES_NORMAL

2. The file is not required to be created for sharing.

dwShareMode: 0 Signature That File will Not Be Shared

3. Security attributes should be default.

IpsecurityAttributes

4. If the same name filename exists, then it is overwritten by the new file.

```
ofstream myfile ("vuSample.txt");
    myfile << "Your VU ID.\n";
    myfile << "Your VU ID.\n";
    myfile.close();
```

5. Set all other parameters as NULL.

array_create(NULL,10);

Question 2:

Write a C++ program for getting the last error code generated during execution of a process, convert this code into a user-understandable language format using the Windows **FormatMessage()** API and display the message along with the error code.

Solution

CODE:

```
#include <windows.h>
#include <everything.h>
#include <strsafe.h>

void ErrorExit(LPTSTR lpszFunction)
{
    LPVOID lpMsgBuf;
    LPVOID lpDisplayBuf;
    DWORD dw = GetLastError();

    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | 
        FORMAT_MESSAGE_FROM_SYSTEM | 
        FORMAT_MESSAGE_IGNORE_INSERTS,
        NULL,
        dw,
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR) &lpMsgBuf,
        0, NULL );
```

```
lpDisplayBuf = (LPVOID)LocalAlloc(LMEM_ZEROINIT,
    (lstrlen((LPCTSTR)lpMsgBuf) + lstrlen((LPCTSTR)lpszFunction) +
40) * sizeof(TCHAR));
StringCchPrintf((LPTSTR)lpDisplayBuf,
    LocalSize(lpDisplayBuf) / sizeof(TCHAR),
    TEXT("%s failed with error %d: %s"),
    lpszFunction, dw, lpMsgBuf);
MessageBox(NULL, (LPCTSTR)lpDisplayBuf, TEXT("Error"),
MB_OK);

LocalFree(lpMsgBuf);
LocalFree(lpDisplayBuf);
ExitProcess(dw);
}

void main()
{
if(!GetProcessId(NULL))
    ErrorExit(TEXT("GetProcessId"));
}
```