COMPUTER EDUCATION

CLASS - X

Strictly as per KVS syllabus

तत् त्वं पूषन् अपावृणु

केन्द्रीय विद्यालय संगठन

Developed by - KVS REGIONAL OFFICE GUWAHATI

तत् त्वं पूषन् अपावृणु
केन्द्रीय विद्यालय संगठन

Patron
Shri C. Neelap
Deputy Commissioner, KVS RO Guwahati

Regional Co-ordination & Compilation
Shri D. Patle
Asstt. Commissioner, KVS RO Guwahati

School Co-ordination
Shri . K. Alung Khumba
Principal, KV NFR, Maligaon, Guwahati

Subject Expert
Dr. K. K. Motla, PGT(CS),
KV NFR, Maligaon, Guwahati

Sh. Rajendra Gavahle, PGT(CS),
KV NFR, Maligaon, Guwahati

Syllabus Content X

Quarter – April to June

S.no.	Units	Contents	No. of Periods
1.	**Review of Autodesk 3D Max**	1.1 What is 3D? 1.2 Navigation and Display 1.3 Creating Objects & Selecting Objects. 1.4 Parenting & Grouping objects. 1.5 Transforming Objects.	4
2.	Animation in 3D Max	**2.1.** Animation 2.1.1 Keys and Key framing. 2.1.2 Animation Controllers 2.1.3 Parametric vs. Transform Animation 2.1.4 Animation Playback 2.1.5 Acceleration 2.1.6 Time Configuration 2.1.7 Cycling &Linking **2.2** Editing tools: 2.2.1 Select Modifiers 2.2.2 Edit Modifiers 2.2.3 Editable Object Commands 2.2.4 Sub-Object Selection 2.2.5 Modifying & Transforming Sub-Objects 2.2.6 Mesh Editing Levels **2.2.7** Spline Editing Levels	12

Quarter 2- July to Sept

3.	**Customizing & Embedding Multimedia components in Webpages**	3.1 **Inserting Multimedia files in Webpages** **3.1.1 Compatible Multimedia Files formats for Webpages.** **3.1.2 Embedding Audio File.** **3.1.3 Embedding Video File.** **3.1.4 Embedding Flash File.**	**16**

Quarter 3- Oct to Dec

4.	**Web Scripting Using Java Script**	4.1 **REVIEW OF JAVA SCRIPT OF CLASS IX** **4.1.1 Variables & Operators** **4.1.2 If & Switch** **4.1.3 Iteration (Loops)** **4.1.4 Window Object** **4.1.5 Pop – Up Boxes –Alert, Confirm Etc.** **4.2 FUNCTIONS –USER DEFINED** **4.2.1 Function Definition** **4.2.2 Calling a Function** **4.2.3 Function Parameters** **4.2.4 Return Statement** **4.3 String Object** **4.3.1 Syntax** **4.3.2 String Properties** **4.3.3 String Methods** **4.3.4 String HTML Wrappers** **4.4 Math Object** **4.4.1 Syntax** **4.4.2 Array Properties** **4.4.3 Array Methods** 4.5 **Event** **4.5.1 What is an Event?** **4.5.2 An click Event Type** **4.5.3 On Submit Event Type** **4.5.4 On Mouseover & On Mouseout** **4.5.5 HTML 4 Standard Events**	**20**

Quarter 4- Jan to Mar

5.	**Operating Web Based Application**	**5.1 E-GOVERANCE** **5.1.1 What is E-Governance** **5.1.2 Major E- Governance Projects in India** **5.1.3 Societal Imports of E- Governance** **5.2. E-BUSSINESS** **5.2.1 What is E- Business** **5.2.2 Major E- Business Portals** **5.2.3 Societal Impacts of E- Business** **5.3 E- LEARNING** **5.3.1 What is E- Learning?** **5.3.2 Major E- Learning Sites** **5.3.3 Societal Impacts of E- Learning.**	**6**

Project	**Developing a Project on coding a website using HTML, JavaScript & CSS.**

Lesson-1

Review of Autodesk 3D Max

You will learn

- ✓ What is 3D?
- ✓ Navigation and Display
- ✓ Creating Objects & Selecting Objects.
- ✓ Parenting & Grouping objects.
- ✓ Transforming Objects.

What is 3D?

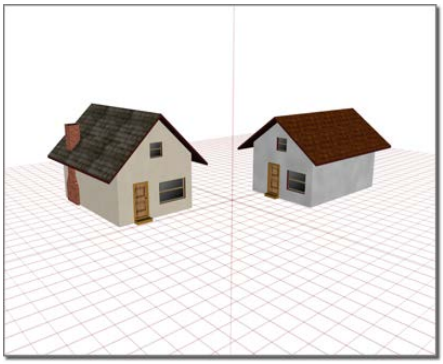
3D means three-dimensional, i.e. something, that has width, height and depth (length). Our physical environment is three-dimensional and we move around in 3D every day. Humans are able to perceive the spatial relationship between objects just by looking at them because we have 3D perception, also known as depth perception. As we look around, the retina in each eye forms a two-dimensional image of our surroundings and our brain processes these two images into a 3D visual experience.

Navigation and Display

- Coordinate Systems & The Home Grid

Grids are two-dimensional arrays of lines similar to graph paper, except that you can adjust the spacing and other features of the grid to the needs of your work.

Using the home grid to position houses



Grids have these primary uses:

- As an aid in visualizing space, scale, and distance
- As a construction plane where you create and align objects in your scene
- As a reference system for using snap

The home grid is the basic reference system, defined by three fixed planes on the world coordinate axes . The home grid is visible by default when you start 3ds Max, but can be turned off with an option in the right-click viewport menu. You can use any view of the home

instead.

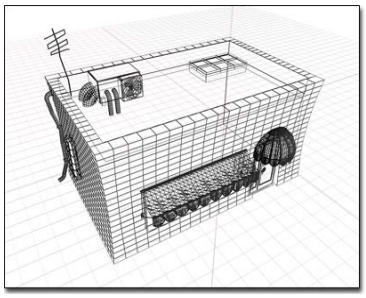
- Orthogonal vs. Perspective Views

“Orthogonal” is a term used to describe two vectors that are perpendicular (at 90 degrees) to each other. In 3D space, when the X, Y, or Z-Axes are not perpendicular, they are considered “non-orthogonal” and the FBX plug-in does not support their representation as a matrix.

Because the FBX plug-in assumes that there is always a 90-degree angle between the X, Y, and Z axes, it can support only orthogonal matrices. Any transformed axes that have non-orthogonal TRS matrices are ignored by the FBX plug-in, so it does not import or export effects created when axes are not orthographic.

Perspective Views: most closely resemble human vision. Objects appear to recede into the distance, creating a sense of depth and space. For most 3D computer graphics, this is the view used in the final output that the client sees onscreen or on the page. Perspective view of the ice-cream shop

There are three ways to create a perspective view in a viewport perspective view, camera view, and light view.

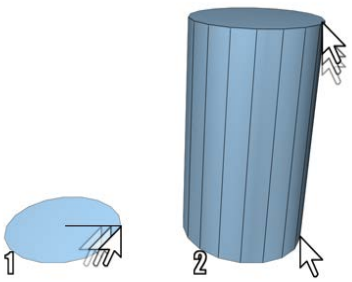


A perspective viewport, labeled Perspective, is one of the default viewports. You can change any active viewport to this eye-like point of view by pressing the keyboard shortcut P.

A camera view requires that you first create a camera object in your scene. The camera viewport tracks the view through the perspective of that camera. As you move the camera (or target) in another viewport, you see the scene swing accordingly. If you alter the camera's field of view on the Modify command panel, you see the changes as they are applied.

Creating Objects & Selecting Objects.

With some variations, the steps shown in the following images apply to creating any type of object on the Create panel. For specific examples, see the Procedures section in any object's topic.



1. Radius defined 2. Height defined

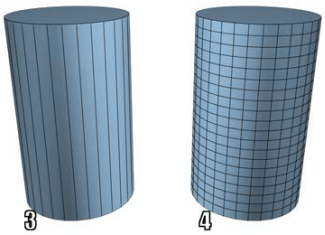
3. Sides increased 4. Height Segments increased

To choose an object category:

1. Click (Create tab) to view the Create panel.
2. Click one of the buttons at the top of the Create panel. For example, (Geometry).
3. Choose the subcategory Standard Primitives from the list.

A number of buttons appear on the Object Type rollout.

To choose an object type:



- Click the button for the type of object you want to create.
  - ✓ The button highlights, showing that it is active. Four rollouts appear Name and Color, Creation Method, Keyboard Entry, and Parameters.
  - ✓ To choose a creation method (optional):
  - ✓ You can accept the default method and skip this step.
- Choose a method in the Creation Method rollout.
  - ✓ To preset the creation parameters (optional):
  - ✓ You can adjust all creation parameters after you create an object.

- In the Parameters rollout, you can set parameters before you create an object. However, the values of parameters you set by dragging the mouse (for example, the Radius and Height of a cylinder) have no effect until after you create the object.
-

To create the object:

1. Put the cursor at a point in any viewport where you want to place the object, and hold the mouse button down (do not release the button).
2. Drag the mouse to define the first parameter of the object; for example, the circular base of a cylinder.
3. Release the mouse button. The first parameter is set with this release. In some cases, such as Sphere, Teapot, and Plane, this completes the object. You can skip the remaining steps.
4. Move up or down without touching the mouse button. This sets the next parameter; for example, the height of a cylinder.

   If you want to cancel: Until you complete the next step, you can cancel the creation process with a right-click.
5. Click when the second parameter has the value you want, and so on.

The number of times you press or release the mouse button depends on how many spatial dimensions are required to define the object. (For some kinds of objects, such as Line and Bones, the number is open-ended.)

Parenting & Grouping objects.

1. Main toolbar ➤  (Select And Link)
2. Use the Select and Link button to define the hierarchical relationship between two objects by linking them as child and parent.
3. You link from the currently selected object (child) to any other object (parent).

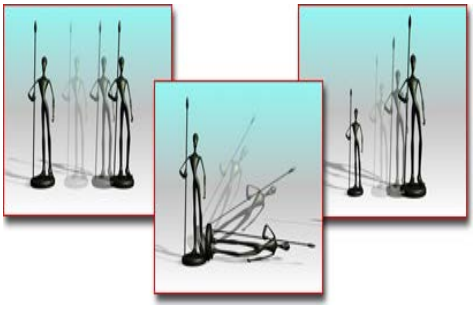
group parent rather than any member of the group. The entire group flashes to show that you've linked to the group.

5. A child inherits the transformations (move, rotate, scale) applied to the parent, but the child's transformations have no effect on the parent. If you want the child not to inherit the transforms,

use the Link Inheritance (Selected) Utility or use the controls found in Link Info in the Hierarchy panel.

