



MUHAMMAD FAISAL

MIT 4th Semester

Al-Barq Campus (VGJW01) Gujranwala

faisalgrw123@gmail.com

CS410 Short Questions for Mid Term Exams

CS410 - WINDOWS PROGRAMMING

Q.No.1 How can I use the CopyTo method of the Windows Forms controls collection to copy controls into an array?

Answer:-

The CopyTo method will take the current controls collection and copy all the controls into an array. In order to use it, you must specify the array and the starting point. For instance, the following code copies the controls to the MyArrayOfControls array starting at the first element

Reference: <http://msdn.microsoft.com/en-us/magazine/cc164035.aspx>

Q.No.2 Differentiate pen & brush? (Page#69)

Answer:-

PEN	Brush
A pen is a graphics tool that an application for Microsoft Windows uses to draw lines and curves	A brush is a graphics tool that a Windows based application uses to paint the interior of polygons, ellipses, and paths
Pens to draw freehand lines, straight lines, and curves.	brushes use to paint shapes
pens to designate trends in graphs and to outline bar graphs and pie charts	Brushes to paint the sections of pie charts and the bars in bar graphs.

Q.No.3 It is sometimes more efficient for an application to draw directly in a window without relying on the WM_PAINT message. How this task can be accomplished (i.e. how can we draw in a window directly without using WM_PAINT message)?

(Page#167)

Answer:-

The task can be accomplished which is given below:-

This can be useful when the user needs immediate feedback, such as when selecting text and dragging or sizing an object. In such cases, the application usually draws while processing keyboard or mouse messages.

To draw in a window without using a WM_PAINT message, the application uses the GetDC or GetDCEX function to retrieve a display device context for the window. With the display device context, the application can draw in the window and avoid intruding into other windows. When the application has finished drawing, it calls the ReleaseDC function to release the display device context for use by other applications. application draws a selection and an intervening WM_PAINT message occurs, the application must ensure that any drawing done during the message does not corrupt the selection. To avoid this, many applications remove the selection, carry out usual drawing operations, and then restore the selection when drawing is complete

Q.No.4 How Windows keep track of the files? (Page#78)

Answer:-

The system requires instance handles to keep track of all modules. The system assigns a handle to each copy of a running executable or .dll.

Qno.5 Can you write a class without specifying namespace? Which namespace does it belong to by default?

Answer:-

Yes, you can, and then the class belongs to global namespace which has no name. For commercial products, naturally, you wouldn't want global namespace.

Reference: http://www.netsqlinterviewquestions.com/-net_interview_questions/580_can-you-write-a-class-without-specifying-namespace-which-namespace-does-it-belong-to-by-default-.aspx?topicid=2

Q.No.6 "In the GDI environment there are two working spaces", Name these two. (Page#140)

Answer:-

In the GDI environment there are two working spaces

1. Logical
2. Physical

Q.No.7 How many kinds of macros are there? (Page#40)

Answer:-

There are two kinds of macros.

1. Object-like macros resemble data objects when used,
2. Function-like macros resemble function calls.

Q.No.8 What are the GDI environment working space names? (Page#140)

Answer:-

In the GDI environment there are two working spaces

1. Logical
2. Physical

Q.No.9 List down three Pre-Defined GDI objects in window. (Page#178)

Answer:-

Pre-defined GDI objects in Windows are:

- Pens
- Brushes
- Fonts
- Palettes

Q.No.10 What are macros and its types? Explain it with example. (Page#40)

Answer:-

There are two kinds of macros.

- 1 Object-like macros resemble data objects when used,
- 2 Function-like macros resemble function calls.

Example:-

Here's a macro that computes the maximum of two numeric values:

```
#define min(X, Y) ((X)>(Y) ? (X):(Y))
```

Explain

To define a macro that takes arguments, you use the #define command with a list of parameters in parentheses after the name of the macro. The parameters may be any valid C identifiers separated by commas at the top level (that is, commas that aren't within parentheses) and, optionally, by white-space characters. The left parenthesis must follow the macro name immediately, with no space in between

Q.No.11 What is a process?

