1.----------is used to check the predefined identifiers.

1. #include

2. #ifdef ANS

3. #def

4. #elif

Microsoft Windows consists of three important components. These are:

1. Kernel

2. Pointer

3. User

Option

a. I

b. Ii

c. I and iii Ans

d. I AND ii

--------function to convert the message into character messages.

Ans: TranslateMessage

If you pass WS_CAPTION style to create window(),-------style will be included automatically in it.

Ans WS_BORDER style

WM_PAINT tells the window procedure that the window's client area has changed and must be repainted

Q:1 Define the structure of PAINTSTRUCT

Answer: chapter no : 14 , Page # 168

The PAINTSTRUCT structure contains information for an application. This information can be used to paint the client area of a window owned by that application.

typedef struct tagPAINTSTRUCT {

HDC hdc; //Handle to the Device context

BOOL fErase; /*erase back ground of this parameter is true*/

RECT rcPaint; /*rectangle to the invalidate region*/

BOOL fRestore;

BOOL fIncUpdate; //updation true/false

BYTE rgbReserved[32]; //rgb values

} PAINTSTRUCT, *PPAINTSTRUCT;

Q2: Define window according MICROSOFT.

Answer: **chapter no : 1 , Page # 3**

On November 10, 1983, Microsoft announced Microsoft Windows, an extension of the MS-DOS® operating system that would provide a graphical operating environment for PC users. Microsoft called Windows 1.0 a new software environment for developing and running applications that use bitmap displays and mouse pointing devices. With Windows, the graphical user interface (GUI) era at Microsoft had begun.

Q3: Define Macro types with example

Answer: **chapter no : 5 , Page # 30**

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

#define min(X, Y) ((X)>(Y) -→ (X):(Y))

Q4: write the structure of CREATWINDOW and tell about parameters which are used in ?

Answer: **chapter no : 9 , Page # 80**

HWND CreateWindow(

LPCTSTR lpClassName; //class name (identification)

LPCTSTR lpWindowName; //Window caption bar Name

DWORD dwStyle; // style of the windows

Int x; //starting X point of window on screen

Int y; //starting Y point of window on screen

Int width; //Width of the window from starting point

Int height; //height of the window from starting Y point

HWND hWndParent; //handle the parent window if any

HMENU hMenu; // handle the Menu if any

HINSTANCE hInstance; //handle of the instance LPVOID lpParam; //void parameter

);

Q5: define virtual key function mesg?

Answer: **chapter no : 10 , Page # 100**

A virtual-key message contains a virtual-key code that identifies which key was pressed, but not its character value. To retrieve this value, the message loop must contain TranslateMessage, which translates the virtual-key message into a character message (WM_CHAR) and places it back into the application message queue

Q6 : Define typedef

Answer: **chapter no : 6 , Page # 38**

A typedef declaration lets you define your own identifiers that can be used in place of type specifiers such as int, float, and double. The names you define using typedef are NOT new data types. They are synonyms for the data types or combinations of data types

they represent. The following statements declare LENGTH as a synonym for int, then use this typedef to declare length, width, and height as integral variables.

typedef int LENGTH;

LENGTH length, width, height;

q3:steps to handle message generate against any action

Q7: Define clipboard

Answer: **chapter no : 8 , Page # 65**

- User32.dll manages clipboard.
- Clipboard is used to cut copy and paste operations.
- Clipboard is temporary storage area. When you shut down windows, data saved in

clipboard will be lost.

Q8: Write at least two responsibility of GDI?

Answer: **chapter no : 8 , Page # 61**

GDI is a subsystem responsible for displaying text and images on display devices and printers.

Q9: In 32 bit version of the Microsoft wind, when a window is created, which structure is used by operating system?

Answer: **chapter no : 9 , Page # 80**

HWND CreateWindow(

LPCTSTR lpClassName; //class name (identification)

LPCTSTR lpWindowName; //Window caption bar Name

DWORD dwStyle; // style of the windows

Int x; //starting X point of window on screen

Int y; //starting Y point of window on screen

Int width; //Width of the window from starting point

Int height; //height of the window from starting Y point

HWND hWndParent; //handle the parent window if any

HMENU hMenu; // handle the Menu if any

HINSTANCE hInstance; //handle of the instance LPVOID lpParam; //void parameter

);

Q10 : Program executed or not .Give reason .code given below

```
#include<iostream.h>
#include<conio.h>
Const int z=6
Void sum(int a,int b)
{
```

```
Z=a;
Coutz+b;   }
Int main()
{
Sum(5,9);
Getch();
}
```