6. You can also create hierarchical linkages using Schematic View. Use the Connect button on the Schematic View toolbar to create hierarchical linkages between nodes.

Transforming Objects:



When you create any object, 3ds Max records its position, rotation, and scale information in an internal table called a transformation matrix. Subsequent position, rotation, and scale adjustments are called transforms.

Moving, rotating, and scaling a figure. An object's actual position within the world coordinate system is always calculated in relation to its internal, or local coordinate system, which is based on the object's transformation matrix. The origin of the local coordinate system is the center of the object's bounding box.

An object can carry any number of modifiers, but only one set of transforms. Although you can change transform values from frame to frame, each object always has only one position, one rotation, and one scale transform.

You can animate your transforms by turning on the Auto Key button and then performing the transform at any frame other than frame 0. This creates a key for that transform at the current frame.

SUMMARY

1. 3D means three-dimensional, i.e. something, that has width, height and depth (length).
2. **“Orthogonal”** is a term used to describe two vectors that are perpendicular (at 90 degrees) to each other. In 3D space, when the X, Y, or Z-Axes are not perpendicular, they are considered “non-orthogonal”.
3. There are three ways to create a perspective view in a viewport perspective view, camera view, and light view.
4. When you create any object, 3ds Max records its position, rotation, and scale information in an internal table called a transformation matrix.
5. Subsequent position, rotation, and scale adjustments are called transforms.

EXERCISES

A. Fill in the blanks:

i. 3D means_____dimension.
ii. 3D perception is also called _____perception.
iii. Subsequent position, _____and_____ are referred as transforms.
iv. In 3D space when X,Y or Z Axes are not _____they considered “non - orthogonal”.

i. Grid are three dimensional arrays of links similar to graph paper.
ii. The home grid is visible by default when you start 3Ds max.
iii. An object can carry any number of modifiers but only one set of transforms.
iv. A camera view requires that you first create light view in your scene.

C. Short Answer Questions:

i. What is Grid in Autodesk?
ii. What is the difference between orthogonal and non-orthogonal views?
iii. What are the ways to create perspective view in a viewport?
iv. Write the steps to create object.?
v. What is transform?

In the lab

1. Teachers are requested to give revision of class IX practical on 3D Max.

Lesson-2

Animation in 3D Max

Animation

Keys and Key framing.

Animation is based on a principle of human vision. If you view a series of related still images in quick succession, your brain perceives them as continuous motion. Each image is called a frame.

Historically, the major difficulty in creating animations has been that the animator must produce a large number of frames. Depending on the quality you want, one minute of animation might require between 720 and 1800 separate still images. Creating images by hand is a big job. That is where key framing comes in.

Animation Controllers:

Most of the frames in an animation are routine, incremental changes from the previous frame directed toward some predefined goal. Early animation studios quickly realized they could increase the productivity of their master

You will learn

Animation

- ✓ Keys and Key framing.
- ✓ Animation Controllers
- ✓ Parametric vs. Transform Animation
- ✓ Animation Playback
- ✓ Acceleration
- ✓ Time Configuration
- ✓ Cycling &Linking

Editing tools:

- ✓ Select Modifiers
- ✓ Edit Modifiers
- ✓ Editable Object Commands
- ✓ Sub-Object Selection
- ✓ Modifying & Transforming Sub-Objects
- ✓ Mesh Editing Levels
- ✓ Spline Editing Levels

called keyframes.

Assistants could then figure out the frames that were required in between the key frames. These frames were (and still are) called tweens. Use 3ds Max as your animation assistant. As the master animator, you create the key frames that record the beginning and end of each transformation. The values at

these key frames are called keys. 3ds Max calculates the interpolated values between each key value, resulting in tweened animation.

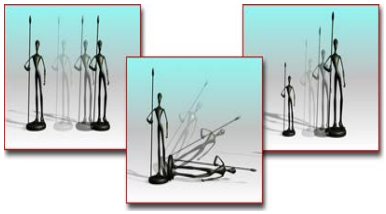
Parametric vs. Transform Animation

Parametric animation is an animation technique used in computer software, such as in computer games that blends two or more separate animations together to form a new animation. This new animation is constructed in real-time by the game engine, and is not stored in a separate file like a regular animation. The technique was first used in an early build of the Half-Life mod, Team Fortress 2, and it not only heavily reduces artist workload during game development, it provides for much smoother animation as well.

How parametric animation works?

It combines different layers of animation automatically. The whole process controls by program, determines which animation layer should be used and controls the method of combination like override, additive, overly or and the amount of density.

A **transform** is an adjustment of an object's position, orientation, or scale, relative to the 3D world (or world space) in which you are working.Changing a model by changing its position, rotation, or scale



You can apply three basic types of transform to an object:

- Position
-  Rotation
-  Scale

  And a fourth transform command lets you position an object automatically on an underlying surface:

-  Placement

This section presents brief topics to help you quickly start learning how to transform objects and how to animate your transforms.

Failure to Move or Rotate

In some cases, an object might fail to move or rotate, even when the proper command is active and the object is selected. This could be due to one of the following reasons:

- The object is frozen.
- A transform controller has been assigned to the object.
- Inverse Kinematics mode is on and the preference called Always Transform Children of the World is off.

Procedures : To transform an object using the main toolbar:

1. On the main toolbar, click one of the three transform buttons:  (Select And Move),  (Select And Rotate), or  (Select And Uniform Scale). These buttons are usually referred to as Move, Rotate, and Scale.

   Alternatively, to position an object on another object's surface, click  (Select And Place).

2. Position the mouse over the object you want to transform.
   a. If the object is already selected, the cursor changes to indicate the transform.
   b. If the object is not selected, the cursor changes to a crosshairs to show that the object can be selected.

If you start the drag over an unselected object, it becomes selected and is also transformed.You can restrict transforms to one or two axes easily with the Using Transform Gizmos.

Animation Playback

Play/Stop

 The Play button plays the animation in the active viewport. If you click another viewport to make it active, the animation continues playing in that viewport. When the animation is playing, the Play button becomes a Stop button. The Play button is a flyout for playing only the animation of selected objects.

- Status bar > Time controls >  (Play Animation)
- Status bar > Time controls >  (Stop Animation)
- Keyboard > / (to play) (3ds Max mode)
- Keyboard > .)  +V (Maya mode: See Interaction Mode Preferences
- Keyboard >  (to stop)

Procedures: To play the animation in the viewport:

1. Activate the viewport where you want to play the animation.
2. Click  (Play Animation).

3. Click  (Stop Animation) to end the playback.

The speed of the animation playback is determined by the settings in the Time Configuration dialog, the complexity of the scene and the speed of the graphics card and processor.

To play the animation looped backward:

1. In the animation controls, click  (Time Configuration).

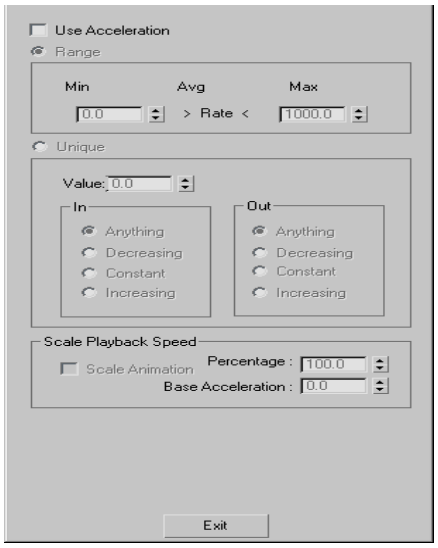
2. In the Time Configuration dialog ➤ Playback group, turn off Real Time.
    i. The Direction buttons are now available.
3. Turn on Reverse and click OK.
4. Click  (Play Animation).
    i. The animation plays backward.
5. To play the animation front-to-back and then back-to-front in a continuous loop, turn on Ping-Pong as the Direction.

To play the animation of a selected object only:

1. In a viewport,  select a single animated object or a set of animated objects.
2. Choose  (Play Selected) from the Play/Stop flyout.
    i. Only the selection is animated in the viewport.
3. To end playback, click  (Stop Animation) or press  .

Acceleration

unit is the current 3ds Max system unit. That is, the acceleration is determined by the rate at which the speed changes per frame.**Use Acceleration**Turn on to have the motion synthesis engine consider delegate acceleration in determining whether to activate the state.



When you choose Range, the motion synthesis engine activates the clip when the delegate's acceleration falls inside the specified range.

Min/Avg/Max display

After you synthesize the Master Motion Clips, displays delegates' minimum, average, and maximum acceleration.

Min

Set a minimum acceleration value for the range.

Max

Set a maximum acceleration value for the range.

When you choose Unique, the motion synthesis engine activates the clip when the delegate's acceleration matches a specific value, optionally with a rising, falling, or constant value before or after the specified value.

Value

Set a unique acceleration value.

In and Out groups

These radio buttons let you specify the behaviour of the parameter before and after the unique value is met.

- **Anything** Acceleration before or after the target value is not relevant.
- **Decreasing** Acceleration decreases before or after it reaches the target value.
- **Constant** Acceleration before or after the target value is constant.
- **Increasing** Acceleration increases before or after it reaches the target value.

acceleration of the delegate.

Scale Animation

Scale the clip's animation based on acceleration.

For example, as a bird accelerates, its wings beat more rapidly. Scaling an animation scales the keys of the animation.

Percentage

Specify how much to alter the playback speed based upon the difference between the delegate's acceleration and the Base Acceleration setting.

The formula used is this:

Animation Speed change % = (current speed/Base Acceleration – 1) x Percentage %

For example, if a delegate is accelerating 50 percent faster than the base acceleration, and the Scale Percentage value is 50, then the playback speed is scaled up by 25 percent.

Base Acceleration

Specifies the delegate acceleration at which the animation should be played back at its normal rate.

Time Configuration

The Time Configuration dialog provides settings for frame rate, time display, playback, and animation. You use this dialog to change the length of your animation, or stretch or rescale it

- Status bar > Time controls >  (Time Configuration) > Time Configuration dialog

Procedures: To define the active time segment:

1. Click  (Time Configuration).
2. In the Time Configuration dialog ➢ Animation group, set Start Time to specify the beginning of your active time segment.

- Set End Time to specify the end of your active time segment.
- Set Length to specify the amount of time in the active time segment and automatically set the correct End Time.

You can enter positive or negative values in any spinner, but you must use the same format used by the time display.

To stretch out your existing animation over a longer time:

1. In the Time Configuration dialog ➤ Animation group, click Re-scale Time.
2. Change the value in Length to be the number of frames you want the action to fill.
3. Click OK.

The animation is rescaled to the new number of frames.

To add frames onto your existing animation:

This procedure adds new frames to the end of your animation, without affecting your existing work.

1. In the Time Configuration dialog ➤ Animation group ➤ End Time field, enter the number of the last frame of the animation.
   For example, if your existing animation is 100 frames long and you want to add 50 frames, enter **150**.