Answer:-

A series of actions or steps taken to achieve an end, a process is a collection of interrelated work tasks initiated in response to an event that achieves a specific result for the customer of the process.

Reference: https://www.google.com/webhp?source=search_app#hl=en&tbo=d&sclient=psy-ab&q=What+is+Process&oq=What+is+Process&gs_l=hp.3..014.1060964.1063012.3.1063209.15.14.0.1.1.2.515.2870.0j11j2j5-1.14.0.les%3B.0.0...1c.1.z147022bvT0&pbx=1&bav=on.2,or.r_gc.r_pw.r_cp.r_qf.&fp=1cb31f4b3d702abf&bpcl=39650382&biw=1024&bih=624

Q.No.12 briefly defines Modal Loop? (Page#246)

Answer:-

Modal loop is run by Modal dialogs and process message as does application message loop. That's why program execution is transfer to modal loop so the modal loop itself gets messages and dispatch message.

Q.No.13 Write windows Programming control process? (Page#120)

Answer:-

- 1 Edit Control
- 2 Static Control

Q.No.14 Explain Pointer to Constant, and constant to Painter? (Page#64)

Answer:-

Constant pointer to variable data:

```
char * const ptr = buff.           // constant pointer to variable data
*ptr = 'a';
ptr = buff2;                       // it will be an error
```

since we have declared ptr as a “constant pointer to variable data”, so we can change the contents of the place where ptr is pointing at, i.e. data but being a constant variable, the ptr value i.e. the address it contains cannot be modified.

Variable pointer to Constant data:

```
const char * ptr = buff.           //variable pointer to constant data
*ptr = 'a';                         // it will be an error
ptr = buf2;
```

Q.No.15 Write the complete syntax or "get parent function". (Page#185)

Answer:-

GetParent function returns the parent handle of the specified child. This function will be useful when the parent of the child window to use.

Syntax::

```
HWND GetParent
(
HWND hWnd // handle to child window
);
```

Q.No.16 Types of assertion and name them? (Page#49)

Answer:-

There are three types of assertion:

1 Preconditions

- Specify conditions at the start of a function.

2 Post conditions

- Specify conditions at the end of a function.

3 Invariants

- Specify conditions over a defined region of a program

Q.No.17 Write the characteristics of child windows? (Page#183)

Answer:-

Following are the characteristics of child windows.

- A child window always appears within the client area of its parent window.
- Child windows are most often as controls.
- A child window sends WM_COMMAND notification messages to its parent window.
- When a child window is created a unique identifier for that window is specified in hWnd parameter of CreateWindow()

Q.No.18 What will happen if GetUpdateRect returns zero? (Page#164)

Answer:-

An application should call the GetUpdateRect function to determine whether the window has an update region. If GetUpdateRect returns zero, the application should not call the BeginPaint and EndPaint functions.

Q.No.19 Define Client area? (Page#84)

Answer:-

The client area is the part of a window where the application displays output, such as text or graphics. For example, a desktop publishing application displays the current page of a document in the client area. The application must provide a function, called a window procedure, to process input to the window and display output in the client area.

Q.No.20 WIN MAIN describe with detail. (Page#73)

Answer:-

WinMain is the starting point in Every Win32 GUI programs. WinMain has four Parameters these are,

1. First is instance of the current application.
2. Second parameter is also an instance of this application which is used for the previous application of the same type that is already running. It is used only in Window 16bit editions or Windows 3.1. Windows 32bit editions do not support this parameter. It is here just for compatibility.
3. Third parameter is a command line argument of string type which is a type defined as char *.
4. Fourth parameter is windows style.

Qno.21 Message queues of each application. (Page#104)

Answer:-

An application must remove and process messages posted to the message queues of its threads

Q.No.22 Paint function usage. (Page#160)

Answer:-

Paint function performs following tasks.

- The BeginPaint() function prepares the specified window for painting and fills a PAINTSTRUCT structure with information about the painting.
- BeginPaint() first erases the background of window's client area by sending WM_ERASEBKGND message.
- If the function succeeds, the return value is the handle to a display device context for the specified window.

