

Model Answer  
Of  
Multimedia system design (IT4111)  
B.tech 7<sup>th</sup> semester

**Multiple Choice**

1. Graphic programs widely used in the graphic arts profession include \_\_\_\_\_
- a) desktop publishing programs, image editors and illustration programs
  - b) artificial intelligence, virtual reality, and illustration programs
  - c) megamedia programs, image editors, and desktop publishing programs
  - d) virtual reality, desktop publishing programs, and illustration programs

Answer: A

Response: The graphic arts profession would use desktop publishing, image editors and illustrator programs in their work.

2. Which of the following programs is not a popular desktop publishing program?
- a) Adobe PageMaker
  - b) Microsoft Publisher
  - c) Lotus AmiPro
  - d) QuarkXPress

Answer: C

Response: Lotus AmiPro is a word processing package.

3. Programs used to create or modify bitmap images are called \_\_\_\_\_
- a) illustration programs
  - b) paint programs
  - c) graphical modifiers
  - d) bit publishing packages

Answer: B

Response: Image editors (also known as paint programs) are programs for creating and editing bitmap images.

4. Paint programs and image editors are used for creating and editing \_\_\_\_\_
- a) bitmap images
  - b) vector images
  - c) text
  - d) HTML codes

Answer: A

Response: Paint programs and image editors are programs for creating and editing bitmap images.

5. Raster images are also known as
- a) bitmap images
  - b) vector images
  - c) clip art images
  - d) multimedia images

Answer: A

Response: Raster images are also known as bitmap images.

6. Images made up of thousands of pixels are called \_\_\_\_\_

- a) bitmap
- b) vector
- c) story boards
- d) graphics

Answer: A

Response: Bitmap images use thousands of dots or pixels to represent images.

7. Which of the following programs is not a popular professional image editor program?

- a) Adobe PageMaker
- b) Microsoft Paint
- c) Adobe Photoshop
- d) Corel PhotoPaint

Answer: A

Difficulty level: Medium

Page: 96

Response: Adobe PageMaker is a desktop publishing package.

8. Vector images are \_\_\_\_\_

- a) composed of pixels
- b) composed of thousands of dots
- c) slightly more difficult to manipulate than other images

d) composed of objects such as lines, rectangles, and ovals

Answer: D

Response: Vector images use geometric shapes or objects.

9. Programs used to create or modify vector images are called \_\_\_\_\_

- a) illustration programs
- b) image editors
- c) graphical modifiers
- d) bit publishing packages

Answer: A

Response: Illustration programs (also known as drawing programs) are used to create and to edit vector images.

10. CorelDraw is an example of a(n) \_\_\_\_\_

- a) groupware application
- b) bit publishing package
- c) paint program
- d) graphics suite

Answer: D

Q.2

B B Short Answer Type

(a)What are the Multimedia Elements?

Ans: THE FIVE MAIN ELEMENTS OF MULTIMEDIA

## TEXT

Text is the basic element of multimedia. It involves the use of text types, sizes, colours and background colour. In a multimedia application, other media or screen can be linked through the use of text. This is what you call Hypertext. To produce an effective multimedia program there are three things that need to be considered. They are:

- The position of the text on the screen.
- Length of the message
- And legibility of the text.

## GRAPHIC

Graphics make the multimedia application attractive. They help to illustrate ideas through still pictures.

There are two types of graphics used: bitmaps (paint graphics) and vector (draw graphics).

## AUDIO

A multimedia application may require the use of speech, music and sound effects. These are called audio or the sound element. There are two basic types of audio or sound: analog and digital audio.

## VIDEO

Video provides a powerful impact in a multimedia program. In multimedia applications, the digital video is gaining popularity because of the following reasons:

- Video clips can be edited easily. The digital video files can be stored like any other files in the computer and the quality of the video can still be maintained. The video files can be transferred within a computer network it allows non-linear editing in any part of the video. However, these digital video files are large in size. Transferring these files can take a long time especially when using the Internet.

## ANIMATION

Animation is a process of making a static image look like it is moving. In multimedia, digital animation is used. Digital animation can be categorised into two broad areas: 2D (2 Dimension) and 3D (3 Dimension) animations.

(b) What is the Fractal Compression?

**Fractal compression** is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image.

(c) What are the Hierarchical Storage Management?

HSM (Hierarchical Storage Management) is policy-based management of file backup and archiving in a way that uses storage devices economically and without the user needing to be aware of when files are being retrieved from backup storage media. Although HSM can be implemented on a standalone system, it is more frequently used in the distributed network of an enterprise. The hierarchy represents different types of storage media, such as redundant array of independent disks systems, optical storage, or tape, each type representing a different level of cost and speed of retrieval when access is needed. For example, as a file ages in an archive, it can be automatically moved to a slower but less expensive form of storage. Using an HSM product, an administrator can establish and state guidelines for how often different kinds of files are to be copied to a backup storage device. Once the guideline has been set up, the HSM software manages everything automatically.

(d) Describe JPEG, DIP File Format.

JPEG: JPEG (Joint Photographic Experts Group) is an ISO/IEC group of experts that develops and maintains standards for a suite of compression algorithms for computer image files. JPEG (usually pronounced JAY-pehg) is also a term for any graphic image file produced by using a JPEG standard. A JPEG file is created by choosing from a range of compression qualities (actually, from one of a suite of compression algorithms). When you create a JPEG or convert an image from another format to a JPEG, you are asked to specify the quality of image you want. Since the highest quality results in the largest file, you can make a trade-off between image quality and file size. Formally, the JPEG file format is specified in ISO standard 10918. The JPEG scheme includes 29 distinct coding processes although a JPEG implementer may not use them all.

### ***File Extension DIP Description:***

The DIP file extension is associated with the software, DVDINFOPro, and is for files that contain the data graph of a burned DVD. The DIP file extension was created by Nic Wilson to show information after burning of a DVD in the program. DVDINFOPro provides tools and information for HD-DVD, Blu-Ray, DVD-R and RW, DVD players, and DVD burners.

Another application that uses the DIP file extension is Dr. Watson, which is included in Microsoft Windows Operating Systems. The purpose of files in the .dip file extension is to collect error information or symptoms, after a problem that occurred in the system. This file format defines debug information processing files, for data obtained and logged by the Dr. Watson program. Files in the .dip file extension processes essential information that are necessary for technical support to diagnose errors in the computer running in Windows.

(e) What are the multimedia operating system components?

**Very High Processing Power**

-- needed to deal with large data processing and real time delivery of media. Special hardware commonplace.

**Multimedia Capable File System**

-- needed to deliver real-time media -- *e.g.* Video/Audio Streaming. Special Hardware/Software needed *e.g.* RAID technology.