2. Click OK.
   The number you entered is now the new length of the animation, shown on the time slider.

To move to an exact time in your animation:

- In the Time Configuration dialog ➤ Animation group, enter the frame number in the Current Time field, and press Enter.

In the Time Configuration dialog ➤ Frame Rate group, do one of the following:

1. Choose one of the standard frame rates such as PAL or NTSC.
2. Choose Custom, and specify a frame rate in the FPS (frames-per-second) field.

To play your animation in reverse or back and forth:

1. In the Time Configuration ➤ Playback group, turn off the Real Time.
2. Choose the direction of the animation playback by selecting Forward, Reverse, or Ping- Pong.
3. Play the animation in the viewport using  (Play Animation) or the ***/*** key.

To play your animation only once:

1. In the Time Configuration ➤ Playback group, turn off Loop.
2. Choose the direction of the animation playback by selecting Forward, Reverse or Ping-Pong.

3. Play the animation in the viewport using  (Play Animation) or the ***/*** key.

The animation will play once and stop.

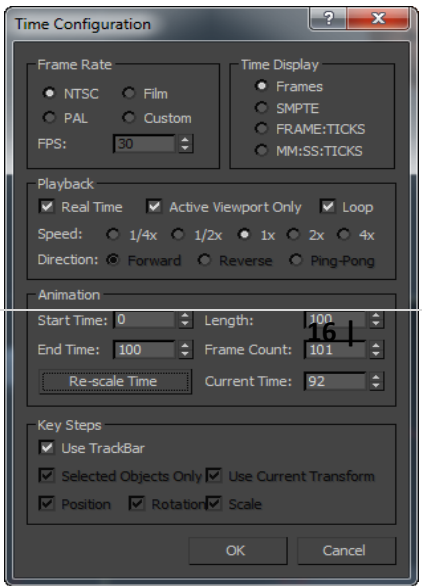
To play your animation in multiple viewports:

1. In the Time Configuration ➤ Playback group, turn off Active Viewport Only. Click OK.
2.  Play your animation.

To play an animation with sound:

- In the Time Configuration ➤ Playback group, be sure you have Real Time turned on. If Real Time is not on, the sound will not play back during the animation.

Interface



These are the controls for the Time Configuration dialog. You can display this dialog by right-clicking any of the time control buttons to the right of the Auto Key button.

These four option buttons, labeled NTSC, Film, PAL, and Custom let you set the frame rate in frames-per-second (FPS).

FPS (Frames Per Second)

Sets the frame rate of your animation in Frames per Second. Use frame rates of 30 fps for video, 24 for film, and lower rates for web and media animations.

Cycling & Linking

Linking Objects

The general process of creating links is to build the hierarchy from child to parent. You click Select and Link on the toolbar, select one or more objects as children, and then drag the link cursor from the selection to a single parent object. The selected objects become children of the parent object.

Once objects are linked, any transformations applied to the parent are also applied to its children. For example, if you scale the parent to 150%, the size of its children and the distance between the children and the parent are also scaled by 150%.

Unlinking Objects

Click Unlink Selection to remove the link from selected objects to their parents. Any children of the selected object are unaffected.

You can quickly unlink an entire hierarchy by double-clicking the root object to select the object and all of its children. Then click Unlink Selection..

Linking Animated Objects

You should establish links before you begin animating objects. The linkage of objects with Select and Link cannot be animated; the link remains in force throughout the entire animation.

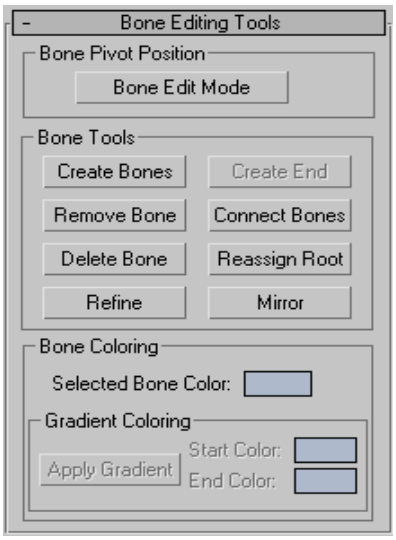
If you want your objects to be linked during one part of the animation but not another, you can a Link constraint to change the linkage at specific frames.

Editing tools:

Select Modifiers

Controls on the Bone Editing Tools rollout let you create and modify bone geometry and structure, and set bone color for one or more bones

Interface
Bone Pivot Position group



Bone Edit Mode

Lets you change the lengths of bones and their positions relative to one another.

When this button is on, you can change the length of a bone by moving its child bone. In effect, you can scale or stretch a bone by moving its child bone while in this mode. You can use this tool both before and after assigning an IK chain to the bone structure.

When Bone Edit Mode is on, you cannot animate, and when Auto Key or Set Key is on, Bone Edit Mode is unavailable. Turn off Auto/Set Key to edit bones.

Bone Tools group

Create Bones Begins the bone-creation process. Clicking this button is the same as clicking Create panel ➤ Systems ➤ Bones System.

Create End Creates a nub bone at the end of the currently selected bone. If the selected bone is not at the end of a chain, the nub is linked in sequence between the currently selected bone and the next bone in the chain.

Remove Bone Removes the currently selected bone. The bone's parent bone is stretched to reach the removed bone's pivot point, and any children of the removed bone are linked to its parent. Any IK chains that included the removed bone will remain intact.

When you click this button, a dotted line appears in the active viewport from the first selected bone. Move the cursor to another bone to create a new connecting bone. The first selected bone will become a parent to the connecting bone, which is in turn a parent to the second selected bone.

Delete Bone Deletes the currently selected bone, removing all its parent/child associations. A nub is placed at the end of the deleted bone's parent. Any IK chains that included this bone become invalid.

Reassign Root Makes the currently selected bone the root (parent) of the bone structure.

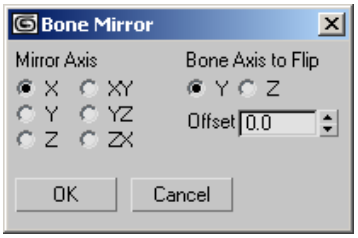
Refine Splits a bone in two. Click Refine, and then click a bone where you want it to split.

Mirror Opens the Bone Mirror dialog (see following), which lets you create mirror copies of selected bones without changing the sign of the bones' scale. Instead, Mirror flips one of the bone axes: Y or Z. You can specify the mirroring axis and the flip axis with the dialog controls.

Bone Mirror dialog

Opens when you click the Mirror button. Use it to specify the mirroring axis, the flip axis, and an offset value.

While the dialog is open, you can see a preview of the mirrored bone(s) in the viewports. Click OK to create the bones, or Cancel to prevent creation.



Mirror Axis Choose an axis or plane about which the bones will be mirrored: X/Y/Z or XY/YZ/ZX.

Bone Axis to Flip To avoid creating a negative scale, choose the bone axis to flip: Y or Z.

Offset The distance between the original bones and the mirrored bones. Use this to move the mirrored bones to the other side of the character.

Edit Modifiers

To be demonstrated by teacher in practical class

Editable Object Commands

To be demonstrated by teacher in practical class

Sub-Object Selection

manipulate the sub-object components of the modifier. When a modifier's parameters are being edited, it has the option of providing the system with different levels of sub-object selection. There are two types of sub-object selection:

1. Selection of a modifier's Gizmo or Center - A modifier can have a visual representation in the scene which users can manipulate called a gizmo. The Bend modifier uses a 3D box. When Gizmo is selected in the sub-object selection list of Max's UI, the user can transform the modifier's gizmo itself -- that is, they can adjust its location, orientation, and scale in relation to the object it is affecting. When Center is selected in the list, they can transform the pivot point of the modifier's gizmo. For example, if you want a bend to occur about the base of the Bend gizmo, you could move the center to the base of the gizmo.

2. Selection of components in the pipeline - The second type of sub-object selection is selection of components of the object flowing through the pipeline. If the object in the pipeline is a triangle object, then sub-object selection level might be things like vertex, face, edge, etc. These are options for the Edit Mesh modifier, for instance.

Modifying & Transforming Sub-Objects

Selecting Sub-Objects

- To select a vertex, edge, or face/polygon/element, click it.
- To add to the sub-object selection, hold down Ctrl and click, or drag to specify a region.
- To subtract from the sub-object selection, hold down Alt and click, or drag to specify a region.

Note: When you drag to specify a region with an existing sub-object selection, if any transform tool is active, you will transform the selection instead of changing the selection. To avoid this, start the region away from the object, or first activate  (Select Object) on the main toolbar.

Following is a general procedure in setting up an object for sub-object selection.

editable poly, and so on.

2. On the modifier stack display, click ⊞ (the plus-sign icon to the left of the name of the modifier or editable object).

3. On the stack display, choose the kind of sub-object geometry you want to work with: for example, Vertex, Face, or Edge. Each sub-object selection level has rollouts with their own sets of options.

   The sub-object level is highlighted in the list.

4. Use standard selection techniques (see preceding) to select sub-object geometry, from a single sub-object to the entire object. By default, the sub-object selection highlights in red.

Storing Sub-Object Selection Sets

The surface formats (mesh, poly, and so on), automatically remember the most recent selection for each sub-object level: vertex, edge, and so on (there is overlap in some cases, such as poly edge and border). These selection sets are saved with the file. With sub-object selections, you have these options:

- Choose one of the selection sets to pass geometry up the stack to other modifiers. Only one selection set is active at a time.
- Change to one of the other selection sets at any time by activating its sub-object level.
- Use named selection sets for sub-object selections you want to reuse.
-

Using Sub-Object Selections

- Apply any options supplied for the kind of object and the selection level.
- Apply standard transforms: Move, Rotate, Scale. For more information, see Transforming a Sub-Object Selection, following.
- Apply object-space modifiers (Bend, Taper, and Twist, for example) to perform useful modelling operations.

surfacing operations.

- Bind a space warp to the selection. The rest of the object is unaffected by the warping.
- Use the toolbar commands Align, Normal Align, and Align To View with face selections.

Transforming a Sub-Object Selection

Using an editable mesh, poly, patch, or spline, you can directly transform any sub-object selection. However, “Select” modifiers like Mesh Select and Spline Select enable only selection.

To transform a sub-object selection made with a Select modifier:

1. Add an XForm modifier to the stack, following (or somewhere above) the Select modifier.
2. In the stack, open the Select modifier and make a sub-object selection.
3. Choose XForm in the stack. You then transform the XForm gizmo, which applies the transform to the selection.

Mesh Editing Levels

1. Select an editable mesh object. ➤  Modify panel ➤ Selection rollout ➤ Choose any sub-object level.

2. Select an editable mesh object. ➤  Modify panel ➤ Modifier stack display ➤ ⊞ Expand the Editable Mesh entry. ➤ Choose any sub-object level.
3. Select an editable mesh object. ➤ Quad menu ➤ Tools 1 quadrant ➤ Choose any sub-object level.