Q.No.23 Differentiate Desktop Window and Application Window. (Page#114)

Answer:-

<u>Desktop Windows</u>	<u>Application Windows</u>
When you start the system, it automatically creates the desktop window. The desktop window is a system-defined window	When you start an application, the system also associates a taskbar button with the application. The taskbar button contains the program icon and title
uses a bitmap to paint the background of the screen	Most applications also create other windows, either directly or indirectly, to perform tasks related to the main window
A system configuration application, such as a Control Panel item, changes the desktop wallpaper by using the System Parameters Info function	An application window includes elements such as a title bar, a menu bar, the window menu (formerly known as the system menu),

Q.No.24 Show the implementation of _cdecl calling convolution with respect to (Page#54)

Answer:-

- 1: Argument passing order.
- 2: stack maintenance responsibility.
- 3: name decoration convention.

Q.No.25 What happened if GetUpdateWn returns zero? (Page#164)

Answer:-

If GetUpdateRect returns zero, the application should not call the BeginPaint and EndPaint functions.

Q.No.26 A window may have more than one windows inside it explain the line. (Page#113)

Answer:-

A Window may further contain more windows inside it. For example let's take a calculator; A calculator contains more windows in forms of buttons, radio buttons and check boxes.

- Every Window has its parent and zero or more siblings.
- Top level window has desktop as its parent.

Q.No.27 What is the function of ws_paint in Windows class? (Page#113)

Answer:-

DispatchMessage function to a window procedure when the application obtains a WM_PAINT message from message Queue by using the GetMessage or PeekMessage functions.

Q.No.28 Message queuing (Page#332)

Answer:-

Message Queue is created when every any GDI function call is made or send message or post

Message function calls are made. Message Queue can be attached to every thread either it is User interface thread or worker threads. User Interface threads always a message queue

Q.No.29 Kernel tasks (Page#68)

Answer:-

Kernel is a main module of the operating system. This provides system services for managing threads, memory, and resources.

Kernel has to perform very important responsibilities e.g.

1. Process Management
2. File Management
3. Memory Management (System and Virtual Memory)

REF :: Handouts Page No.68

Q.No.30 Difference between __stdcall and __cdecl calling convention. (Page#54)

Answer:-

cdecl and __stdcall just tells the compiler whether the called function or the calling function

cleans up the stack. In __stdcall calling convention, the called function cleans up the stack when

it is about to return. So if it is called in a bunch of different places, all of those calls do not need

to extra code to clean up the stack after the function call.

In __cdecl calling convention, it is the caller function that is responsible for cleaning the stack, so every function call must also need to include extra code to clean up the stack after the function call.

Q.No.31 Erase window function explanation. (Page#154)

Answer:-

Erase windows uses the GetClipBox function to retrieve the logical coordinates of the area to erase and passes these coordinates to the FillRect function. Applications that process these messages can use similar techniques. The system supplies a window device context with the WM_ICONERASEBKGND message regardless of whether the corresponding window has a private device context.

Q.No.32 Clipboard Working (Page#72)

Answer:-

In Windows, data is shareable among applications. We can use it for copying the data from one file to the other in same format e.g. from notepad to MS Word.. All the text or image data you have previously copied can now be pasted in other application.

Q.No.33 Write down complete syntax of "getDC" function? (Page#151)

Answer:-

The system retrieves a device context from the cache whenever an application calls the GetDC or BeginPaint function; the system returns the DC to the cache when the application subsequently calls the ReleaseDC or EndPaint function.

Q.No.34 infinite recursion

Answer:-

Infinite recursion, a special case of an infinite loop that is caused by recursion. This revised function will only run out of stack space if n is less than 1 or n is too large; error checking would remove the first case. For information on recursive functions which never run out of stack.

Reference: http://en.wikipedia.org/wiki/Infinite_loop#Infinite_recursion

Q.No.35 What is stack?

Answer:-

In computer science, a stack is an area of memory that holds all local variables and parameter used by any function, and remembers the order in which functions are called so that function returns occur correctly.