**Data Representations/File Formats that support multimedia**

-- Data representations/file formats should be easy to handle yet allow for compression/decompression in real-time.

**Efficient and High I/O**

-- input and output to the file subsystem needs to be efficient and fast. Needs to allow for real-time recording as well as playback of data. *e.g.* Direct to Disk recording systems.

**Special Operating System**

-- to allow access to file system and process data efficiently and quickly. Needs to support direct transfers to disk, real-time scheduling, fast interrupt processing, I/O streaming *etc.*

**Storage and Memory**

-- large storage units (of the order of 50 -100 Gb or more) and large memory (50 - 100 Mb or more). Large Caches also required and frequently of Level 2 and 3 hierarchy for efficient management.

**Network Support**

-- Client-server systems common as distributed systems common.

**Software Tools**

-- user friendly tools needed to handle media, design and develop applications, deliver media.

**Section B****Unit 1**

3: Describe a multimedia Application used in your college or work environment if there is no such application, describe what activities can benefit from multimedia applications.

By incorporating multimedia in their instruction, teachers can capture attention, engage learners, explain difficult concepts, inspire creativity, and have fun. However, there are many tools available and many ways to use those tools. If you are a teacher searching for a technology tool to accomplish certain learning goals or outcomes, you can easily be overwhelmed. Given the limited amount of time teachers have during the day, why should you take the time to learn about and use these tools? I'll try to answer this question.

Following is a list of ten reasons you should use multimedia in your classroom. As you investigate and try these and other tools, you will notice that they are social–encouraging sharing, feedback, publication, and other experiences that support learning. By modelling

social learning and learning with technology, you will be preparing your students for success in today's rapidly changing world.

**1. Facilitate and develop a community of learners through online ice-breaker activities.** These activities offer fun and easy ways to get to know each other while also providing outlets for student creativity. A neat tool that works well for this is [VoiceThread](#). Students can use a computer web-cam to record a video of themselves and view other students videos, all on one page.

**2. Help students visualize difficult concepts or procedures more easily** by using static or dynamic multimedia. I have used a very simple and efficient software called [ScreenSteps](#), which allows you to quickly create visual handouts for learners. Teachers and students can use [Jing software](#) to record a screen shot or video, which produces a video tutorial or information about a website, embedding the video on their website or sending it to the student as an email attachment. These types of software provide a great way for teachers to make the most out of their multimedia and online resources.

**3. Scaffold learning through activities enhanced by videos and online games.** When assigning reading about an obscure historical event, you might want to create pre-reading activities by having students watch and comment on videos that fill in needed background knowledge. Searching for videos about events on Global Grid for Learning ([GGfL](#)), for instance, can provide needed support and add to a student's gap in knowledge. Then you could embed these videos on your class website, blog, or wiki. Or have students add to a playlist as they locate more videos on the topic.

**4. Make language and culture come alive** through the viewing and creation of audio and video instruction. Students could view videos and television programs available online and stay up to date on current events in that country. They could also create their own videos and share them with another class, comparing cultural norms and addressing other questions through a group blog or wiki.

**5. Provide a "menu" of authentic assignment options** for students to complete, allowing them to explore and identify their passions and talents. Encourage them to create and publish an original digital story. Have them produce a weekly podcast show for the classroom, highlighting events of the week, using blogs. They might also want to film their developing skills in a sport or record their progress in learning a musical instrument.

**6. Enhance accessibility through the use of powerful multimedia software tools.** Encourage students to use a speech-to-text tool to write their next essay or short story. This is especially helpful for students who have fine motor challenges or students who have trouble with keyboarding. Use auto-captioning features to create accessible multimedia for students with hearing challenges.

**7. Enable visualization of concepts and their connections** through collaborative construction and discussion of concept maps. One of my all-time favourites is [CMap](#)



[tools](#), a free, multi-platform software tool that can be downloaded to your computer. Students could work in groups, constructing a concept map and even recording within CMap tools this construction.

**8. Encourage collaboration and feedback** by integrating assignments with tools that support conversations and comments. For instance, have students post their slideshows and have them view their classmates' presentations, and post comments. Or have students create video comments on video sharing sites such as [TeacherTube](#). Use collaborative software such as blogs and wikis for students to easily create, edit, and publish their work. And make sure you provide information for parents to access these social media sites to see what their children are doing.

**9. Make learning situated and personal** with easy to access information from you and the rest of the world. Have students subscribe to your class Twitter and blog feeds and enable them on their mobile devices, if possible. Or, have them use a Twitter aggregator, such as [Tweetdeck](#), to stay on top of news announcements. Show them how to subscribe to dynamic sites using RSS Readers and how to read and track updated content. Have them subscribe to podcasts and rate those podcasts. Allow students to contact you using SMS.

**10. Help students document and present their learning through authentic assessments.** Instead of taking an end of term test, have students collect their work and detail their progress on their Learning Log, using any number of free blogging tools. Show them how to tag their posts, how to create categories (which could be the course objectives or standards), how to link to artefacts, how to write reflections, and then set aside time at the end of each week for reflection and documentation of their work. At the end of the term or semester, students could then refine their Learning Log, turning it into a showcase Portfolio, presenting it to the class and parents, discussing their work, what they learned, and where they want to go from there. Not only would this individualize their learning experience, but it would make students more responsible for their work and enable them to experience learning as being life-long and active.

As you can see from these ideas, you can easily align instructional goals and empower instruction through using appropriate multimedia tools. It takes some planning, time, and expenditures (video cameras, software), but in the long run, your students will reap many benefits, such as taking more responsibility for their learning, becoming aware of their learning and how to document it, and realizing their own creative potential.

#### **4. Explain the key difference between lossless and lossy compression.**

Ans:

Lossless compression reduces a file's size with no loss of quality. This seemingly magical method of reducing file sizes can be applied to both image and audio files. While JPEGs and MP3s use [lossy](#) compression, newer compression algorithms, such as JPEG 2000 and Apple Lossless compression, can be used to create lossless compressed files.

Lossless compression basically rewrites the data of the original file in a more efficient way. However, because no quality is lost, the resulting files are typically much larger than image and audio files compressed with lossy compression. For example, a file compressed using lossy compression may be one tenth the size of the original, while lossless compression is unlikely to produce a file smaller than half of the original size.

### ***Lossless vs. lossy compression***

The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than any known lossless method, while still meeting the requirements of the application.

Lossy methods are most often used for compressing sound, images or videos. The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file) of lossy video codecs are nearly always far superior to those of the audio and still-image equivalents. Audio can be compressed at 10:1 with no noticeable loss of quality, video can be compressed immensely with little visible quality loss, eg 300:1. Lossily compressed still images are often compressed to 1/10th their original size, as with audio, but the quality loss is more noticeable, especially on closer inspection.

