1. What is the base class for MFC Framework ?  
2. If i derive a new class from CObject what are the basic features my derived wil get ?  
3. What is the use of CCmdTarget ?  
4. What is document-view architecture ? Give me one real time example for SDI ?  
5. Can you explaing the relashionship between document,frame and view ?  
6. How to access document object from view ?  
7. What is the entry point for window based applications ?  
8. Explain the flow for a simple win32 based application ?  
9. How to handle command line arguements from simple MFC application ?  
10. What is model and modeless dialog box ? Give some examples?  
11. How to create open & save dialogs ?  
12.What is CSingleDocTemplate?  
13.What is the difference between hinsrtance and hprevinstance in WinMain function?  
14.Explaing about MDI and CMultiDocTemplate ?  
15.Tell me the different controls in MFC ?  
16. What is the use of OninitDialog ?  
17. What is the use of UpdateData funciton ?  
18.What is the difference between GetMessage and PeekMessage ?  
19.How to update all the views whenver document got updated ?  
20.How to handle RTTI in MFC ?  
21.What is serialization ?which function is responsible for serializing data ?  
22.What is CArchive class dowes?  
23.What is thread & process?  
24.Explain about different kinds of threads in MFC?  
25.Tell me about different kinds of synchranization objects ?  
26.what is the use of Mutex and critical section ?  
27.What is socket?  
28.What is the difference between Synchronous sockets and asynchronous sockets?  
29.Have you ever used win32 APIs ?  
30.What is the difference between ANSI Code and UNICODE ?  
31.What is the difference between regular dlls and extended dlls?  
32.what is the use of AFX\_MANAGE\_STATE ?  
33. What’s the difference between PostMessage and SendMessage?  
34. Describe the Document/View architecture.  
35. What’s the difference between Modal and Modeless Dialog?  
36. How to create a Modeless Dialog?  
37. How to setup a timer?  
38. Name the Synchronization objects.  
39. What is a critical section and how is it implemented?  
40. What is CMutex?  
41. Given two processes, how can they share memory?  
42. What is a message map, and what is the advantage of a message map over [virtual functions](http://www.go4expert.com/articles/virtual-static-friend-functions-cpp-t29967/)?  
43. What is the difference between [ASSERT](http://www.go4expert.com/articles/c-cpp-assert-function-t27488/) and VERIFY?  
44. Why wizards generate enum IDD for dialogs? [Added by Shabbir]

**Tell me the different controls in MFC ?**  
CAnimateCtrl,CButton,CEdit,CListBox,CComboBox,CRichEditCtrl,CStatic, CTreeCtrl,CToolTipCtrl,CIPAddressCtrl,CTabCtrl,CDateTimeCtrl,CHeaderCtrl,CListCtrl,CMonthCalCtrl,COleCtrl,CProgressCtrl,CScrollBar,CSliderCtrl,CStatusBarCtrl,CTollBarCtrl etc.,

**How to update all the views whenever document got updated ?**  
  
call UpdateAllViews()- which updates all views associated with the document by calling OnUpdate() function of all the views.

**How to handle RTTI in MFC ?**  
  
Run-Time Type Information меха that allows the type of an object to be determined during the program execution.  
3 main elements to RTTI in MFC are  
  
1.Dynamic\_cast operator  
Used for conversion of polymorphic types.  
2.typeid - used for identifying the exact type of an object   
3. type\_info class  
used to hold the type information returned by typeid.

**What is serialization ?which function is responsible for serializing data ?**  
  
Searialization is the process of streaming the object data to or from a persistent storage medium. It's useful in Doc-View Architecture. CObject :: Serialize() function is used to do serialization.

**Explain about different kinds of threads in MFC?**  
  
2 types of thread in MFc are UserInterface thread and [worker thread](http://www.go4expert.com/articles/worker-threads-mfc-t3197/). UserInterface threads maintain the message loops and used to handles user input,creates windows and process messges sent to those windows.Worker thread don't use message loops and mainly used to perform background operations such as printing etc.,Created using AfxBeginThread bypassing ThreadFunction to create worker thread and Runtime class object to create a user interface thread.