Reference: [http://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](http://en.wikipedia.org/wiki/Stack_(abstract_data_type))

Q.No.36 What is extern storage class? (Page#60)

Answer:-

Extern defines a global variable that is visible to all object modules. When you use 'extern' the variable cannot be initialized as all it does is to point the variable name at a storage location that has been previously defined

Q.No.37 An application can set up for itself any logical coordinates system, using API. Write down any two. (Page#70)

Answer:-

There are two types of brushes: logical and physical. A logical brush is one that you define in code as the ideal combination of colors and/or pattern that an application should use to paint shapes. A physical brush is one that a device driver creates, which is based on your logical-brush definition.

Q.No.38 What happened if an application does not process WM_ErasebkGrd messages but pass it defWindowProcs.(5) (Page#167)

Answer:-

If it processes **WM_ERASEBKGND**, the application should use the message's *wParam* parameter to draw the background. This parameter contains a handle to the display device context for the window. After drawing the background, the application should return a nonzero value. This ensures that **BeginPaint** does not erroneously set the **fErase** member of the **PAINTSTRUCT** structure to a nonzero value (indicating the background should be erased) when the application processes the subsequent **WM_PAINT** message.

An application can define a class background brush by assigning a brush handle or a system color value to the **hbrBackground** member of the **WNDCLASS** structure when registering the class with the **RegisterClass** function.

The **GetObject** or **CreateSolidBrush** function can be used to create a brush handle. A system color value can be one of those defined for the **SetSysColors** function. (The value must be increased by one before it is assigned to the member.)

Q.No.39 Define "Virtual-Key" message - (Page#192)

Answer:-

Virtual-key code is a device-independent value defined by the system that identifies the purpose of a key. After translating a scan code, the keyboard layout creates a message that includes the scan code, the virtual-key code, and other information about the keystroke, and then places the message in the system message queue.

Qno.39 Differentiate between Super Classing and Sub Classing. (Page#132)

Answer:-

SuperClassing	SubClassing
Super-classing defines a class that adds new functionality to a predefined window class,	Sub-classing is allowed only within a process.
Button or list box controls.	Win32 processes have separate address spaces
Super-classing involves creating a new class that uses the window procedure of an existing class for basic functionality.	An application cannot subclass a window or class that belongs to another process.

Q.No.40 Why the entries in the parent process table and child table?

Answer:-

It means that the handle value that identifies a kernel object is identical in both the parent and the child processes

Reference: <http://www.coolinterview.com/interview/54829/>

Q.No.41 Define instance Handle? Briefly explain. (Page#90)

Answer:-

This member is Application instance handle.

Q.No.42 List down the important distance between subclassing and superclassing? (Page#132)

Answer:-

SuperClassing	SubClassing
Super-classing defines a class that adds new functionality to a predefined window class,	<i>Subclassing is allowed only within a process.</i>
Button or list box controls.	Win32 processes have separate address spaces
<i>Superclassing</i> involves creating a new class that uses the window procedure of an existing class for basic functionality.	<i>An application cannot subclass a window or class that belongs to another process.</i>

Q.No.43 Write down the code c/c++ program that has 2 functions on take 4 integer variable as parameters and return their sum and other take 4 integer as arguments and return their multiplication.