When a user acquires a lossily-compressed file, (for example, to reduce download-time) the retrieved file can be quite different from the original at the bit level while being indistinguishable to the human ear or eye for most practical purposes. Many methods focus on the idiosyncrasies of the human anatomy, taking into account, for example, that the human eye can see only certain frequencies of light. The psycho-acoustic model describes how sound can be highly compressed without degrading the perceived quality of the sound. Flaws caused by lossy compression that are noticeable to the human eye or ear are known as compression artifacts.

Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely, but nevertheless perfectly. Lossless compression is possible because most real-world data has statistical redundancy. For example, in English text, the letter 'e' is much more common than the letter 'z', and the probability that the letter 'q' will be followed by the letter 'z' is very small.

Another kind of compression, called lossy data compression, is possible if some loss of fidelity is acceptable. For example, a person viewing a picture or television video scene might not notice if some of its finest details are removed or not represented perfectly. Similarly, two clips of audio may be perceived as the same to a listener even though one is missing details found in the other. Lossy data compression algorithms introduce relatively minor differences and represent the picture, video, or audio using fewer bits.

Lossless compression schemes are reversible so that the original data can be reconstructed, while lossy schemes accept some loss of data in order to achieve higher compression. However, lossless data compression algorithms will always fail to compress some files; indeed, any compression algorithm will necessarily fail to compress any data containing no discernible patterns. Attempts to compress data that has been compressed

already will therefore usually result in an expansion, as will attempts to compress encrypted data.

In practice, lossy data compression will also come to a point where compressing again does not work, although an extremely lossy algorithm, which for example always removes the last byte of a file, will always compress a file up to the point where it is empty.

## Unit II

Q.5. What are the benefits offered by compression schemes in designing multimedia systems?

Ans: ADVANTAGES OF DATA COMPRESSION:

Less disk space (more data in reality)

Faster writing and reading

Faster file transfer

Variable dynamic range

Byte order independent

DISADVANTAGES OF DATA COMPRESSION:

Added complication

Effect of errors in transmission

Slower for sophisticated methods (but simple methods can be faster for writing to disk).

**Following are the advantages of Video Compression:**

- Occupies less disk space.
- Reading and writing is faster.
- File transferring is faster.
- The order of bytes is independent.

Following are the disadvantages of video compression”

- Compression need to be done again for compression.

- Errors may occur while transmitting data.
- The byte / pixel relationship is unknown
- Has to decompress the previous data.

Q.6. Describe the Algorithm for the CCITT Group 3 standard how does CCITT group 4 differ from CCITT group 3?

Many facsimile and document imaging file formats support a form of lossless data compression often described as CCITT encoding. The CCITT (International Telegraph and Telephone Consultative Committee) is a standards organization that has developed a series of communications protocols for the facsimile transmission of black-and-white images over telephone lines and data networks. These protocols are known officially as the CCITT T.4 and T.6 standards but are more commonly referred to as CCITT Group 3 and Group 4 compression, respectively.

Sometimes CCITT encoding is referred to, not entirely accurately, as Huffman encoding. Huffman encoding is a simple compression algorithm introduced by David Huffman in 1952. CCITT 1-dimensional encoding, described in a subsection below, is a specific type of Huffman encoding. The other types of CCITT encodings are not, however, implementations of the Huffman scheme.

Group 3 and Group 4 encodings are compression algorithms that are specifically designed for encoding 1-bit image data. Many document and FAX file formats support Group 3 compression, and several, including [TIFF](#), also support Group 4.

Group 3 encoding was designed specifically for bi-level, black-and-white image data telecommunications. All modern FAX machines and FAX modems support Group 3 facsimile transmissions. Group 3 encoding and decoding is fast, maintains a good compression ratio for a wide variety of document data, and contains information that aids a Group 3 decoder in detecting and correcting errors without special hardware.

Group 4 is a more efficient form of bi-level compression that has almost entirely replaced the use of Group 3 in many conventional document image storage systems. (An exception is facsimile document storage systems where original Group 3 images are required to be stored in an unaltered state.)

Group 4 encoded data is approximately half the size of 1-dimensional Group 3-encoded data. Although Group 4 is fairly difficult to implement efficiently, it encodes at least as fast as Group 3 and in some implementations decodes even faster. Also, Group 4 was designed for use on data networks, so it does not contain the synchronization codes used for error detection that Group 3 does, making it a poor choice for an image transfer protocol.

Group 4 is sometimes confused with the IBM MMR (Modified Modified READ) compression method. In fact, Group 4 and MMR are almost exactly the same algorithm and achieve almost identical compression results. IBM released MMR in 1979 with the introduction of its Scanmaster product before Group 4 was standardized. MMR became IBM's own document compression standard and is still used in many IBM imaging systems today.

Document-imaging systems that store large amounts of facsimile data have adopted these CCITT compression schemes to conserve disk space. CCITT-encoded data can be decompressed quickly for printing or viewing (assuming that enough memory and CPU resources are available). The same data can also be transmitted using modem or facsimile protocol technology without needing to be encoded first.

The CCITT algorithms are non-adaptive. That is, they do not adjust the encoding algorithm to encode each bitmap with optimal efficiency. They use a fixed table of code values that were selected according to a reference set of documents containing both text and graphics. The reference set of documents were considered to be representative of documents that would be transmitted by facsimile.

Group 3 normally achieves a compression ratio of 5:1 to 8:1 on a standard 200-dpi (204x196 dpi), A4-sized document. Group 4 results are roughly twice as efficient as Group 3, achieving compression ratios upwards of 15:1 with the same document. Claims that the CCITT algorithms are capable of far better compression on standard business documents are exaggerated--largely by hardware vendors.

Because the CCITT algorithms have been optimized for type and handwritten documents, it stands to reason that images radically different in composition will not compress very well. This is all too true. Bi-level bitmaps that contain a high frequency of short runs, as typically found in digitally halftoned continuous-tone images, do not compress as well using the CCITT algorithms. Such images will usually result in a compression ratio of 3:1 or even lower, and many will actually compress to a size larger than the original.

The CCITT actually defines three algorithms for the encoding of bi-level image data:

- Group 3 One-Dimensional (G31D)
- Group 3 Two-Dimensional (G32D)
- Group 4 Two-Dimensional (G42D)

G31D is the simplest of the algorithms and the easiest to implement. For this reason, it is discussed in its entirety in the first subsection below. G32D and G42D are much more complex in their design and operation and are described only in general terms below.

The Group 3 and Group 4 algorithms are standards and therefore produce the same compression results for everybody. If you have heard any claims made to the contrary, it is for one of these reasons:

- Non-CCITT test images are being used as benchmarks.
- Proprietary modifications have been made to the algorithm.
- Pre- or post-processing is being applied to the encoded image data.
- You have been listening to a misinformed salesperson.