**what is the use of Mutex and critical section ?**  
  
Mutex as the name suggest allows a mutullay exclusive access to a shared resource among the threads. Critical section is a piece of code that can be executed safely to be accessed by two or more threads. Criticalsection provides synchronization means for one process only, while mutexes allow data synchronization across processes. Means two or more threads can share the common resources among more than one application or process boundaries in mutex.

**What is socket?**  
  
A "socket" is an endpoint of communication: an object through which your application communicates with other Windows Sockets applications across a network.The two MFC Windows Sockets programming models are supported by the following classes: CAsyncSocket and CSocket

**What is the difference between Synchronous sockets and asynchronous sockets?**  
  
Consider a server application that is listening on a specific port to get data from clients. In synchronous receiving, while the server is waiting to receive data from a client, if the stream is empty the main thread will block until the request for data is satisfied. Hence, the server cannot do anything else until it receives data from the client. If another client attempts to connect to the server at that time, the server cannot process that request because it is blocked on the first client. This behavior is not acceptable for a real-world application where we need to support multiple clients at the same time.   
  
In asynchronous communication, while the server is listening or receiving data from a client, it can still process connection requests from other clients as well as receive data from those clients. When a server is receiving asynchronously, a separate thread (at the OS level) listens on the socket and will invoke a callback function when a socket event occurs. This callback function in turn will respond and process that socket event.

**Have you ever used win32 APIs ?**  
  
MFC is a wrapper around win32 API, It provides classes which uses the win32 API, Some of the API's we usually work with are : GetDlgItemInt,GetDlgItemText,GetWindowTextA,MessageBoxA,CreateFile,CreateMutex,CreateEvent,WaitForSingleObject,CreateWindow,ShowWindow etc.,

**What is the difference between ANSI Code and UNICODE ?**  
  
(!) ANSI code represents 8bits( one byte) data where UNICODE represents 16bits (2 bytes) data for supporting universal languages.

ANSI code represents 8bytes data where UNICODE represents 16bytes data for supporting universal languages. One major draw back to ASCII was you could only have 256 different characters. However, languages such as Japanese and Arabic have thousands of characters. Thus ASCII would not work in these situations. The result was Unicode which allowed for up to 65,536 different characters

**What is the difference between regular dlls and extended dlls?**  
  
Regular dlls wraps only the c/c++ functions. Where extention dlls include c++ interfaces where we can create the objects of it and use in our classes. Extended dlls support object oriented concepts.Regural dlls uses mfc internally and exported functions can be used by any mfc or non-mfc applications.Extention dlls implements reusable classes derived from mfc library,built using dll version of mfc.Only mfc executables(applications/dll-shared version of mfc) can use extention dlls.extention dlls used for passing mfc derived objects b/w applications and dlls. Regulardlls linked both statically and dynamically but extended dlls links dynamically.

**What is a message map, and what is the advantage of a message map over virtual functi**  
  
MessageMap is a logical table that maps the windows messages to the member functions of the class. We use message maps over virtual function because of lots of overhead. If every windows message had a virtual function associated with it , there would be several hundred bytes per window class of vtable. Message maps means we only pay for the messages we use.

**Given two processes, how can they share memory?**  
  
Processes and thread are very similar but they differ in the way they share their resources. Processes are independent and have its own address space. If two independent processes want to communicate they do this by using the following techniques

1.Message Passing

if it is a windows based application, the simplest one is to use [WM\_COPYDATA](http://www.go4expert.com/articles/interprocess-communication-wmcopydata-t19730/) message

2.Sockets

3. named pipes

4. Shared Memory over memory-mapped files to share data memory between  
processes

**How to restrict only one instance of a class object to be created?**

1. Create a Named Mutex.  
   HANDLE hMutex=CreateMutex(TRUE,\_T(“NamedMutex”))

And check the Mutex existence for each of your instance launch and dont allow it to launch if it exists.

1. This answer is sutable for restrict only one instance of a process,  
   if u want make singleton class make custructor as private

**How do I dynamically change the mainframe menu?**  
  
CMenu newMenu;  
newMenu.LoadMenu (IDR\_MENU1);  
AfxGetMainWnd()->SetMenu( &newMenu );  
AfxGetMainWnd()->DrawMenuBar();  
newMenu.Detach ();

**What is the base class for MFC Framework ?**  
  
CObject

**If I derive a new class from CObject what are the basic features my derived will get?**  
  
Searialization, Debugging support, Runtime time class information, compatibility with collection classes.