   The Edit Geometry rollout for Meshes contains most of the controls that let you alter the geometry of the mesh, at either the Object (top) level, or one of the sub-object levels. The controls that the rollout displays can vary, depending on which level is active; if a

appear at all.

Spline Editing Levels

1. Create or select a spline ➤  Modify panel ➤ Right-click spline entry in the stack display ➤ Convert To: Editable Spline
2. Create a line ➤ Modify panel
3. Create or select a spline ➤ Right-click the spline ➤ Transform (lower-right) quadrant of the quad menu ➤ Convert To: ➤ Convert to Editable Spline
   Editable Spline provides controls for manipulating an object as a spline object and at three sub-object levels: **vertex, segment, and spline.**

   The functions in Editable Spline are the same as those in the **Edit Spline modifier**. The exception is that when you convert an existing spline shape to an editable spline, the creation parameters are no longer accessible or animatable. However, the spline's interpolation settings (step settings) remain available in the editable spline.

SUMMARY

1. Animation is based on a principle of human vision. Each image is called a frame.
2. Parametric animation is an animation technique used in computer software, such as in computer games that blends two or more separate animations together to form a new animation.

world (or world space) in which you are working.

4. Acceleration is measured in units per frame per frame, where the unit is the current 3ds Max system unit. That is, the acceleration is determined by the rate at which the speed changes per frame.
5. The general process of creating links is to build the hierarchy from child to parent.
6. **Offset is** the distance between the original bones and the mirrored bones. Use this to move the mirrored bones to the other side of the character.

EXERCISES

A. Fill in the blanks:

i. Each image is called_____.
ii. Parametric animation is an animation technique used in_____.
iii. A _____is an adjustment of an object position.
iv. The _____button plays the animation in the active viewport.
v. Acceleration is measured in_____ per frame.
vi. Click _____selection to remove the link from selected objects to their parents.
vii. _____sets the frame rate of your animation in frame per second.

B. State true or false:

i. When you drag to specify a region with an existing sub-object selection , if any transform tool is active, you will transform the selection instead of changing the selection.

ii. If the object in the pipeline is a triangle object, then sub object selection level might be things like vertex, face, edge etc.

iii. To avoid creating a negative scale, choose the bone axis to flip X or Y.

iv. Refine splits a bone in two.

vi. The general process of creating links is to build the hierarchy from child to parent.

vii. Use frame rates of 30 fps for video, 26 for film and highest rate for web and media animation.

C. Short Answer Questions:

i. What is animation?
ii. What is keyframes?
iii. Write one difference between parametric and transform animation.?
iv. How parametric animation work?
v. Write name of three basic transform which we can apply on object.?
vi. Write the steps to play animation in viewport.?
vii. What is Acceleration?
viii. Write the steps to add frames in existing animation.?
ix. What do you mean by linking object?
x. Write one difference between linking object and un-linking object.

In the lab

1. Create an animation on moving object and play it.
2. Create an animation on “Save Tree” and display it.(Teachers are requested to guide the student in proper sequence).

Lesson-3

Customizing & Embedding Multimedia components in

You will learn

Inserting Multimedia files in Webpages

- ✓ Compatible Multimedia Files formats
- ✓ for Webpages.
- ✓ Embedding Audio File.
- ✓ Embedding Video File.
- ✓ Embedding Flash File.

Inserting Multimedia files in Webpages

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see. Examples: Pictures, music, sound, videos, records, films, animations, and more.
Web pages often contain multimedia elements of different types and formats.

Browser Support

The first web browsers had support for text only, limited to a single font in a single color. Later came browsers with support for colors and fonts, and even support for pictures!
The support for sounds, animations, and videos is handled differently by various browsers. Different types and formats are supported , and some formats require extra helper programs (plug-ins) to work.

Compatible Multimedia Files formats for Webpages.

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4)
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.

Sound Formats

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Commonly used in music players. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. MP4 video is the upcoming video format on the internet. This leads to automatic support for MP4 audio by all browsers.

Embedding Audio File.

Method 1

1. Open your Web page in an HTML editor or Notepad
2. Start with an object element:
   <object>
3. You'll add 4 parameters to the object. The first is "src" that tells the browser where to find the sound file. In this example, the sound file is eureka.wav and is found in the same directory as the Web page:
   <param name="src" value="eureka.wav" />
4. If you want the sound file to play immediately after it's loaded, make the autostart parameter "true" otherwise make it "false":
   <param name="autostart" value="true" />
5. The parameter autoplay is similar to autostart, just used by other browsers, set it the same as the autostart parameter:

   <param name="autoplay" value="true"/>
6. Use the controller parameter to tell the browser if a controller should be displayed to give your readers more control over the sound:

   <param name="controller" value="true" />
7. Inside the <object></object> element, add an embed element:
   <embed />
8. Add the following four attributes that are the same as the parameters to the object:

   <embed src="eureka.wav" controller="true" autoplay="true" autostart="True" />
9. Add the correct MIME type for your sound file into the type attribute:

type="audio/wav" />

10. Add the plugins page attribute so that people who don't have the correct plugin for your sound file can go download it. For WAV files, recommended QuickTime:

<embed src="eureka.wav" controller="true" autoplay="true" autostart="True"
type="audio/wav" pluginspage="http://www.apple.com/quicktime/download/" />

11. When you're done, your HTML should look like this:

<object>
<param name="autostart" value="true">
<param name="src" value="eureka.wav">
<param name="autoplay" value="true">
<param name="controller" value="true">
<embed src="eureka.wav" controller="true" autoplay="true" autostart="True"
type="audio/wav" />
</object>

Method 2

Make a hyperlink directly to the audio file using the following code (change "audiofilename.wmv" to your own file name):

**Click here to listen audio **

A link is created like this: Click here to listen audio. When the end user clicks this hyperlink, their Windows Media Player will open and load the audio file for playing.

Embedding Video File.

1. First Upload Your Video to Your Web Server
2. Create your video as a .mov or .mp4 file - and then upload it to your Web server
3. Write down the full URL to your video file - you'll need it when you write your HTML.
4. Use the Object Tag to Embed Your Video Validly

The object tag is not widely supported, but it is the only way to add a QuickTime video into your pages using valid HTML.

There are two attributes that you need to include to get the video to play correctly:

Class id="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
codebase=http://www.apple.com/qtactivex/qtplugin.cab

5. You'll also want to set the width and height of the object - set the width to the width of the movie and the height to the height plus 15-20 pixels extra to include the controls.

6. Your object tag will look like:

   <object width="960" height="555" classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" codebase="http://www.apple.com/qtactivex/qtplugin.cab">

7. Parameters Make the Object Work

8. The only parameter that you need is the src parameter - this tells the browser where to find the movie to play. Set the src attribute to the URL of your movie:

   <param name="src" value="http://URL to movie.mov" />

9. Some other parameters that can be useful to control your video and make it more user-friendly are:

   - controller - this can be true or false. Remember that most people get very frustrated if they cannot control multimedia elements on Web pages that they visit. So I strongly recommend setting this to true.

   - autoplay - this can be true or false. Autoplay also annoys customers. They want to decide when and whether to watch the video. So I strongl recommend setting this to false.

10. Inside the <object> tag, place the following parameters:

    <param name="controller" value="true" /> <param name="autoplay" value="false" />

11. Finally, Include an Embed Tag with the Same Information

    If it is very important to you that all your pages validate, then you should not include this section. But remember that only people with some browsers like Safari and Opera will be able to view your video.

The embed tag should look like this. Note that all the fields are the same as the :

<embed src="http://URL to movie.mov" width="960" height="555" autoplay="true" controller="true" pluginspage="http://www.apple.com/quicktime/download/"> </embed>

12. And then close the object tag:

</object>

Embedding Flash File.

Method 1: Using Dreamweaver to insert SWF file to webpage

Dreamweaver is a visual HTML editor with built-in objects that allow you to easily insert Flash content. The required HTML tags are automatically inserted by Dream weaver.

1. Copy the SWF file to the same folder as the HTML document.
2. In Dreamweaver, select the location of the page where you wish to add the Flash content (including inside a div, table cell or frame).
3. Click the Insert Flash button in Dreamweaver's Object palette, or choose Insert> Media > Flash. Browse to and choose the SWF file.
4. Upload both the SWF and HTML files to your Web server, placing them in the same directory.

Method 2: Manually adding the proper tags to an existing web page

If you need to incorporate the flash slideshows created by Photo Slideshow Maker into an existing HTML or other type of webpage document, a better way is manually adding the proper tags to the existing page.

Web page must contain tags that reference the actual Flash movie file to be opened and played. These tags are the <OBJECT> and <EMBED> tags.

The OBJECT tag is used by Internet Explorer on Windows and the EMBED is used by Netscape Navigator (Macintosh and Windows) and Internet Explorer (Macintosh) to direct the browser to load the Macromedia Flash Player.

1. Open the PFM-created HTML document in a text or HTML editor

2. Copy the HTML code included in the <object> and <embed> tags (see example below) and paste it into the existing HTML document. This should contain the <OBJECT> and <EMBED> tags

codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=6,0,0,0" width="777" height="528" id="tech" align="middle">
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="showcase/flash/fashionshow.swf" />
<param name="quality" value="high" />
<embed src="showcase/flash/fashionshow.swf" quality="high" width="777"
height="528" name="tech" align="middle" allowScriptAccess="sameDomain"

type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>

In this case, showcase/flash/ is the folder directory, while the fashionshow.swf is the SWF file name.

3. Open the existing webpage page, and paste the <object> and <embed> tags into the desired location in the body of the document. This can also be pasted into a table cell or frame.

4. Edit the attributes of the tags for the movie.

Change the height and width parameters to match the height and width of the movie dimensions or use percentage values, if desired.

5. Upload the HTML and SWF files to the same folder on the Web server.

SUMMARY

different formats.

2. Pictures, music, sound, videos, records, films, animations are example of multimedia.
3. The first web browsers had support for text only, limited to a single font in a single color.
4. Plug-ins are the extra helper program of web browser.
5. MPEG, AVI, and WMV are example of common video format.
6. MIDI, RealAudio, and WMA are example of audio format.

EXERCISES

A. Fill in the blanks:

i. Multimedia comprises text, sound, _____, image and video.
ii. MPEG developed by_____.
iii. _____tells the browser where to find the sound file.
iv. The parameter auto play is similar to _____ just used by other web browser.
v. _____ is visual HTML editor with built in objects that allow you to easily insert Flash content.

B. State true or false:

i. The first web browser had support for flash only.
ii. AVI developed by Microsoft.
iii. To play audio file we place the source in href attribute.
iv. Plug – ins are extra format to play audio in web page.
v. WMA is example of audio format.