### Group 3 One-Dimensional (G31D)

Group 3 One-Dimensional encoding (G31D) is a variation of the Huffman keyed compression scheme. A bi-level image is composed of a series of black-and-white 1-bit pixel runs of various lengths (1 = black and 0 = white). A Group 3 encoder determines the length of a pixel run in a scan line and outputs a variable-length binary code word representing the length and color of the run. Because the code word output is shorter than the input, pixel data compression is achieved.

The run-length code words are taken from a predefined table of values representing runs of black or white pixels. This table is part of the T.4 specification and is used to encode and decode all Group 3 data.

The size of the code words were originally determined by the CCITT, based statistically on the average frequency of black-and-white runs occurring in typical type and handwritten documents. The documents included line art and were written in several different languages. Run lengths that occur more frequently are assigned smaller code words while run lengths that occur less frequently are assigned larger code words.

In printed and handwritten documents, short runs occur more frequently than long runs. Two- to 4-pixel black runs are the most frequent in occurrence. The maximum size of a run length is bounded by the maximum width of a Group 3 scan line.

Run lengths are represented by two types of code words: *makeup* and *terminating*. An encoded pixel run is made up of zero or more makeup code words and a terminating code word. Terminating code words represent shorter runs, and makeup codes represent longer runs. There are separate terminating and makeup code words for both black and white runs.

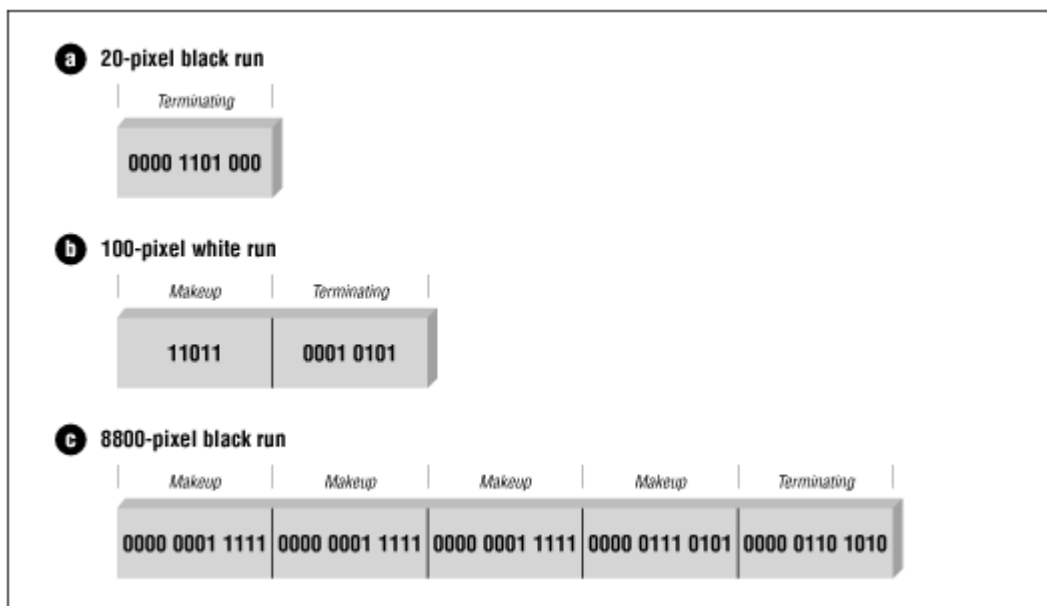
Pixel runs with a length of 0 to 63 are encoded using a single terminating code. Runs of 64 to 2623 pixels are encoded by a single makeup code and a terminating code. Run lengths greater than 2623 pixels are encoded using one or more makeup codes and a terminating code. The run length is the sum of the length values represented by each code word.

Here are some examples of several different encoded runs:

- A run of 20 black pixels would be represented by the terminating code for a black run length of 20. This reduces a 20-bit run to the size of an 11-bit code word, a compression ratio of nearly 2:1.

- A white run of 100 pixels would be encoded using the makeup code for a white run length of 64 pixels followed by the terminating code for a white run length of 36 pixels ( $64 + 36 = 100$ ). This encoding reduces 100 bits to 13 bits, or a compression ratio of over 7:1.
- A run of 8800 black pixels would be encoded as three makeup codes of 2560 black pixels (7680 pixels), a makeup code of 1088 black pixels, followed by the terminating code for 32 black pixels ( $2560 + 2560 + 2560 + 1088 + 32 = 8800$ ). In this case, we will have encoded 8800 run-length bits into five code words with a total length of 61 bits, for an approximate compression ratio of 144:1. This is illustrated below.

### CCITT Group 3 encoding



The use of run lengths encoded with multiple makeup codes has become a de facto extension to Group 3, because such encoders are necessary for images with higher resolutions. And while most Group 3 decoders do support this extension, do not expect them to do so in all cases.

Decoding Group 3 data requires methods different from most other compression schemes. Because each code word varies in length, the encoded data stream must be read one bit at a time until a code word is recognized. This can be a slow and tedious process at best. To make this job easier, a state table can be used to process the encoded data one byte at a time. This is the quickest and most efficient way to implement a CCITT decoder.

All scan lines are encoded to begin with a white run-length code word (most document image scan lines begin with white run lengths). If an actual scan line begins with a black run, a zero-length white run-length code word will be prepended to the scan line.

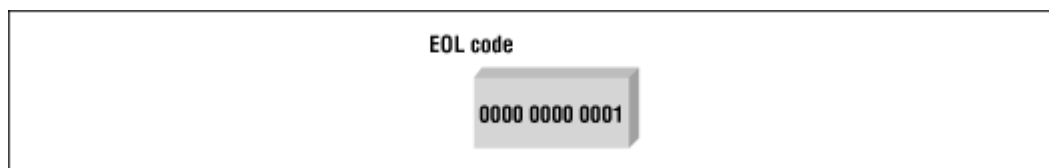
A decoder keeps track of the color of the run it is decoding. Comparing the current bit pattern to values in the opposite color bit table is wasteful. That is, if a black run is being decoded, there is no reason to check the table for white run-length codes.

Several special code words are also defined in a Group 3-encoded data stream. These codes are used to provide synchronization in the event that a phone transmission experiences a burst of noise. By recognizing this special code, a CCITT decoder may identify transmission errors and attempt to apply a recovery algorithm that approximates the lost data.