**What is the use of CCmdTarget ?**  
  
It is the base class for the MFC library message map architecture.Which maps commands/messages to the member functions to handle them. Classes derived from this are CWnd,CWinApp,CFrameWnd,CView, CDocument

**What is document-view architecture ? Give me one real time example for SDI ?**  
  
Document/view architecture, which defines a program structure that relies on document objects to hold an application's data and on view objects to render views of that data. MFC provides the infrastructure for documents and views in the classes CDocument and CView.  
  
example of SDI is a wordpad application

**Can you explain the relashionship between document,frame and view ?**  
  
The frame window is the application's top-level window. It's normally a WS\_OVERLAPPEDWINDOW-style window with a resizing border, a title bar, a system menu, and minimize, maximize, and close buttons.  The view is a child window sized to fit the frame window so that it becomes the frame window's client area.  The application's data is stored in the document object, a visible representation of which appears in the view.   
For an SDI application, the frame window class is derived from CFrameWnd, the document class is derived from CDocument, and the view class is derived from CView or a related class such as CScrollView.

**How to access document object from view ?**  
  
Using GetDocument() function within a CView class.

**5 What is the entry point for window based applications ?**  
  
WinMain() is the entry point for window based applications.

**16 What is the difference between GetMessage and PeekMessage ?**  
  
GetMessage function waits for a message to be placed in the queue before returning where as PeekMessage function does not wait for a message to be placed in the queue before returning.

**31 What’s the difference between PostMessage and SendMessage?**  
  
The PostMessage function places (posts) a message in the message queue associated with the thread that created the specified window and then returns without waiting for the thread to process the message.  
  
The SendMessage function sends the specified message to a window or windows. The function calls the window procedure for the specified window and does not return until the window procedure has processed the message.

**34 How to create a Modeless Dialog?**  
  
m\_pModeless is a variable of type CDialog or any of its descendants.  
  
m\_pModeless->Create(IDD\_DIALOG1, this);  
m\_pModeless->ShowWindow(SW\_SHOW);   
  
this pointer as a paramter suggest we are creating a child dialog of the current dialog/window.

**35 How to setup a timer?**  
  
Use the SetTimer function

Code:

UINT\_PTR SetTimer( HWND hWnd,

UINT\_PTR nIDEvent,

UINT uElapse,

TIMERPROC lpTimerFunc

);

To kill the timer use

Code:

BOOL KillTimer(int nIDEvent);

Where the nIDEvent is the ID returned by the SetTimer Function.

**36 Name the Synchronization objects ?**  
  
**Following are the synchronization objects**  
1) Critical Section 2) Event 3) Mutex 4) Semaphore  
  
**Classes provided for above synchronization objects are:**  
  
1) CCriticalSection  
2) CEvent  
3) CMutex  
4) CSemaphore

**38 What Is CMutex ?**  
  
An object of class **CMutex** represents a **“mutex”** — a synchronization object that allows one thread mutually exclusive access to a resource. Mutexes are useful when only one thread at a time can be allowed to modify data or some other controlled resource. For example, adding nodes to a linked list is a process that should only be allowed by one thread at a time. By using a **CMutex** object to control the linked list, only one thread at a time can gain access to the list.  
  
To use a **CMutex** object, construct the CMutex object when it is needed. Specify the name of the mutex you wish to wait on, and that your application should initially own it. You can then access the mutex when the constructor returns. Call CSyncObject::Unlock when you are done accessing the controlled resource.  
  
An alternative method for using **CMutex** objects is to add a variable of type CMutex as a data member to the class you wish to control. During construction of the controlled object, call the constructor of the CMutex data member specifying if the mutex is initially owned, the name of the mutex (if it will be used across process boundaries), and desired security attributes.  
  