Answer: code isn't complete

Q11 .Can we pass argument to Macro? Explain with example? 5 marks

Answer: chapter no : 5 , Page # 30

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 12:

Separate the system window class that can be used by the user process from system window classes that only use by the system

i.Button	**user processes**
ii.DDEMLEvent	**system**
iii.Mesage	**system**
iv. ScrolBar	**user processes**
v. ComboLBOx	**system**

Q 13: What is window ? how many components its have write the name of components of window

Answer: chapter no : 1 , Page # 3

On November 10, 1983, Microsoft announced Microsoft Windows, an extension of the MS-DOS® operating system that would provide a graphical operating environment for PC users. Microsoft called Windows 1.0 a new software environment for developing and running applications that use bitmap displays and mouse pointing devices. With Windows, the graphical user interface (GUI) era at Microsoft had begun.

Microsoft Windows consists of three important components. These are:

1. Kernel

2. GDI (Graphics Device Interface)

3. User

Q.14: How can we select the specific msg from queue

Answer: ch#9 Page no : 88

BOOL GetMessage()

(

LPMSG lpMsg,

HWND hWnd,

UINT wMsgFilterMin,

UINT wMsgFilterMax

)

Q.15: Write three conditions in which paint msg sent?

Answer:

• Any hidden part of window becomes visible Window is resized (and CS_VREDRAW, CS_HREDRAW style bits were set while registering the window class).

• Program scrolls its window.

• Invalidate Rector InvalidateRgnis called by the application.

Q.16: Write the syntax of send message function and also write what will happen when send message function is used?

Answer: Chapter:10 Page no : 103

An application typically sends a message to notify a window procedure to perform a task immediately. The **SendMessage** function sends the message to the window procedure corresponding to the given window. The function waits until the window procedure completes

processing and then returns the message result. Parent and child windows often communicate by sending messages to each other.

Q.17: Write the name of the two type of implicit? 2 marks

Answer: Chapter:6 Page no : 39

Implicit type casting (coercion) is further divided in two types

- Promotion
- Demotion

Q.18: Super classing se related ek question jo scenario based tha

Answer: #not sure as q is not so clear

After implementation of Sub-classing or Super Classing don't forget to call window

procedure function.

Q.19: Write a 5 line code for handle when the massage is……bhool gaya :P

Answer b bhol gya :D

Q.20: Code for window procedure while creating window

Answer:

Repeated

Q.21: Type of brushes and briefly explain?

Answer: Chapter#8, Page no : 63

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.22: program to write in c/c++.of multiplication 4 integer taken as parameter and taken as argument . Also make two macro of Multiplication 4 integer (5 Marks)

Answer: #not sure about macro stuff

```
#include<iostream.h>
#include<conio.h>
#define mul(a, b)(c=a*b)
#define mul1(x, y)(d=x*y)
void main()
{
int a,b,mul1,x,y,mul;
cout<<"Enter the first no."<<endl;
cin>>a;
cout<<"Enter the second no."<<endl;
cin>>b;
cout<<"Enter the 3rd no."<<endl;
cin>>x;
cout<<"Enter the 4th no."<<endl;
cin>>y;
mul=a*b*x*y;
cout<<"multiplication of four numbers are "<<mul;
getch();
}
```

Q.23: SubClassing respond at which three manners when got message . (3 marks)

Answer: Chapter#12 Page no : 129

When application subclasses a window, it can take three actions with the message:

(1) Pass the message to the original window procedure;

(2) Modify the message and pass it to the original window procedure;

(3) Not pass the message.

Q.24: Make a statement of "Variable pointer to const data "(2 Marks)

Answer: Chapter :7 Page no : 56

- Variable pointer to Constant data:

const char * ptr = buff. //variable pointer to constant data

*ptr = 'a'; // it will be an error ptr = buf2;

Here, ptr has been declared as "variable pointer to constant data". In this case, the data to

which the ptr is pointing to remains constant and cannot be modified after initialization.

The contents of ptr (address) are variable and we can change the contents of ptr

Q.25: Scenario was given of three Multi Dimension array . need to declare that 3 kind of data in Array (Marks 3)

Answer: Chapter#3 Page : 15

int provinces[50][500][1000];

// This will declare a three dimensional array.

Q.26: How do i translate GETLASTERROR()into string ? (5 marks)

Answer: Chapter# 9 Pageno: 89

To get extended error information, use GetLastError function.