The EOL code is a 12-bit code word that begins each line in a Group 3 transmission. This unique code word is used to detect the start/end of a scan line during the image transmission. If a burst of noise temporarily corrupts the signal, a Group 3 decoder throws away the unrecognized data it receives until it encounters an EOL code. The decoder would then start receiving the transmission as normal again, assuming that the data following the EOL is the beginning of the next scan line. The decoder might also replace the bad line with a predefined set of data, such as a white scan line.

A decoder also uses EOL codes for several purposes. It uses them to keep track of the width of a decoded scan line. (An incorrect scan-line width may be an error, or it may be an indication to pad with white pixels to the EOL.) In addition, it uses EOL codes to keep track of the number of scan lines in an image, in order to detect a short image. If it finds one, it pads the remaining length with scan lines of all white pixels. A Group 3 EOL code is illustrated in below

### **CCITT Group 3 encoding (EOL code)**



Most FAX machines transmit data of an "unlimited length," in which case the decoder cannot detect how long the image is supposed to be. Also, it is faster not to transmit the all-white space at the end of a page, and many FAX machines stop when they detect that the rest of a page is all white; they expect the receiver to do white padding to the end of the negotiated page size.

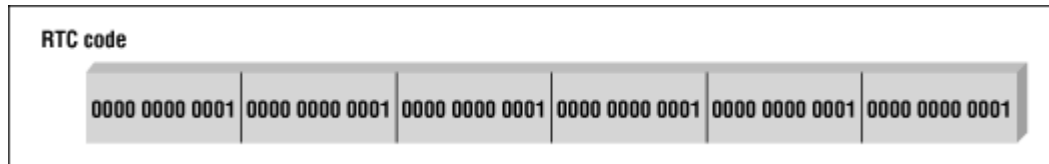
When Group 3 data is encapsulated in an image file, information regarding the length and width of the image is typically stored in the image file header and is read by the decoder prior to decoding.

Group 3 message transmissions are terminated by a return to control (RTC) code that is appended to the end of every Group 3 data stream and is used to indicate the end of the message transmission. An RTC code word is simply six EOL codes occurring consecutively. The RTC is actually part of the facsimile protocol and not part of the



encoded message data. It is used to signal the receiver that it should drop the high-speed message carrier and listen on the low-speed carrier for the post-page command. A Group 3 RTC code is illustrated below

### CCITT Group 3 encoding (RTC code)



A fill (FILL) is not actually a code word but a run of one or more zero bits that occurs between the encoded scan-line data and the EOL code (but never in the encoded scan line itself). Fill bits are used to pad out the length of an encoded scan line to increase the transmission time of the line to a required length. Fill bits may also be used to pad the RTC code word out to end on a byte boundary.

### UNIT-III

7. Compare and contrast the TIFF file formats with the RIFF File formats.

Ans:

## TIFF File Format

**Also Known As:** Tag Image File Format

---

<b>Type</b>	Bitmap
<b>Colors</b>	1- to 24-bit
<b>Compression</b>	Uncompressed, RLE, LZW, CCITT Group 3 and Group 4, JPEG
<b>Maximum Image Size</b>	2 <sup>32</sup> -1
<b>Multiple Images Per File</b>	Yes
<b>Numerical Format</b>	See article for discussion
<b>Originator</b>	Aldus
<b>Platforms</b>	MS-DOS, Macintosh, UNIX, others
<b>Supporting Applications</b>	Most paint, imaging, and desktop publishing programs

#### Usage

Used for data storage and interchange. The general nature of TIFF allows it to be used in

any operating environment, and it is found on most platforms requiring image data storage.

## ***File Organization***

TIFF files are organized into three sections: the Image File Header (IFH), the Image File Directory (IFD), and the bitmap data. Of these three sections, only the IFH and IFD are required. It is therefore quite possible to have a TIFF file that contains no bitmapped data at all, although such a file would be highly unusual. A TIFF file that contains multiple images has one IFD and one bitmap per image stored.

TIFF has a reputation for being a complicated format in part because the location of each Image File Directory and the data the IFD points to--including the bitmapped data--may vary. In fact, the only part of a TIFF file that has a fixed location is the Image File Header, which is always the first eight bytes of every TIFF file. All other data in a TIFF file is found by using information found in the IFD. Each IFD and its associated bitmap are known as a TIFF *subfile*. There is no limit to the number of subfiles a TIFF image file may contain.

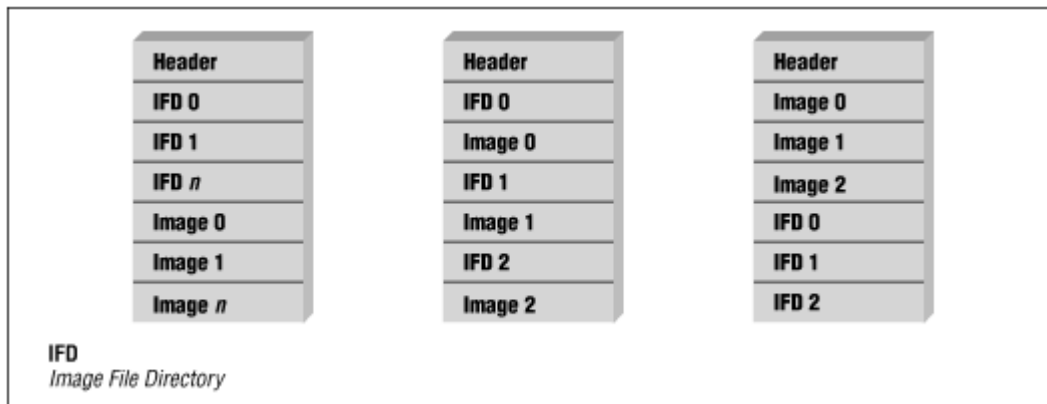
Each IFD contains one or more data structures called *tags*. Each tag is a 12-byte record that contains a specific piece of information about the bitmapped data. A tag may contain any type of data, and the TIFF specification defines over 70 tags that are used to represent specific information. Tags are always found in contiguous groups within each IFD.

Tags that are defined by the TIFF specification are called *public tags* and may not be modified outside of the parameters given in the latest TIFF specification. User-definable tags, called *private tags*, are assigned for proprietary use by software developers through the Aldus Developer's Desk. See the TIFF 6.0 specification for more information on private tags.

Note that the TIFF 6.0 specification has replaced the term *tag* with the term *field*. Field now refers to the entire 12-byte data record, while the term tag has been redefined to refer only to a field's identifying number. Because so many programmers are familiar with the older definition of the term tag, the authors have chosen to continue using tag, rather than field, in this description of TIFF to avoid confusion.