C. Short Answer Questions:

i. What is multimedia program?
ii. Name four common video format which is compatible with web pages.?
iii. Write the steps to play an audio file from web page.?
iv. Write the steps to embed video file in web page.?
v. How to import Plug – ins in web page write their steps?

In the lab

1. Create a web page which has information on Mr. Amitabh Bacchan with one link to the audio of him. When user click on the given link, it should play Mr.Bacchan voice.
2. Create a web page on “Save Tiger” which has some information on tiger with one link to the video. When user click on the given link, it should play video on Save Tiger.
3. Create a web page which has link to the download the missing Plug –ins for the web page.

Lesson-4

Web Scripting using JavaScript

You will learn

REVIEW OF JAVA SCRIPT OF CLASS IX
- ✓ Variables & Operators
- ✓ If & Switch
- ✓ Iteration (Loops)
- ✓ Window Object
- ✓ Pop – Up Boxes –Alert, Confirm Etc.

FUNCTIONS –USER DEFINED
- ✓ Function Definition
- ✓ Calling a Function
- ✓ Function Parameters
- ✓ Return Statement

String Object
- ✓ Syntax
- ✓ String Properties
- ✓ String Methods
- ✓ String HTML Wrappers

Math Object
- ✓ Syntax

Array Properties
Array Methods
Event
What is an Event?
An click Event Type

REVIEW OF JAVA SCRIPT OF CLASS IX

Variables & Operators

JavaScript Data Types: One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types:

- Numbers eg. 123, 120.50 etc.
- Strings of text e.g. "This text string" etc.

Java does not make a distinction between integer values and floating-point values.

JavaScript Variables:

Variable is the stored memory location that can hold a value in it. Variables are declared with the **var** keyword. Storing a value in a variable is called variable initialization.

Example:

```
<script type="text/javascript">
<!--
var name = "Ali";
var money;
money = 2000.50;
//-->
</script>
```

JavaScript Variable Scope:

The scope of a variable is the region of your program in which it is defined. JavaScript variable will have only two scopes.

- **Global Variables:** A global variable has global scope which means it is defined everywhere in your JavaScript code.
- **Local Variables:** A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

```
<script type="text/javascript">
<!--
var myVar = "global"; // Declare a global variable
function checkscope( ) {
  var myVar = "local";  // Declare a local variable
  document.write(myVar);
}
```

```
</script>
```

This produces the following result:

```
Local
```

JavaScript Variable Names:

While naming your variables in JavaScript keep following rules in mind.

- You should not use any of the JavaScript reserved keyword as variable name.
- JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or the underscore character.
- JavaScript variable names are case sensitive.

JavaScript Reserved Words:

The following are reserved words in JavaScript. They cannot be used as JavaScript variables, functions, methods, loop labels, or any object names.

boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Types of Operators: logical, arithmetic, relational

Operators: An Operator is a symbol that performs an operation**.** JavaScript language supports following type of operators.

- Arithmetic Operators
- Comparision Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators

The Arithmetic Operators:

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example

-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide numerator by denumerator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

The Logical Operators:

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non zero then then condition becomes true.	(A && B) is true.
||	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false.

The Comparison Operators:

Assume variable A holds 10 and variable B holds 20 then:

==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B will assign value of A + B into C
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	C %= A is equivalent to C = C % A

Decision Making using if & Switch

if...else Statements: While writing a program, you need to make use of conditional statements that allow your program to make correct decisions and perform right actions.

JavaScript supports following forms of **if..else** statement:

- if statement
- if...else statement
- if...else if... statement.

if statement:

The **if** statement allows JavaScript to make decisions and execute statements conditionally.

Syntax:

```
if (expression){
   Statement(s) to be executed if expression is true
}
```

Example:

```
<script type="text/javascript">
var age = 20;
if( age > 18 ){
   document.write("<b>Qualifies for driving</b>");
}
</script>
```

This will produce following result:

```
Qualifies for driving
```

if...else statement:

The **if...else** statement is the next form of control statement that allows JavaScript to execute statements in more controlled way i.e to choose from given options.

Syntax:

```
if (expression){
   Statement(s) to be executed if expression is true
}else{
   Statement(s) to be executed if expression is false
}
```

Example:

```
<script type="text/javascript">
var age = 15;
if( age > 18 ){
   document.write("<b>Qualifies for driving</b>");
}else{
   document.write("<b>Does not qualify for driving</b>");
}
</script>
```

This will produce following result:

Does not qualify for driving

if...else if... statement: The **if...else if...** statement allows JavaScript to make correct decision out of several conditions.

Syntax:

```
if (expression 1){
   Statement(s) to be executed if expression 1 is true
}else if (expression 2){
   Statement(s) to be executed if expression 2 is true
}else if (expression 3){
   Statement(s) to be executed if expression 3 is true
}else{
   Statement(s) to be executed if no expression is true
}
```

Example:

```
<script type="text/javascript">
var book = "maths";
if( book == "history" ){
   document.write("<b>History Book</b>");
}else if( book == "maths" ){
   document.write("<b>Maths Book</b>");
}else if( book == "economics" ){
   document.write("<b>Economics Book</b>");
}else{
  document.write("<b>Unknown Book</b>");
}
```

This will produce following result:

Maths Book

Switch Case: You can use multiple *if...else if* statements, to perform a multiway branch. However, this is not always the best solution, especially when all of the branches depend on the value of a single variable.

Syntax: The basic syntax of the switch statement is to give an expression to evaluate and several different statements to execute based on the value of the expression. The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.

The break statements indicate to the interpreter the end of that particular case. If they were omitted, the interpreter would continue executing each statement in each of the following cases.

```
switch (expression)
{
  case condition 1: statement(s)
                    break;
  case condition 2: statement(s)
                    break;
   ...
  case condition n: statement(s)
                    break;
  default: statement(s)
}
```

Example: Following example illustrates a basic Switch case:

```
<script type="text/javascript">
var grade='A';
document.write("Entering switch block<br />");
```

```
{
  case 'A': document.write("Good job<br />");
            break;


case 'B': document.write("Pretty good<br />");
            break;
  case 'C': document.write("Passed<br />");
            break;
  case 'D': document.write("Not so good<br />");
            break;
  case 'F': document.write("Failed<br />");
            break;
  default:  document.write("Unknown grade<br />")
}
document.write("Exiting switch block");
</script>
```

This will produce following result:

```
Entering switch block
Good job
Exiting switch block
```

Example: Consider a case if you do not use break statement:

```
<script type="text/javascript">
var grade='A';
document.write("Entering switch block<br />");
switch (grade)
{
  case 'A': document.write("Good job<br />");
  case 'B': document.write("Pretty good<br />");
  case 'C': document.write("Passed<br />");
  case 'D': document.write("Not so good<br />");
  case 'F': document.write("Failed<br />");
  default:  document.write("Unknown grade<br />")
}
document.write("Exiting switch block");
</script>
```

This will produce following result:

```
Entering switch block
Good job
Pretty good
Passed
Not so good
Failed
Unknown grade
Exiting switch block
```

Iteration – Loops

A loop is a block of code that allows you to repeat a section of code a certain number of times; perhaps changing certain variable values each time the code is executed.

This not only saves you the time and trouble of repeatedly typing the same lines of code, but also avoids typing errors in the repeated lines.

JavaScript allows you to use the *for*, *while*, and *do while* loops.

❖ **For loop: Structure of a for Loop**

```
for (initialization; test condition; iteration statement)
{
  JavaScript Code Here
}
```

This line determines how many times the loop will run

The JavaScript code for the loop will be inside the brackets here

Parts of for loop are:

1. **Initialization:** Here we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
2. **Test condition:** If condition is true then code given inside the loop will be executed otherwise loop will come out.

counter.

Example: JavaScript to write a sentence to the page 5 times

```
<script type="text/javascript">
<!--
for (var count=1;count<6;count+=1) {
   document.write("I am part of a loop!<br />");
}
//-->
</script>
```

This will produce following result:

```
I am part of a loop!
I am part of a loop!
I am part of a loop!
I am part of a loop!
I am part of a loop!
```

- **While loop:**

syntax : initialization;

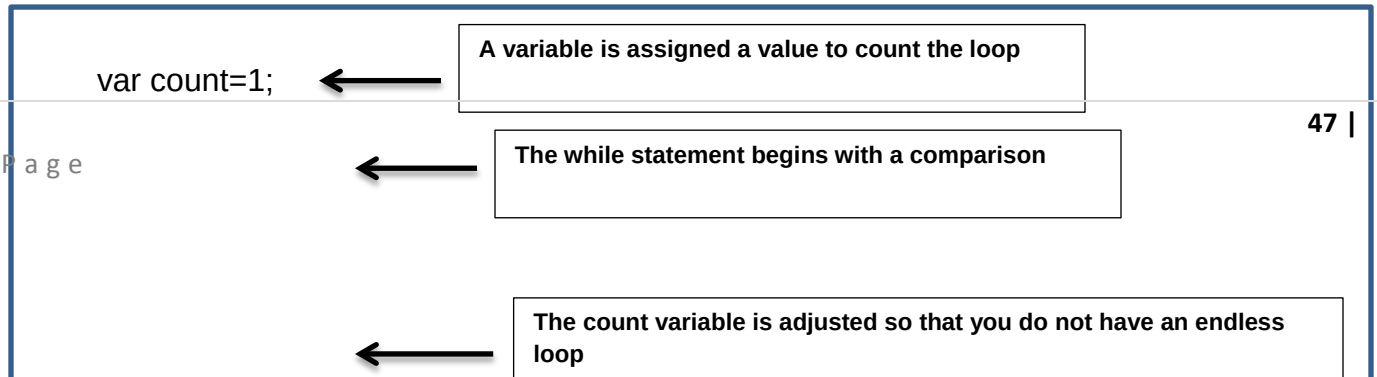
while(condition)

{ JavaScript Code Here;

Update expression;

}

Example:



```
{
        JavaScript Code Here
        count++;
```

5 a **Program to print numbers using basic while loop:**

```
<script type="text/javascript">
var count = 1;
document.write("Starting Loop" + "<br />");
while (count <= 5)
{
  document.write("Number : " + count + "<br />");
  count++;
}
document.write("Loop stopped!");
</script>
```

This will produce following result:

```
Starting Loop
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
Loop stopped!
```

Window Object

The window object represents an open window in a browser.

If a document contain frames (<iframe> tags), the browser creates one window object for the HTML document, and one additional window object for each frame.

Some common Window Object Methods:

alert() Displays an alert box with a message and an OK button

blur() Removes focus from the current window

close() Closes the current window

confirm() Displays a dialog box with a message and an OK and a Cancel button

focus() Sets focus to the current window

getSelection() Returns a Selection object representing the range of text selected by the user

moveTo() Moves a window to the specified position

open() Opens a new browser window

print() Prints the content of the current window

prompt() Displays a dialog box that prompts the visitor for input

resizeBy() Resizes the window by the specified pixels

scrollBy() Scrolls the document by the specified number of pixels

stop() Stops the window from loading

Popup Boxes – alert, confirm etc.

