1. Introduction to Multimedia

- **Definitions and Types**:
  - What is multimedia? (A mixture of text, graphics, sounds, etc.) .
  - **Linear Multimedia**: Presentations that play from start to finish without user interaction.
  - **Non-Linear/Interactive Multimedia**: Allows users to select content to access.

2. Computer Graphics

- **Bitmapped (Raster) Graphics**:
  - Pixel composition and binary code for color.
  - **Categories**: Line art (1-bit), Grayscale (8-bit), and Color (24 bit).
  - **Image Quality**: Spatial resolution (ppi/dpi) and Color resolution (bit depth).
  - **Techniques**: Indexing (palettes) and Dithering.
- **Vector Drawn Graphics**:
  - Mathematically defined lines and shapes using Cartesian coordinates.
  - **Advantages**: Scalability without distortion and low file sizes.
  - **Device Independence**: Preserving original dimensions across different devices.
- **Color Reproduction**:
  - **Subtractive Process (CMYK)**: Used for printed surfaces.
  - **Additive Process (RGB)**: Used for computer monitors.
- **Image Quality and Resolution**

  - **Spatial Resolution**: The density of pixels per inch.
    - **ppi**: Pixels per inch (monitor output).
    - **dpi**: Dots per inch (print output).
  - **Color Resolution**: Determined by bit-depth.
    - **Quantization**: Breaks in the shades of continuous-tone images caused by low color resolution.
    - **Dithering**: Combining pixels of different colors to simulate a color not available in the palette.

- **File Formats and Compression**:
  - **Lossy vs. Lossless compression**.
  - Common formats: BMP, GIF, TIFF, JPEG, PNG, and SVG.

3. The HTML DOM (Document Object Model):

  - Finding elements using `getElementById()`.
  - Manipulating elements via scripting.

4. Drawing on the HTML5 Canvas

- **Canvas Basics**:
  - The `<canvas>` tag as a graphics container.
  - Accessing the **2D Context** via `getContext("2d")`.
- **Drawing Techniques**:
  - **Rectangles**: Using `fillStyle` and `fillRect(x, y, width, height)`.
  - **Lines and Paths**: Using `beginPath()`, `moveTo()`, `lineTo()`, and `stroke()`.
  - **Obtaining Crisp Lines**: Understanding pixel boundaries and path stroking.
- **Sprites and Images**:
  - Loading images with the `Image()` object.
  - Drawing images using `drawImage()` and its various parameters.
- **Transformations and State**:
  - **Translate**: Remapping the (0,0) position on the canvas.
  - **Save and Restore**: Managing the stack of properties that define the drawing state.

5. Product Overview and Design Specifications

- **Design and Planning**:
  - **Specification**: Hardware/software requirements, media elements, and functionality.
  - **Navigation**: Storyboards, sketches, and Spidergrams.
  - **Layout Principles**: Symmetrical vs. Asymmetrical balance, unity, and movement.
- **Development and Testing**:
  - **Usability Testing**: Feedback on program functionality and specifications.