[Figure TIFF-1](#) shows three possible arrangements of the internal data structure of a TIFF file containing three images. In each example, the IFH appears first in the TIFF file. In the first example, each of the IFDs has been written to the file first and the bitmaps last. This arrangement is the most efficient for reading IFD data quickly. In the second example, each IFD is written, followed by its bitmapped data. This is perhaps the most common internal format of a multi-image TIFF file. In the last example, we see that the bitmapped data has been written first, followed by the IFDs. This seemingly unusual arrangement might occur if the bitmapped data is available to be written before the information that appears in the IFDs.

**Figure TIFF-1: Three possible physical arrangements of data in a TIFF file**



Each IFD is a road map of where all the data associated with a bitmap can be found within a TIFF file. The data is found by reading it directly from within the IFD data structure or by retrieving it from an offset location whose value is stored in the IFD. Because TIFF's internal components are linked together by offset values rather than by fixed positions, as with stream-oriented image file formats, programs that read and write TIFF files are often very complex, thus giving TIFF its reputation.

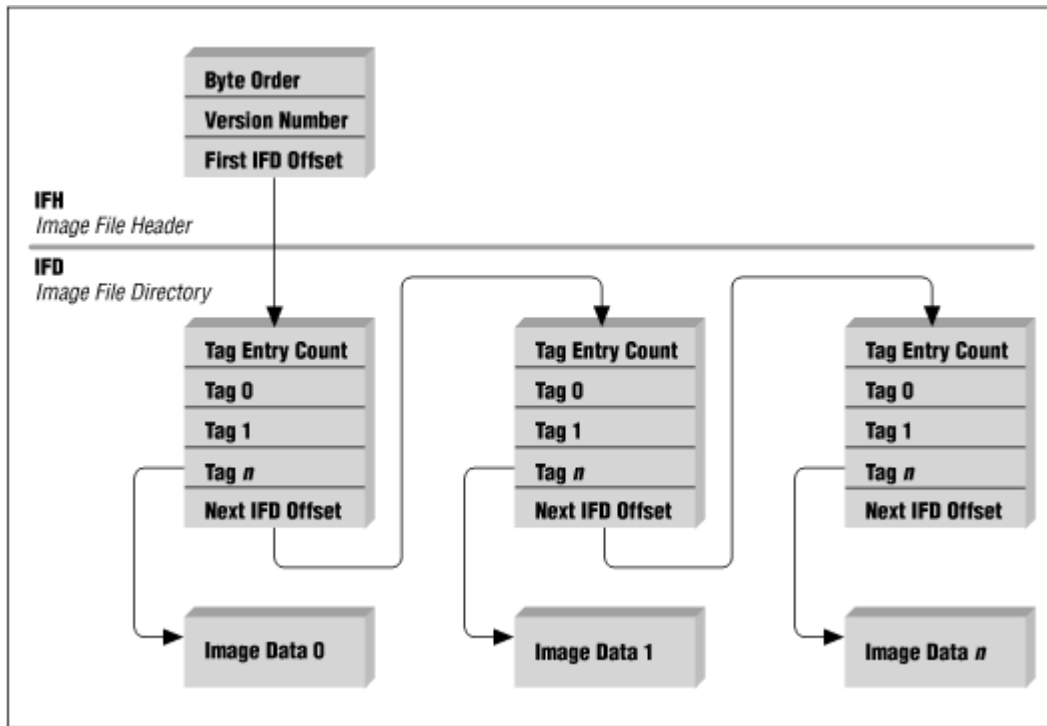
The offset values used in a TIFF file are found in three locations. The first offset value is found in the last four bytes of the header and indicates the position of the first IFD. The last four bytes of each IFD is an offset value to the next IFD. And the last four bytes of each tag may contain an offset value to the data it represents, or possibly the data itself.

**NOTE:**

**Offsets are always interpreted as a number of bytes from the beginning of the TIFF file.**

[Figure TIFF-2](#) shows the way data structures of a TIFF file are linked together.

**Figure TIFF-2: Logical organization of a TIFF file**



## ***File Details***

This section describes the various components of a TIFF file.

### **Image File Header**

TIFF, despite its complexity, has the simplest header of all of the formats described in this book. The TIFF Image File Header (IFH) contains three fields of information and is a total of only eight bytes in length:

```
typedef struct _TiffHeader
{
    WORD Identifier; /* Byte-order Identifier */
    WORD Version; /* TIFF version number (always 2Ah) */
    DWORD IFDOffset; /* Offset of the first Image File Directory*/
} TIFHEAD;
```

Identifier contains either the value 4949h (II) or 4D4Dh (MM). These values indicate whether the data in the TIFF file is written in little-endian (Intel format) or big-endian (Motorola format) order, respectively. All data encountered past the first two bytes in the file obey the byte-ordering scheme indicated by this field. These two values were chosen because they would always be the same, regardless of the byte order of the file.

Version, according to the TIFF specification, contains the version number of the TIFF format. This version number is always 42, regardless of the TIFF revision, so it may be regarded more as an identification number, (or possibly the answer to life, the universe, etc.) than a version number.

A quick way to check whether a file is indeed a TIFF file is to read the first four bytes of the file. If they are:

49h 49h 2Ah 00h

or:

4Dh 4Dh 00h 2Ah

then it's a good bet that you have a TIFF file.

IFDOffset is a 32-bit value that is the offset position of the first Image File Directory in the TIFF file. This value may be passed as a parameter to a file seek function to find the start of the image file information. If the Image File Directory occurs immediately after the header, the value of the IFDOffset field is 08h.

## RIFF File Format

**Also Known As:** RIFF, Resource Interchange File Format, RIFX, .WAV, .AVI, .BND, .RMI, .RDI

---

<b>Type</b>	Multimedia
<b>Colors</b>	24-bit
<b>Compression</b>	RLE, uncompressed, audio, video
<b>Maximum Image Size</b>	Varies
<b>Multiple Images Per File</b>	No
<b>Numerical Format</b>	Little- and big-endian
<b>Originator</b>	Microsoft Corporation
<b>Platform</b>	Microsoft Windows 3.x, Windows NT
<b>Supporting Applications</b>	Microsoft Windows and OS/2 multimedia applications

### Usage

RIFF is a device control interface and common file format native to the Microsoft Windows system. It is used to store audio, video, and graphics information used in multimedia applications.

IFF is a binary file format containing multiple nested data structures. Each data structure within a RIFF file is called a *chunk*. Chunks do not have fixed positions within a RIFF file, and therefore standard offset values cannot be used to locate their fields. A chunk contains data such as a data structure, a data stream, or another chunk called a *subchunk*. Every RIFF chunk has the following basic structure:

```
typedef struct _Chunk
{
    DWORD ChunkId;           /* Chunk ID marker */
    DWORD ChunkSize;        /* Size of the chunk data in bytes
*/
    BYTE ChunkData[ChunkSize]; /* The chunk data */
} CHUNK;
```

ChunkId contains four ASCII characters that identify the data the chunk contains. For example, the characters RIFF are used to identify chunks containing RIFF data. If an ID is smaller than four characters, it is padded on the right using spaces (ASCII 32). Note that RIFF files are written in little-endian byte order. Files written using the big-endian byte ordering scheme have the identifier RIFX.