JavaScript supports three important types of dialog boxes. These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the users.

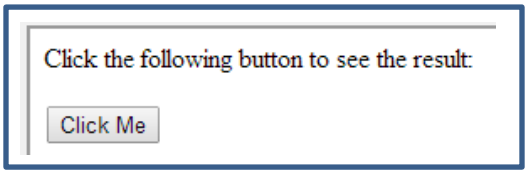
- **Alert Dialog Box:**

An alert dialog box is mostly used to give a warning message to the users.

```
<html>
<head>
<script type="text/javascript">
<!--
function Warn()
```

```
   alert("This is a warning message!");
}
//-->
</script>
</head>
<body>
<p>Click the following button to see the result: </p>
<form>
<input type="button" value="Click Me" onclick="Warn();" />
</form>
</body>
</html>
```

Nonetheless, an alert box can still be used for friendlier messages. Alert box gives only one button "OK" to select and proceed.



On clicking 'Click Me' button, an alert window appears.

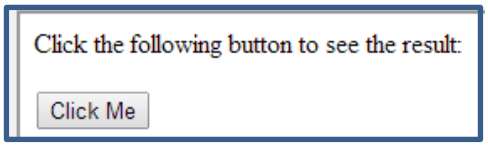
❖ Confirmation Dialog Box:

A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: OK and Cancel.

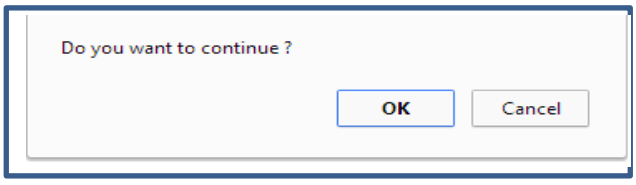
If the user clicks on OK button the window method confirm() will return true. If the user clicks on the Cancel button confirm() returns false. You can use confirmation dialog box as follows:

```
<head>
<script type="text/javascript">
<!--
function getConfirmation(){
   var retVal = confirm("Do you want to continue ?");
   if( retVal == true ){
      alert("User wants to continue!");
               return true;
   }else{
      alert("User does not want to continue!");
               return false;
   }
}
//-->
</script>

</head>
<body>
<p>Click the following button to see the result: </p>
<form>
<input type="button" value="Click Me" onclick="getConfirmation();" />
</form>
</body>
</html>
```

Output will appear on the screen.



On clicking button 'Click Me' a confirmation, dialog box appears.



❖ Prompt Dialog Box

enable you to interact with the user. The user needs to fill in the field and then click OK.

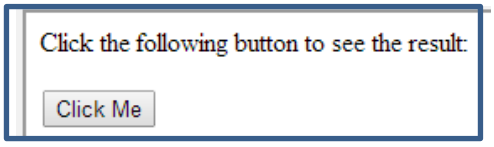
This dialog box is displayed using a method called prompt() which takes two parameters (i) A label which you want to display in the text box (ii) A default string to display in the text box.

This dialog box with two buttons: OK and Cancel. If the user clicks on OK button the window method prompt() will return entered value from the text box. If the user clicks on the Cancel button the window method prompt() returns null.

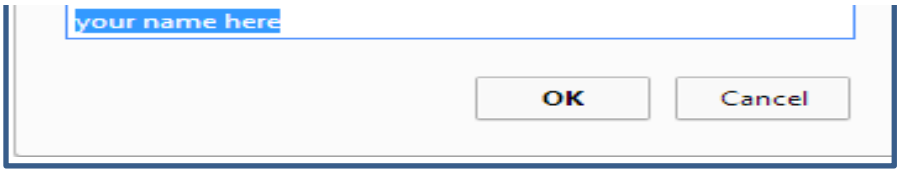
You can use prompt dialog box as follows:

```
<html>
<head>
<script type="text/javascript">

function getValue(){



 var retVal = prompt("Enter your name : ", "your name here");
   alert("You have entered : " +  retVal );
}
</script>
</head>


<body>
<p>Click the following button to see the result: </p>
<form>
<input type="button" value="Click Me" onclick="getValue();" />
</form>
</body>
</html>
```

Output will appear on the screen.



On clicking button ‘Click Me’ prompt dialog box will appear on the screen.

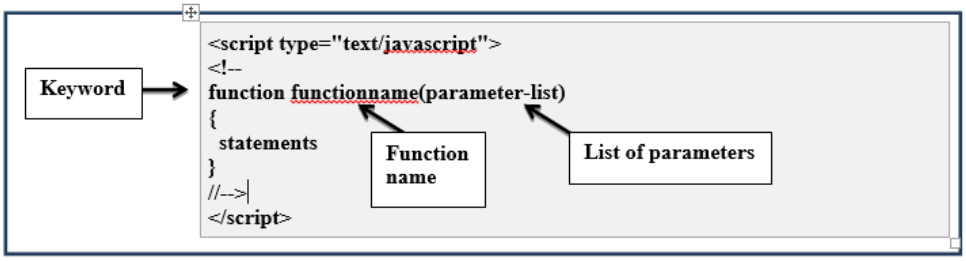


FUNCTIONS –USER DEFINED

A function is a group of reusable code which can be called anywhere in your programme. This eliminates the need of writing same code again and again. This will help programmers to write modular code. You can divide your big programme in a number of small and manageable functions.

Function Definition:

The most common way to define a function in JavaScript is by using the **function keyword**, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces. The **basic syntax** is shown here:



Example:

A simple function that takes no parameters called sayHello is defined here:

```
<!--
function sayHello()
{
   alert("Hello there");
}
//-->
</script>
```

Calling a Function:

To invoke a function , you would simple need to write the name of that function as follows:

```
<script type="text/javascript">

<!--

sayHello();

//-->

</script>
```

Function Parameters:

We can pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters.

A function can take multiple parameters separated by comma.

Example: Let us do a bit modification in our *sayHello* function. This time it will take two parameters:

```
<script type="text/javascript">
<!--
function sayHello(name, age)
{
   alert( name + " is " + age + " years old.");
}
//-->
</script>
```

mind in adding numbers into strings.Now we can call this function as follows:

```
<script type="text/javascript">
<!--
sayHello('Zara', 7 );
//-->
</script>
```

The return Statement

A JavaScript function can have an **optional return statement**. This is required if you want to return a value from a function. This statement should be the last statement in a function.

For example you can pass two numbers in a function and then you can expect from the function to return their multiplication in your calling program.

Example:

This function takes two parameters and concatenates them and return resultant in the calling program:

```
<script type="text/javascript">
<!--
function concatenate(first, last)
{


 var full;

   full = first + last;
   return  full;
}
//-->
</script>
```

Now we can call this function as follows:

```
<script type="text/javascript">
<!--
   var result;
```

```
   alert(result );
//-->
</script>
```

String Object

JavaScript is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers:

- **Encapsulation .** the capability to store related information, whether data or methods, together in an object
- **Aggregation .** the capability to store one object inside of another object
- **Inheritance .** the capability of a class to rely upon another class (or number of classes) for use of its properties and methods
- **Polymorphism .** the capability to write one function or method that works in a variety of different ways

The **String** object let's you work with a series of characters and wraps Javascript's string primitive data type with a number of helper methods.

Syntax : var val = new String(string)

Example:

```
var  guitar_string = new String("Gagan");
```

OR

var val = string;

Example:

```
var  s1= “Gagan”;
```

String Properties

Here is a list of each property and their description.

Property	Description
length	Returns the length of the string.

EXAMPLE:

```
<body>
   <script type="text/javascript">
      var myname="John";
      document.write("The name has "+myname.length+" characters.");
  </script>
</body>
```

Result is :

The name has 4 characters.

String Methods

Here is a list of each method and its description.

Method	Description
charAt()	Returns the character at the specified index.
concat()	Combines the text of two strings and returns a new string.
substr()	Returns the characters in a string beginning at the

	characters.
substring()	Returns the characters in a string between two indexes into the string.
toLowerCase()	Returns the calling string value converted to lower case.
toString()	Returns a string representing the specified object.
toUpperCase()	Returns the calling string value converted to uppercase.

Example:

```
var str1= new String("Gagan");
var str2 = new String("Sagan");
 document .write(concat(str1,str2));
```

Result is : **GaganSagan**

String HTML Wrappers

Here is a list of each method, which returns a copy of the string wrapped inside the appropriate HTML tag.

Method	Description
big()	Creates a string to be displayed in a big font as if it were in a <big> tag.

bold()	Creates a string to be displayed as bold as if it were in a <b> tag.
fontcolor()	Causes a string to be displayed in the specified color as if it were in a tag.
fontsize()	Causes a string to be displayed in the specified font size as if it were in a tag.
italics()	Causes a string to be italic, as if it were in an <i> tag.
link()	Creates an HTML hypertext link that requests another URL.
small()	Causes a string to be displayed in a small font, as if it were in a <small> tag.
strike()	Causes a string to be displayed as struck-out text, as if it were in a <strike> tag.

Example:

var **text** = "I am so mad I am red!";
document.write(**text.fontcolor("red")**);

This script places the following code into the page source:
I am so mad I am red!

Result is: **I am so mad I am red!**

Math Object

The **math** object provides you properties and methods for mathematical constants and functions.

Syntax

var variablename = mathfunction;

Example:

var sq = Math.SQRT2; // sq = 1.414

Math Properties

Here is a list of each property and their description.

Property	Description
PI	Ratio of the circumference of a circle to its diameter, approximately 3.14159.
SQRT2	Square root of 2, approximately 1.414.

Math Methods

Here is a list of each method and its description.

Method	Description
abs()	Returns the absolute value of a number.
max()	Returns the largest of zero or more numbers.
min()	Returns the smallest of zero or more numbers.
pow()	Returns base to the exponent power, that is, base exponent.
random()	Returns a pseudo-random number between 0 and 1.
round()	Returns the value of a number rounded to the nearest integer.
sqrt()	Returns the square root of a number.

Example :

```
var num =4;
document .write(“ SQURE ROOT OF NUM = “+ sqrt(num) );
```

Result is : SQURE ROOT OF NUM = 2

Event

What is an Event?

An occurance of an activity is called an event.

When the page loads, that is an event. When the user clicks a button, that click, too, is an event. Another example of events are like pressing any key, closing window, resizing window etc.

On click Event Type

This is the most frequently used event type which occurs when a user clicks mouse left button. You can put your validation, warning etc against this event type.