Q.27: .I have a tab-based application for windows. I would like to add a subtle gradient to the back ground of my tab control. How would I go around doing this? What is the best method for me to Use? 5 marks

Answer: #internet

To use GDI you'll need the GradientFill function. You can also use GDI+ to get gradients. Here's a plain GDI example:

```
TRIVERTEX       vert[2] ;
GRADIENT_RECT   gRect;
vert [0] .x     = 0;
vert [0] .y     = 0;
```

```
vert [0] .Red    = 0x0000;
vert [0] .Green  = 0x0000;
vert [0] .Blue   = 0x0000;
vert [0] .Alpha  = 0x0000;

vert [1] .x      = 100;
vert [1] .y      = 32;
vert [1] .Red    = 0x0000;
vert [1] .Green  = 0x0000;
vert [1] .Blue   = 0xff00;
vert [1] .Alpha  = 0x0000;

gRect.UpperLeft  = 0;
gRect.LowerRight = 1;
GradientFill(hdc,vert,2,&gRect,1,GRADIENT_FILL_RECT_H);
```

As for the tab control, you could sub-class the control and override its non-client and client drawing handlers to render the gradient.

To sub class a control, first create the control and then replace its WNDPROC function:

```
OldWndProc = (WNDPROC)SetWindowLongPtr (hControl, GWLP_WNDPROC,
(LONG_PTR)NewWndProc);
```

then, in your new WNDPROC:

```
NewWndProc (usual args)
{
 switch message
 {
 case paint:
   draw gradient
   return result

 default:
   return CallWindowProc (OldWndProc, ..args..); <- important!
 }
}
```

Q.28 : Write a program using WinMain function in which you will display a message box. Message Box display "**just Hello world program !"** in this title bar (title bar of the message box in given) 5.marks

Answer:

```
#define WIN32_LEAN_AND_MEAN

#include <windows.h>       // the main windows headers
#include <windowsx.h>      // a lot of cool macros

// main entry point for all windows programs
int WINAPI WinMain(HINSTANCE hinstance,
HINSTANCE hprevinstance,
LPSTR lpcmdline,
int ncmdshow)
{
// call message box api with NULL for parent window handle
MessageBox(NULL, "Hello World!",
"JUST ANOTHER HELLO WORLD PROGRAM",
MB_OK | MB_ICONEXCLAMATION);
// exit program
return(0);
} // end WinMain
```

Q.29: How an application get handle to Stock Object? 3 marks

Answer: **Chapter#13 Pageno: 142**

An application calls the GetStockObject function to get a handle to a stock object, and the returned handle is then used as a standard object handle.

Q.30: What is the use of "SendMessage()"and "Post Message()"functions? Also give the difference between them 3 marks

Answer: **Chapter# 10 Pageno:103**

SendMessage():

The SendMessage function sends the message to the window procedure corresponding to the given window. The function waits until the window procedure completes processing and then returns the message result.

Post Message():

An application typically posts a message to notify a specific window to perform a task. PostMessage creates an MSG structure for the message and copies the message to the message queue. The application's message loop eventually retrieves the message and dispatches it to the appropriate window procedure.

Q.31: Enlist two responsibilities of kernel? 2 marks

Answer: **Chapter# 8 Page no: 61**

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)

Q.32: Some Times before selecting a new GDI object in any Device Context we keep previously selected Object stored ,why ? 2 marks

Answer: **Chapter# 13 Page no: 144**

Common practice is to track the original object that was selected into the DC and **select it back** when all work is accomplished with the new object.

2-Marks

Q.33: Define realization process?

Answer: **Chapter# 13 Page no: 144**

Objects are converted from logical objects to physical objects using the realization process. Selecting a logical object into a DC involves converting the logical object into a physical object that the device driver uses for output. This process is called realization.

Q.34: Create a Child window of type Button, with

- Title “Select”
- Horizontal position: 70, vertical position: 150
- Width:150 height 200

Answer: **#notsure(really)**

```
DWORD dwStyle[WS_CHILD];

Hwnd  hwndParent;

hWnd = CreateWindow( "BUTTON", "Title",

WS_VISIBLE | WS_OVERLAPPEDWINDOW | WS_CAPTION, 70, 150, 150, 200,

NULL, NULL, hInstance, NULL);
```

Q.35: accoding to __cdecl calling convention

1. Argument Order : **Right to left**
2. Stack maintains responsibility : **Calling function pops up argument from the stack**
3. Name-decoration convention : **_ underscore is used as prefix**

Q.36: Name any five elements include in application window?

Answer: **Page no: 79**

An application window includes elements such as a title bar, a menu bar, the window menu (formerly known as the system menu), the minimize button, the maximize button, the restore button, the close button, a sizing border, a client area, a horizontal scroll bar, and a vertical scroll bar.