To access resources controlled by **CMutex** objects in this manner, first create a variable of either type CSingleLock or type CMultiLock in your resource’s access member function. Then call the lock object’s Lock member function (for example, CSingleLock::Lock). At this point, your thread will either gain access to the resource, wait for the resource to be released and gain access, or wait for the resource to be released and time out, failing to gain access to the resource. In any case, your resource has been accessed in a thread-safe manner. To release the resource, use the lock object’s Unlock member function (for example, CSingleLock::Unlock), or allow the lock object to fall out of scope  
  
**40 What is the difference between ASSERT and VERIFY?**  
  
The main difference between ASSERT and VERIFY is when you compile the release build of the program.  
  
ASSERT will not be present in the release build version of the executables/dlls, and its expression that would have been evaluated will be deleted.  
  
VERIFY will be present in the release build version of the executables/dlls but its expression that would have been evaluated will be left intact.

**21 What is thread & process?**  
  
Threads are similar to processes, but differ in the way that they share resources.

- Threads are различаются from processes in that processes are typically independent

- carry considerable state information and have separate address spaces.

Threads typically share the memory belonging to their parent process.

**30 what is the use of AFX\_MANAGE\_STATE ?**  
  
По умолчанию, MFC использует HANDLE-ресурсов (resource handle) основного приложения, для загрузки шаблона ресурсов (resource template). Если в DLL имеется экспортируемая функция, такая, которая к примеру запускает диалоговое окно внутри DLL, этот шаблон на самом деле хранится внутри в DLL-модуле. Нужно сменить состояние модуля, чтобы был использован правильный HANDLE-ресурсов. Это можно сделать путём добавления следующего кода в начало функции:  
AFX\_MANAGE\_STATE(AfxGetStaticModuleState( ));  
Это меняет текущее состояние модуля на состояние, которое вернёт AfxGetStaticModuleState() до конца текущего блока (until the end of the current scope).  
  
Если все ваши ресурсы лежат в одной DLL, вы даже можете поменять дефолтный Хэндл до Хэндла DLL-ки при помощи функции AfxSetResourceHandle.

**44 Почему wizards генерируют enum IDD для диалоговых окон?**  
  
Это хорошая практика поступать именно так , потому как благодаря :: двойному двоеточию через область видимости из нашего класса CMyDld через конструкцию CMyDlg::IDD мы можем получить именно тот айдишник, что нам нужен, не беспокоясь о том, какой именно константе он равен. Плюс мы имеем не просто число, а условное имя айдишника, которое понятно человеку, а не только машине.

**20 What is CArchive class does?**  
  
CArchive класс позволяет сохранять сложные сетевые объекты во временной бинарной-форме (как правило в место постоянного хранения (диск)) который освобождается после того, как эти объекты удаляются. Позже вы можете загрузить объекты из места постоянного хранения, и восстановить их из памяти. Процесс того как вы делаете данные доступными для хранения и переработки называется “Сериализация”.

**8 What is model and modeless dialog box ? Give some examples?**  
  
When we create Modal Dialog Box we can't move to other windows until this dialog is closed. For eg: MessageBox, where we can't move to the other window until we press ok or cancel.   
  
When we create Modeless Dilaog Box we can swap to the other windows. For eg: like a conventional window.

**9 How to create open & save dialogs ?**  
  
In CommonDialogs class we have to use CFileDialog class where the first parameter TRUE for open dialog and FALSE for Save dialog.  
  
For file open:  
CFileDialog SampleDlg(TRUE,NULL,NULL,OFN\_OVERWRITEPROMPT,"Text Files (\*.txt)|\*.txt|Comma Separated Values(\*.csv)|\*.csv||");  
  
int iRet = SampleDlg.DoModal();

**6 Explain the flow for a simple win32 based application ?**  
  
Начинает win32 приложение выполняться с функции WinMain() – это его точка входа.  
  
WinMain начинается с вызова API функции RegisterClass для регистрации класса окна.  
  
Класс окна (WNDCLASS) определяет важные характеристики окна, такие как его адресс оконной процедуры окна , дефолтный цвет фона окна, и оконную иконку.  
После регистрации класса WNDCLASS , функция WinMain вызывает самую гнлавную CreateWindow функцию, чтобы создать окно приложения  
  
WinMain продолжает выполнение CreateWindow и продолжает после делать вызовы функций ShowWindow и UpdateWindow, which make the window visible and ensure that its WM\_PAINT handler is called immediately.   
  