Answer:-

```
#include <stdio.h>

// prototypes, the parameter names are optional,
// print the menu and obtains a selection
int GetChoice();
// inserts a number in the sorted array.
int AddNum (int num[], int sz);
// removes a number from the array.
int DelNum(int num[], int sz);
// prints out the sorted array
void PrintNums(int num[], int sz);

void main()
{
    // the number of elements in array A[]
    int Size = 0,
    // the array that will keep its numbers sorted
    A[20],
    // the selection made from the menu
    Selection;
    // keep doing this loop until a 4 or Quit is
    // selected from the menu...
    Selection = GetChoice();
    for( ; Selection != 4; )
    {
        if(Selection == 1)
            Size = AddNum(A, Size);
        else
            if(Selection == 2)
                Size = DelNum(A, Size);
            else
                PrintNums(A, Size);
        Selection = GetChoice();
    }
}

// assuming that user will enter only a valid int
```

```

int GetChoice()
{
    int Choice;

    printf("Enter 1 - Insert, 2 - Delete, ");
    printf("3 - List and 4 - Quit: ");
    scanf_s("%d", &Choice);
    return Choice;
}

int AddNum(int Num[ ], int sz)
{
    // local variables...
    int i, j, Number;
    // gets the number to insert
    printf("What number to insert? ");
    scanf_s("%d", &Number);
    // finds the place (i) to put the new number
    for(i = 0; i < sz; ++i)
        if(Number < Num[i])
            break;
    // shift the rest of the array by moving
    // the numbers up by one slot.
    for(j = sz; j > i; --j)
        Num[j] = Num[j - 1];
    // place the new number
    Num[i] = Number;
    // the array size is incremented
    // and return to the calling function
    return ++sz;
}

// assuming that the deleted item exists
int DelNum(int Num[], int sz)
{
    // local variables...
    int i, Number;
    // gets the number to be deleted
    printf("What number are you going to delete? ");
    scanf_s("%d", &Number);
    // find the place in the array to be deleted
    for(i = 0; i < sz; ++i)
        if(Number == Num[i])

```

```

        break;
    // the array size is decremented
    --SZ;
    // shift the rest of the array by moving the numbers
    // down by one slot
    for( ; i < sz; ++i)
        Num[i] = Num[i + 1];
    // return sz to the calling function
    return sz;
}

void PrintNums(int Num[ ], int sz)
{
    // local variable...
    int i;

    for(i = 0; i < sz; ++i)
        printf("Num[%d] = %d\t", i, Num[i]);
    printf("\n");
}

```

Output of this Program

```

C:\WINDOWS\system32\cmd.exe
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 1
What number to insert? 40
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 1
What number to insert? 20
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 1
What number to insert? 60
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 3
Num[0] = 20    Num[1] = 40    Num[2] = 60
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 2
What number are you going to delete? 40
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 3
Num[0] = 20    Num[1] = 60
Enter 1 - Insert, 2 - Delete, 3 - List and 4 - Quit: 4
Press any key to continue . . .

```

Reference: <http://www.tenouk.com/clabworksheet/labworksheet13.html>

Q.No.44 Write two macros that perform the same tasks as these functions perform?
(Page#40)

Answer:-

A *macro* is a fragment of code which has been given a name. Whenever the name is used, it is replaced by the contents of the macro. There are two kinds of macros. They differ mostly in what they look like when they are used. *Object-like* macros resemble data objects when used, *function-like* macros resemble function calls.

You may define any valid identifier as a macro, even if it is a C keyword. The preprocessor does not know anything about keywords. This can be useful if you wish to hide a keyword such as `const` from an older compiler that does not understand it. However, the preprocessor operator `defined` can never be defined as a macro, and C++'s named operators cannot be macros when you are compiling C++.

To define a macro that takes arguments, you use the `#define` command with a list of parameters in parentheses after the name of the macro. The parameters may be any valid C identifiers separated by commas at the top level (that is, commas that aren't within parentheses) and, optionally, by white-space characters. The left parenthesis must follow the macro name immediately, with no space in between.

For example, here's a macro that computes the maximum of two numeric values:

```
#define min(X, Y) ((X)>(Y) ? (X):(Y))
```

Q.No.45 Two common controls in window programming. (Page#121)

Answer:-

Common controls are of these types.

1. Date Time Picker Control.
2. List View Control.

Q.No.46 Write at least two tasks performed by BeginPaint () function. (Page#160)

Answer:-

- 1 The BeginPaint() function prepares the specified window for painting and fills a PAINTSTRUCT structure with information about the painting.
- 2 BeginPaint() first erases the background of window's client area by sending WM_ERASEBKGND message.

Qno.47 about _cdecl calling convention? (Page#54)

Answer:-

The _cdecl is the default calling convention for C programs. In this calling convention, the stack is cleaned up by the caller. The _cdecl calling convention creates larger executables than __stdcall, because it requires each function call to include stack cleanup code.

Best of Luck