ChunkSize is the length of the data stored in the ChunkData field, not including any padding added to the data. The size of the ChunkId and ChunkSize fields are not themselves included in this value.

ChunkData contains data that is WORD-aligned within the RIFF file. If the data is an odd length in size, an extra byte of NULL padding is added to the end of the data. The ChunkSize value does not include the length of the padding.

Subchunks also have the same structure as chunks. A subchunk is simply any chunk that is contained within another chunk. The only chunks that may contain subchunks are the RIFF file chunk RIFF and the list chunk, LIST (explained in the next section). All other chunks may contain only data.

A RIFF file itself is one entire RIFF chunk. All other chunks and subchunks in the file are contained within this chunk. If you are decoding, your RIFF reader should ignore any chunks that the reader does not recognize or it cannot use. If you are encoding, your RIFF writer will write out all unknown and unused chunks that were read. Do not discard them.

Q.8. Compare JPEG AVI file format with RIFF AVI format. Explain reason for the differences.

The **JPEG-AVI** format uses the [AVI RIFF](#) form with minor changes. Similar to the AVI, the JPEG-AVI file format uses the mandatory "hdr!" and "movi" LIST chunks. JPEG-DIB frames are [key frames](#), as they all are addressable. Their format is as specified in ISO 10918. The index chunks contain the key frames entries for each data chunk to provide [random access](#) for finding JPEG-DIB video frames from a large interleaved AVI-file.

The JPEG-AVI format is very similar to the AVI file format. The major differences are as follows -

(1) The chunk 'strf' contains EXBMINFOHEADER structure members for describing data of the stream format. (2) The Chunk "strh" contain [FOURCC](#) code "JPEG" for still images and "[MPEG](#)" for motion images with specific data for initializing the appropriate [codec](#). (3) The Chunk "movi" or "rec" contains JPEG-DIB data. (4) The Chunk "index" contains the entries for all JPEG-DIB frames as all frames are the key frames. This allows random access to JPEG-DIB frames, which was not otherwise available.

**AVI**, a short for Audio Video Interleave, is a file format designed to store both audio and video data in a standard package to allow its simultaneous playback. Created by Microsoft in November 1992, AVI is one of the most commonly used video formats. A variation of the AVI format, called DV AVI, is used to create Digital Video that can be transferred losslessly (without loss of quality) between computers and camcorders.

Using Movavi software, you can convert and edit AVI files and export videos in any other format to AVI. You can also use Movavi software as an AVI player as well.

AVI is a file format, but unlike other formats like MP3 or JPEG, AVI is a container format, meaning it can contain video audio compressed using many different combinations of codecs. So while MP3 and JPG can only contain a certain kind of compression, an AVI file may carry audio/visual data in almost any compression scheme, including: Full Frames (Uncompressed), Intel Real Time Video, Indeo, Cinepak, Motion JPEG, Editable MPEG, VDOWave, ClearVideo / RealVideo, QPEG, MPEG-4, and others. AVI all look the same on the "outside", but on the "inside", they may be completely different.

So the AVI format is dependent upon "codecs" (an acronym for compressor / decompressor). Codecs serve two purposes: they tell your computer how to encode (or compress) a video file when you save the file, and they tell your computer how to decode (or play) the file when you open the file. Since the same codec is needed to both compress the file and play the file, the choice of which codec to use is extremely important. If the codec you use to compress your file is not installed on your user's computer, your user will not be able to play the video.

No panic! Using Movavi software as an AVI player, you will never have troubles playing AVI files because all necessary codecs go together with the program!

# RIFF AVI

The Microsoft AVI file format is a RIFF file specification used with applications that capture, edit, and play back audio-video sequences. In general, AVI files contain multiple streams of different types of data. Most AVI sequences use both audio and video streams. A simple variation for an AVI sequence uses video data and does not require an audio stream.

This section does not describe the OpenDML AVI file format extensions. For further information on these extensions, see *OpenDML AVI File Format Extensions*, published by the OpenDML AVI M-JPEG File Format Subcommittee.

## FOURCCs

A FOURCC (four-character code) is a 32-bit unsigned integer created by concatenating four ASCII characters. For example, the FOURCC 'abcd' is represented on a Little-Endian system as 0x64636261. FOURCCs can contain space characters, so ' abc' is a valid FOURCC. The AVI file format uses FOURCC codes to identify stream types, data chunks, index entries, and other information.

## RIFF File Format

The AVI file format is based on the RIFF (resource interchange file format) document format. A RIFF file consists of a RIFF header followed by zero or more *lists* and *chunks*.

- The RIFF header has the following form:

```
'RIFF' fileSize fileType (data)
```

where 'RIFF' is the literal FOURCC code 'RIFF', `fileSize` is a 4-byte value giving the size of the data in the file, and `fileType` is a FOURCC that identifies the specific file type. The value of `fileSize` includes the size of the `fileType` FOURCC plus the size of the data that follows, but does not include the size of the 'RIFF' FOURCC or the size of `fileSize`. The file data consists of chunks and lists, in any order.

- A chunk has the following form:

```
ckID ckSize ckData
```

where `ckID` is a FOURCC that identifies the data contained in the chunk, `ckSize` is a 4-byte value giving the size of the data in `ckData`, and `ckData` is zero or more bytes of data. The data is always padded to nearest **WORD** boundary. `ckSize` gives the size of the valid data in the chunk; it does not include the padding, the size of `ckID`, or the size of `ckSize`.



- A list has the following form:

```
'LIST' listSize listType listData
```

where 'LIST' is the literal FOURCC code 'LIST', `listSize` is a 4-byte value giving the size of the list, `listType` is a FOURCC code, and `listData` consists of chunks or lists, in any order. The value of `listSize` includes the size of `listType` plus the size of `listData`; it does not include the 'LIST' FOURCC or the size of `listSize`.

The remainder of this section uses the following notation to describe RIFF chunks:

```
ckID ( ckData )
```

where the chunk size is implicit. Using this notation, a list can be represented as:

```
'LIST' ( listType ( listData ) )
```

Optional elements are placed in brackets: [ optional element ]

## AVI RIFF Form

AVI files are identified by the FOURCC 'AVI' in the RIFF header. All AVI files include two mandatory LIST chunks, which define the format of the streams and the stream data, respectively. An AVI file might also include an index chunk, which gives the location of the data chunks within the file. An AVI file with these components has the following form:

C++

```
RIFF ( 'AVI '
      LIST ( 'hdr1' ... )
      LIST ( 'movi' ... )
      [ 'idx1' (<AVI Index>) ]
    )
```

The 'hdr1' list defines the format of the data and is the first required LIST chunk. The 'movi' list contains the data for the AVI sequence and is the second required LIST chunk. The 'idx1' list contains the index. AVI files must keep these three components in the proper sequence.