Next comes the message loop. In order to retrieve and dispatch messages, WinMain executes a simple while loop that calls the GetMessage, TranslateMessage, and DispatchMessage API functions repeatedly.   
  
GetMessage checks the message queue. If a message is available, it is removed from the queue and copied to msg;  
TranslateMessage converts a keyboard message denoting a character key to an easier-to-use WM\_CHAR message,   
and DispatchMessage dispatches the message to the window procedure. The message loop executes until GetMessage returns 0, which happens only when a WM\_QUIT message is retrieved from the message queue. When this occurs, WinMain ends and the program terminates.

**10 What is CSingleDocTemplate?**  
  
It’s a document template class used to create single document interface SDI applications. Only one document can be opened at a time. It identifies the document class used to manage the application's data, the frame window class that encloses views of that data, and the view class used to draw visual representations of the data. The document template also stores a resource ID that the framework uses to load menus, accelerators, and other resources that shape the application's user interface.

**12 Explain about MDI and CMultiDocTemplate ?**  
  
MDI applications are designed using the doc-view architectures in which there could be many views associated with a single document object and an application can open multiple docuements at the same time for eg:WORD.  
In MDI terms, your main window is called the Frame, this is probably the only window you would have in a SDI (Single Document Interface) program. In MDI there is an additional window, called the MDI Client Window which is a child of your Frame window. CMultiDocTemplate is the document template class used to create MDI applications..The document template also stores a resource ID that the framework uses to load menus, accelerators, and other resources that shape the application's user interface.

**14 What is the use of OnInitDialog ?**  
  
This message is sent to the dialog box during the Create, CreateIndirect, or DoModal calls, which occur immediately before the dialog box is displayed. This can be used to intialize the dialog controls or show/hide the controls etc.,

**15 What is the functioning of UpdateData() funciton ?**  
  
This is to initialize data in a dialog box, or to retrieve and validate dialog data.  
The framework automatically calls UpdateData with bSaveAndValidate set to FALSE when a modal dialog box is created in the default implementation of CDialog::OnInitDialog. The call occurs before the dialog box is visible. The default implementation of CDialog::OnOK calls this member function with bSaveAndValidate set to TRUE to retrieve the data, and if successful, will close the dialog box. If the Cancel button is clicked in the dialog box, the dialog box is closed without the data being retrieved.

**11 What is the difference between hinstance and hprevinstance in WinMain function?**  
  
hInstance is used for things like loading resources and any other task which is performed on a per-module basis. A module is either the EXE or a DLL loaded into your program. hPrevInstance used to be the handle to the previously run instance of your program (if any) in Win16. It is always NULL for Win32 programs.

**7 How to handle command line arguments from simple MFC application ?**  
  
**m\_lpCmdLine** Corresponds to the lpCmdLine parameter passed by Windows to WinMain. Points to a null-terminated string that specifies the command line for the application. Use m\_lpCmdLine to access any command-line arguments the user entered when the application was started. m\_lpCmdLine is a public variable of type LPTSTR.  
  
Example

BOOL CMyApp::InitInstance()

{

// ...

if (m\_lpCmdLine[0] == \_T('\0'))

{

// Create a new (empty) document.

OnFileNew();

}

else

{

// Open a file passed as the first command line parameter.

OpenDocumentFile(m\_lpCmdLine);

}

// ...

}

**50 Как же работет Document View Архитектура ?**

**Документ** – контейнер, в котором сдержатся данные приложения, Наш документ от CDocument получает данные из Бд / Диска, и т.д. обновляет эти данные внеутри себя, при их обновлении и сохраняет их при необходимости.

- Serialize() – когда пользователь сохраняет свою работу с приложением, или загружает что-нибудь с диска. IsStored() – показывает нужно ли данные сохранить или загрузить с диска.

**Вьюха** – то что даёт пользователю видеть данные, наследуется от CView, усправлять данными

Через мышь, клавитатуру

- OnDraw() стартует при запуске программы. То что показывает юзеру данные на

форме/окне, принтере, связывается с Доком через GetDocument

MESSAGE\_MAP – обработка спец событий и процессоб приходящих от ОС БатнКлики, нажатия клавиш, движения мыши.

SDI SingleDocIndetface и MDI Multi – Мульти может работать как несколькими Док, так и Вьюхами