Example:

```
<html>
<head>
<script type="text/javascript">
<!--
function sayHello() {
   alert("Hello World")
}
//-->
</script>
</head>
<body>
<input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

This will produce following result and when you click Hello button then *onclick* event will occur which will trigger *sayHello()* function.

On Submit Event Type

you can put your form validation against this event type.

Example:

```
<html>
<head>
<script type="text/javascript">
<!--
function validation() {
   all validation goes here
   .........
   return either true or false
}
//-->
</script>
</head>
<body>
<form method="POST" action="t.cgi" onsubmit="return validate()">
.......
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

On Mouseover & On Mouseout

The *onmouseover* event occurs when you bring your mouse over any element and the*onmouseout* occurs when you take your mouse out from that element.

Example: Following example shows how a division reacts when we bring our mouse in that division:

```
<html>
```

```
<script type="text/javascript">
<!--
function over() {
   alert("Mouse Over");
}
function out() {
   alert("Mouse Out");
}
//-->
</script>
</head>
<body>
<div onmouseover="over()" onmouseout="out()">
<h2> This is inside the division </h2>
</div>
</body>
</html>
```

HTML 4 Standard Events

The standard HTML 4 events are listed here for your reference. Here *script* indicates a Javascript function to be executed agains that event.

Event	Value	Description
onchange	script	Script runs when the element changes
onsubmit	script	Script runs when the form is submitted
onreset	script	Script runs when the form is reset
onclick	script	Script runs when a mouse click
onmouseout	script	Script runs when mouse pointer moves out of an element
onmouseover	script	Script runs when mouse pointer moves

SUMMARY

1. One of the most fundamental characteristics of a programming language is the set of data types it supports.
2. Variable is the stored memory location that can hold a value in it. Variables are declared with the **var** keyword.
3. The scope of a variable is the region of your program in which it is defined. JavaScript variable have only two scopes: Global and Local Variables
4. **Global Variables** is a global variable has global scope which means it is defined everywhere in your JavaScript code.
5. **Local Variables** is a local variable will be visible only within a function where it is defined.
6. An Operator is a symbol that performs an operation**.** JavaScript language supports Arithmetic, Comparison, Logical, Assignment, and Conditional Operators.
7. The **if** statement allows JavaScript to make decisions and execute statements conditionally.
8. The **if...else** statement is the next form of control statement that allows JavaScript to execute statements in more controlled way i.e to choose from given options.
9. **if...else if... statement:** The **if...else if...** statement allows JavaScript to make correct decision out of several conditions.
10. Switch case is one of the alternate of multiple if. Else…if…
11. A loop is a block of code that allows you to repeat a section of code a certain number of times; perhaps changing certain variable values each time the code is executed.
12. The window object represents an open window in a browser
13. A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: OK and Cancel.
14. The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus it enable you to interact with the user. The user needs to fill in the field and then click OK.

eliminates the need of writing same code again and again.

16. The **math** object provides you properties and methods for mathematical constants and functions
17. An occurance of an activity is called an event.
18. The *onmouseover* event occurs when you bring your mouse over any element and the*onmouseout* occurs when you take your mouse out from that element.

EXERCISES

A. Fill in the blanks:

i. _____ is the stored memory location that can hold value in it.
ii. _____ cannot be used as variable.
iii. An _____ is a symbol that performs an operation.
iv. _____control statement allows JavaScript to execute statement in more controlled way.
v. _____is one of the alternate of multiple if….else….if.
vi. The _____ object represents an open window in a browser.
vii. _____ is mostly used to give warning message to the users.
viii. _____ is a group of reusable code which can be called anywhere in your program.
ix. _____keyword is used to define function in JavaScript.
x. The _____object provides you properties and methods for mathematical constants and functions.

B. State true or false:

i. The onmouseover event occurs when you take out your mouse out from that element.
ii. An occurrence of an activity is called event.
iii. Pow( ) method returns base to the exponent power.
iv. Encapsulation is the capability of a class to rely upon another class for use of its properties and methods.
v. JavaScript function can have a optional return statement.

C. Short Answer Questions:

i. What is data type in JavaScript?
ii. What data type support JavaScript?
iii. What is variable? What are the types of variable?
iv. Write one difference between Local variable and Global variable.?
v. What do you mean by Reserved word?
vi. What is operator? What operator support JavaScript?
vii. Give your reason why the following operators are used:
    a) Arithmetic Operator
    b) Logical Operator
    c) Comparison Operator
viii. Write syntax for the following:
    a) If else if statement.
    b) Switch case
    c) For loop
ix. What is window object? Write some common window object methods.
x. Write one difference between Confirmation Dialog Box and Prompt Dialog Box.
xi. What is function? Give one example of it.
xii. What is the use of return statement in function?
xiii. Write short notes on:
    a) Encapsulation
    b) Aggregation
    c) Inheritance
    d) Polymorphism
xiv. What is string object in JavaScript?
xv. What is event? Write about Onclick event type with example.

In the lab

1. Write a program in JavaScript which will display all the even numbers between 1 to 20.
2. Write a program in JavaScript which will display your name 10 times and say good Bye at the end.
3. Write a program in JavaScript which will call a function where it is displaying the largest number between the two number.
4. Write a program in JavaScript which will display different warning message in the same program.
5. Write a JavaScript which will display All the month name with reference of their month number(Using Switch Case).

NOTE: Teachers are requested to give them a group project on JavaScript.

Lesson-5

Operating Web based application

You will learn

E-GOVERANCE
- ✓ What is E-Governance
- ✓ Major E- Governance Projects in India
- ✓ Societal Imports of E-Governance

E-BUSSINESS
- ✓ What is E- Business
- ✓ Major E- Business Portals
- ✓ Societal Impacts of E-Business

E- LEARNING
- ✓ What is E- Learning?
- ✓ Major E- Learning Sites
- ✓ Societal Impacts of E-Learning.

E-GOVERANCE

What is E-Governance?

E-GOVERNANCE refers to the application of electronic means in governance with an aim of fulfilling the requirements of common man at affordable costs and in fastest possible time.

Major E- Governance Projects in India

1. MCA 21 ,India's prestigious G2B services Portal (URL: www.mca.gov.in)
2. Consular passport and VISA Division (Indian Passport Office)(URL: http://passport.gov.in)
3. Income Tax Portal (URL: http://www.incometaxindia.gov.in)
4. National Portal of India (URL: http://www.india.gov.in)
5. DRDO(URL : http://www.drdo.nic.in)
6. Supreme Court of India (URL : http://supremecourtofindia.nic.in)
7. Indian Courts (URL: http://www.indiancourts.nic.in)
8. RTI Portal (URL:http://www.rti.gov.in)

Societal Impacts of E- Governance

Positive Impacts:

1. Improve efficiency of administration and service delivery.
2. Reduce waiting time.

3. Reduce cost of availing the services.
4. Keep a tab on corruption to some extent.
5. Increased public participation.
6. Increase transparency & increased accountability of government offices

Negative Impacts:

1. People living in rural & remote areas face lack of computerization.
2. Not all services are part of E-governance so manual methods are used.
3. Lack of awareness prevented people to benefit.
4. Incompatibility of software & hardware.
5. Websites are slow & needs improvement.

E-BUSSINESS

What is E- Business?

E-BUSINESS refers to any form of transaction that uses an electronic medium to facilitate the transaction.

Major E- Business Portals

1. IRCTC portal (URL: http://www.irctc.co.in)
2. Online Reservation site Yatra .com(URL: http://www.yatra.com)
3. Life Insurance Cooperation of India (URL: http://www.licindia.com)
4. State Bank of India (URL: http://www.statebankofindia.com)
5. Online store Amazon.com (URL:http://www.amazon.com)

Societal Impacts of E- Business

Positive Impacts

1. Increase in internet users
2. Middle class attracted towards low cost flights

4. Increase in online payments

Negative Impacts

1. Poor telecom & infrastructure for reliable connectivity
2. Multiple issues of trust
3. Multiple gaps in the current legal & regulatory frame work

E- LEARNING

What is E- Learning?

E-LEARNING is a flexible term used to describe a means of teaching through technology such as a network, browser, CDROM or DVD multimedia platforms.

Major E- Learning Sites

1. Moodle portal (http://www.moodle.org)
2. W3Schools (http://www.w3schools.com)
3. eXe Project (http://www.exelearning.org)
4. Xerte Project (http://www.nottingham.ac.uk/xerte)

Societal Impacts of E- Learning.

Positive Impacts

1. Availability of same course to millions.
2. Boon for Working class.
3. Apprehensive Employers.

Negative Impacts

1. People living in rural & remote areas face lack of computerization.
2. Lack of awareness prevented people to benefit.
3. Cultural differences obstruct the true aim of E-Learning.
4. High dropout rate.

SUMMARY

1. E-GOVERNANCE refers to the application of electronic means in governance with an aim of fulfilling the requirements of common man at affordable costs and in fastest possible time.
2. E-BUSINESS refers to any form of transaction that uses an electronic medium to facilitate the transaction.
3. E-LEARNING is a flexible term used to describe a means of teaching through technology such as a network, browser, CDROM or DVD multimedia platforms.

EXERCISES

A. Short Answer Questions:

i. What is E-Governance? Write few names of major E-Governance project in India.
ii. Write two positive and two negative societal impacts of E-Governance.
iii. What is E-Business? Write few names of E-Business Portals.
iv. Write two positive and two negative societal impacts of E-Business.
v. What is E-Learning? Write few names of E-Learning web sites.
vi. Write two positive and two negative societal impact of E-Learning.

Check Your Progress

Short Answer Questions

Q.1 What are Functions?

Q.2 What is the need of function?

Q.3 Define: a) Function Call b) Function definition

Q.4 What are parameters in functions?

Q.5 What is the significance of return statement ?

Multiple Choice Questions

Q.1 Which Keyword should be used to write functions?
A) var B) String C) function D) return

Q.2 Where should you write the Return statement in a function?
A) Beginning B) Ending C) Middle D) nowhere

Q.3 What are the disadvantages of functions?
A) Reduced time B) Reduced Efforts C) repetition of code D) complexity

Q.4 How many values are returned by the function?
A) one B) two C) Zero D) NULL

Q.5 How can we separate the list of function parameters in the function definition?
A) by commas B) by colon C) by semi colon D) by opeartor

True or False

Q.1 The return statement can return two values from a function at a time.

Q.2 A function can take multiple parameters separated by comma.

Q.3 A function is a group of reusable code which can be called anywhere in your programme.

Q.5 Function definition can be written in body tag or in head tag of HTML.

Find the Output

Q.1 What value will be stored in '**result**'?

```
function change(n1)
{
var result;
result = n1 + 10;
return  result;
}
```

Q.2 What value will be stored in '**a**'?

```
function change(b)
{
var a;
a=b/3;
return  a;
}
```

Q.3 **Consider the following code fragment and Find the Output**

```
function find_data()

{

var ch = ‘b’;

switch(ch)
{
case ‘a’ : document.write(“It is a” + “<BR>" ) ;
break;
case ‘b’ : document.write((“It is b” + “<BR>") ;

default : document.write((“It is not valid” + “<BR>" ) ;
break;
}
```

```
function find_data()

{

var sum = 1;

for( var x=8; x>=1 ; x = x-2)

{

sum=sum * x ;

}

document.write ( " result is "+ sum);

}
```

Practical Exercises with solution:

Q.1 Write the function using Javascript to find the sum of two numbers entered by the user.