## Unit-IV

9.why is the access latency higher in optical storage?

1) In a storage latency, a synonym for *delay*, is an expression of how much time it takes for a packet of data to get from one designated point to another. In some usages (for example, AT&T), latency is measured by sending a packet that is returned to the sender and the round-trip time is considered the latency.

The latency assumption seems to be that data should be transmitted instantly between one point and another (that is, with no delay at all). The contributors to network latency include:

- **Propagation:** This is simply the time it takes for a packet to travel between one place and another at the speed of light.
- **Transmission:** The medium itself (whether [optical fiber](#), [wireless](#), or some other) introduces some delay. The size of the packet introduces delay in a round trip since a larger packet will take longer to receive and return than a short one.
- **Router and other processing:** Each [gateway node](#) takes time to examine and possibly change the header in a packet (for example, changing the [hop](#) count in the [time-to-live](#) field).
- **Other computer and storage delays:** Within networks at each end of the journey, a packet may be subject to storage and hard disk access delays at intermediate devices such as [switches](#) and [bridges](#). (In backbone statistics, however, this kind of latency is probably not considered.)

2) In a computer system, latency is often used to mean any delay or waiting that increases real or perceived [response time](#) beyond the response time desired. Specific contributors to computer latency include mismatches in data speed between the microprocessor and input/output devices and inadequate data [buffers](#).

Within a computer, latency can be removed or "hidden" by such techniques as *prefetching* (anticipating the need for data input requests) and *multithreading*, or using parallelism across multiple execution threads.

Q10. Discuss the pros and cons of the various CD-ROM, CD-I, and CD-R standards. Why would one not be appropriate for all applications?

## ***CD-ROM***

Stands for "Compact Disc Read-Only Memory." A CD-ROM is a CD that can be read by a computer with an optical drive. The "ROM" part of the term means the data on the disc is "read-only," or cannot be altered or erased. Because of this feature and their large capacity, CD-ROMs are a great media format for retail software. The first CD-ROMs could hold about 600 MB of data, but now they can hold up to 700 MB. CD-ROMs share the same technology as audio CDs, but they are formatted differently, allowing them to store many types of data.

**CD-I** (Compact Disc - interactive) is the multimedia CD format specified in 1986 (in the Green Book). CD-i was specified as an entire system, comprising not just a disc and data

format, but a hardware and software system, a variety of special [compression](#) methods for audio and visual data, and a method of interleaving audio, video, and text data.

Developed as a user-friendly alternative to a PC, CD-I players are easier to use, and have TV video output as well. Full screen motion video capabilities were added to the original specification later.

### **CD-R**

Stands for "Compact Disc Recordable." CD-R discs are blank CDs that can record data written by a CD burner. The word "recordable" is used because CD-Rs are often used to record audio, which can be played back by most CD players. However, many other kinds of data can also be written to a CD-R, so the discs are also referred to as "writable CDs."

The data burned onto a CD-R disc is permanent, meaning it can not be altered or erased like the data on a hard drive. Typically, once a CD has been burned, it will not be able to record any more data. Some CD burning programs can record data as "sessions," allowing a disc to be written to multiple times until it is full. Each session creates a new partition on the disc, meaning a computer will read a disc with multiple sessions as multiple discs. CD-RWs, on the other hand, can be erased and completely re-recorded. Like CDs, the data on CD-RWs cannot be altered, meaning the disc has to be completely erased each time you want to add new data.

## **UNIT-V**

Q.11. why must you consider multiple source for objects and multiuser operation for authoring systems?

Ans: An **authoring system** is a program which has pre-programmed elements for the development of interactive multimedia software titles. Authoring systems can be defined as software which allows its user to create multimedia applications for manipulating multimedia objects.

In the development of educational software, an authoring System is a program that allows a non-programmer to easily create software with programming features. The programming features are built in but hidden behind buttons and other tools, so the author does not need to know how to program. Generally Authoring Systems provide lots of graphics, interaction, and other tools educational software needs.

An authoring system usually includes an *authoring language*, a programming language built (or extended) with functionality for representing the tutoring system. The functionality offered by the authoring language may be programming functionality for use by programmers or domain representation functionality for use by subject experts.

There is overlap between authoring languages with domain representation functionality and domain-specific languages

Q.12 how does video conferencing relate to hypermedia messaging? What are the implications of building a system where the user starts with video conferencing and switches to integrated stored messaging?

Ans: **Hypermedia** is used as a logical extension of the term [hypertext](#) in which graphics, audio, video, plain text and [hyperlinks](#) intertwine to create a generally [non-linear medium](#) of information. This contrasts with the broader term [multimedia](#), which may be used to describe non-interactive linear presentations as well as hypermedia. It is also related to the field of [electronic literature](#). The term was first used in a 1965 article by [Ted Nelson](#).<sup>[1]</sup>

The [World Wide Web](#) is a classic example of hypermedia, whereas a non-interactive [cinema](#) presentation is an example of standard multimedia due to the absence of hyperlinks.

The first hypermedia work was, arguably, the [Aspen Movie Map](#). Atkinson's [HyperCard](#) popularized hypermedia writing, while a variety of literary hypertext and [hypertext](#) works, fiction and nonfiction, demonstrated the promise of links. Most modern hypermedia is delivered via [electronic pages](#) from a variety of systems including [media players](#), [web browsers](#), and stand-alone applications (i. e., software that does not require network access). Audio hypermedia is emerging with [voice command devices](#) and [voice browsing](#).

Conducting a [conference](#) between two or more participants at different sites by using [computer networks](#) to transmit audio and [video data](#). For example, a *point-to-point* (two-person) video conferencing [system](#) works much like a video telephone. Each participant has a video camera, microphone, and speakers mounted on his or her computer. As the two participants speak to one another, their voices are carried over the network and delivered to the other's speakers, and whatever images appear in front of the video camera appear in a [window](#) on the other participant's [monitor](#).

Multipoint videoconferencing allows three or more participants to sit in a [virtual](#) conference room and communicate as if they were sitting right next to each other. Until the mid 90s, the [hardware](#) costs made videoconferencing prohibitively expensive for most organizations, but that situation is changing rapidly. Many analysts believe that videoconferencing will be one of the fastest-growing segments of the computer industry in the latter half of the decade.