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">

   var number1 = parseFloat( prompt("Enter the first number") );
   var number2 = parseFloat( prompt("Enter the second number") );

   function add(n1,n2)
    {
       var result;
       result = n1 + n2;
       return  result;
     }
```

```
      document.write("result is"+r);

</script>
</body>
</html
```

Q.2 Write the function using Javascript to find the largest of two numbers.

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">

function large(n1,n2)
 {
     var result;
     if (n1 > n2)
       result = n1;
     else
       result = n2;
     return  result;
 }

       var r = large(10,20) ;
       document.write("result is"+r);

</script>
</body>
</html>
```

Q.3 Write the function using Javascript to find the remainder and quotient of a number, if divided by another number.

Hint: use of math.floor( ) gives you integer quotient

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">

   var n1 = parseInt( prompt("Enter the divisor of a number") );
   var n2 = parseInt( prompt("Enter the dividend of a number") );
   var r, q;

   r = n2 % n1;
   q = Math.floor(n2 / n1);
```

```
    document.write("Quotient is" + q);

</script>
</body>
</html>
```

Q.4 Write the function using Javascript to enter the day number and display the day name.

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">

var daynum =  parseFloat(prompt("Enter a day number") ;

   switch(daynum)
   {
     case 1 : document.write(daynum + " is MONDAY <BR>" ) ;
              break;
     case 2 : document.write(daynum + " is TUESDAY <BR>" ) ;
              break;
     case 3 : document.write(daynum + " is WEDNESDAY <BR>" ) ;
              break;
     case 4 : document.write(daynum + " is THURSNDAY <BR>" ) ;
              break;
     case 5 : document.write(daynum + " is FRIDAY <BR>" ) ;
              break;



     case 6 : document.write(daynum + " is SATURDAY <BR>" ) ;
              break;
     case 7 : document.write(daynum + " is SUNDAY <BR>" ) ;
              break;
    default: document.write(daynum + " is NOT VALID DAYNUMBER <BR>") ;
   }

</script>
</body>
</html>
```

Q.5 Write the function using Javascript to print the table of number entered by the user.

Solution:

```
<html>

<body>

<script language="javascript" type="text/javascript">
```

Check Your Progress

Short Answer Questions

Q.1 Mention the capabilities of OOPs.

Q.2 Define : a) Inheritance b) Encapsulation c) Polymorphism

Q.3 Which method is used to join 2 strings?

Q.4 Write some methods which are used for formatting?

Q.5 Declare one variable name that stores the value of your firstname.

Multiple Choice Questions

Q.1 Which is not the Object oriented programming feature?
A) Inheritance B) Encapsulation C) Polymorphism D) Formatting

Q.2 What is the length of the string x = “COMPUTER”?
A) 6 B) 7 C) 8 D) 9

Q.3 Which keyword is used to declare a string object?
A) String B) str C) string D) any other

True or False

Q.1 bold() Creates a string to be displayed in a big font as if it were in a <big> tag.

Q.2 italics() Causes a string to be italic, as if it were in an <i> tag.

Q.3 blink() Creates an HTML hypertext link that requests another URL.

Find the Output

Q.1 Consider the following code fragment and find the output.

```
var sr ="AbCD Char";
document.write("<p>" + sr + "</p>");
document.write("<p>" + sr.toUpperCase() + "</p>");
document.write("<p>" + sr.toLowerCase() + "</p>");
document.write("<p>" + sr + "</p>");
```

Q.2 Consider the following code fragment and find the output.

```
var sr ="AbCD Char";
document.write("<p>" + sr.substr(2,4) + "</p>");
```

Q.3 Consider the following code fragment and find the output.

```
var str = new String( "This is program" );

  document.write("str.charAt(0) is:" + str.charAt(0));
```

Practical Exercises with Solution:

Q.1 Write the code using Javascript to convert uppercase characters into lowercase and vice-versa.

Solution:

```
<html>
<body>
<script>
var txt="Hello World!";
document.write("<p>" + txt.toUpperCase() + "</p>");
document.write("<p>" + txt.toLowerCase() + "</p>");
document.write("<p>" + txt + "</p>");
</script>
</body>
</html>
```

Q.2 **Write the code to obtain the following output.**

```
Original value1 is ABc

Original value2 is DEFg

Changed value is ABcDEFg
```

Q.3 Write the code using Javascript to find the substring of a given string. Then join the 2 strings and store them into another variable.

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">
```

```
  var sub = x.substr(4);
   var res = x.concat(sub);
   document.write("<p>" + "Substring is " + sub +"</p>");
   document.write("<p>" + "joined string is " + res +"</p>");
</script>
</body>
</html>
```

Q.4 Write the code to change the font color as RED.

```
<html>

<head>

<title>JavaScript String fontcolor() Method</title>

</head>

<body>

<script type="text/javascript">

var str = new String("Hello world");

alert(str.fontcolor( "red" ));

</script>

</body>

</html>
```

Q.5 Write the code to display the output as given below
Hello world

Check Your Progress

Short Answer Questions

Q.1 Expain the significance of Math Object.

Q.3 Mention some of the methods of Math object.

Multiple Choice Questions

Q.1 Which method do not belong to math object?
A) abs() B) pow() C) sqrt() D) minus()

Q.2 Which method is used to find the minimum value from the given list of values?
A) math.pow() B) math.max() C) math.min() D) math.less()

Q.3 Which method is used to find the random values between 0 and 1?
A) math.random() B) math.pow() C) math.square() D) math.round()

True or False

Q.1 Math object is used for mathematical functions.

Q.2 Math.abs() is used to find the absolute value of number entered by the user.

Q.3 At once , we can generate 2 Random numbers .

Find the Output

Q.1 What value will be displayed for the variable '**value**' for the code mentioned below

```
var value = Math.max(10, 20, -1, 100);
document.write("First Test Value : " + value );
```

A) 10 B) 20 C) -1 D) 100

Q.2 What value will be displayed for the variable '**value**' for the code mentioned below

```
var value = Math.pow(0, 10);
document.write("<br />Result is : " + value );
```

A) 10 B) 20 C) 0 D) 100

```
var value = Math.round( -20.3 );
document.write("<br />Fourth Test Value : " + value );
```

A) 10 B) -20 C) 3 D) -20.3

Practical Exercises with solution:

Write the code using Javascript to find the minimum and maximum values from the values entered by the user.

Q.1 **Solution:**

```
<html>

<body>

<script language="javascript" type="text/javascript">



   var n1 =  parseInt(prompt("Enter the first number"));

   var n2 =  parseInt(prompt("Enter the second number"));

   var n3 =  parseInt(prompt("Enter the third number"));

   var small,large;

   small=Math.min(n1,n2,n3);

   large=Math.max(n1,n2,n3);

   document.write("<p>" + "smaller no. is " + small +"</p>");

   document.write("<p>" + "larger no. is " + large +"</p>");



</script>

</body>

</html
```

Q.2 Write the code using Javascript to generate random numbers.

```
<html>

<body>

<script language="javascript" type="text/javascript">

   var num;

   num = Math.random();

 document.write("<p>" + "random no. generated is " + num +"</p>");

</script>

</body>

</html
```

Q.3 Write the code using Javascript to find the square root of a number entered by the user.

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">

   var num;
   num = prompt(parseFloat("Enter the number"));
   var sq = Math.sqrt(num);
   document.write("<p>" + "Squreroot of a number is " + sq +"</p>");


</script>
</body>
</html
```

Q.4 Write the code using Javascript to enter a float value from the user and find the rounded value for that number.

Solution:

```
<html>
<body>
<script language="javascript" type="text/javascript">

   var num;
   num = prompt(parseFloat("Enter the number to be rounded off"));
```

```
 document.write("<p>" + "Rounded number is " + r1 +"</p>");

</script>
</body>
</html>
```

Q.5 Write the code to display the output as given below

```
<html>
<body>
<script language="javascript" type="text/javascript">

   var num;
   num = prompt(parseFloat("Enter the number to be rounded off"));
   var r1 = Math.round(num);
   document.write("<p>" + "Rounded number is " + r1 +"</p>");


</script>
</body>
</html>
```

Check Your Progress

Short Answer Questions

Q.1 What is an Event?

Q.2 Give some examples of event

Q.3 What is the difference between the On Mouseover & On Mouseout Event?

Multiple Choice Questions

Q.1 Which event is fired when the element changes?

A) Onclick() B) onsubmit() C) onchange() D) onreset()

Q.2 Which event is fired , when the form is reset?

A) Onclick() B) onsubmit() C) onchange() D) onreset()

Q.3 Which event is fired , when mouse pointer moves out of an element?

A) Onclick() B) onsubmit() C) onchange() D) onmouseout()

True or False

Q.1 onsubmit() event is fired when we submit a form by clicking a button.

Q.2 Clicking a button is not an event.

Q.3 Resizing a window is an example of an event.

Practical Exercises with solution:

Q.1 Write the code using Javascript to concatenate 2 words entered by the user on clicking the button.

Solution:

```
<html>
 <body>
    <script language="javascript" type="text/javascript">
       document.write("<p>"+"LETS CONCATENATE 2 WORDS "+"</p>");
function JOIN()
 {
    alert("Lets join 2 words");
    var ch1=prompt("Enter first name");
    var ch2=prompt("Enter second name");
    document.write("joining is " + ch1 +ch2);
}
</script>
<input type="button" onclick="JOIN()" value="calculate" />
</body>
</html>
```

Q.2 Write the code using javascript to show the working of onsubmit event.

Solution:

```
<html>
<body>

<p>Click the "Submit" button to submit the form.</p>

<form onsubmit="myFunction()" action="form_action.asp">
        Firstname: <input type="text" name="fname" value="Donald"><br>
        Lastname: <input type="text" name="lname" value="Duck">
        <br><br>
      <input type="submit" value="SUBMIT">
</form>

<script>
 function myFunction()
{
   alert("The form will be submitted");
}
</script>

</body>
</html>
```

Q.3 Write the code using javascript to show the working of reset event.

Solution:

```
<html>

<body>

<p>Enter some text in the fields below, then press the "Reset" button to reset the form.</p>

<form onreset="myFunction()">

Firstname: <input type="text" name="fname" value=" "><br>

Lastname: <input type="text" name="lname" value=" ">

<br><br>

<input type="reset" value="Reset">

</form>

<script>

function myFunction()

{

      alert("The form will be reset. Enter Again ");

}

</script>

</body>

</html>
```