"CODING FORMS", THE OUTSTANDING SYSTEM OF EXPANDING THE HORIZON OF ARCHITECTURE

Professor Bijon B. Sarma

ABSTRACT : Architecture deals with superior quality living and working accommodation. For this purpose they incessantly conduct studies and researches for creating new FORMS. Unfortunately only a negligible number of Architectural Forms have identifying names. Modern computer and drawing technologies have made it possible to design innumerable new forms. But even this one cannot solve the problem because it is not possible for any human being to keep so many names in memory. In this situation two teachers have suggested some simple principles for Coding Architectural Forms on logical basis. In this system, the learners among others would have to know the basic principles of generating various types of Forms and Coding those. The codes would help in instantly visualizing and conveying the forms.

INTRODUCTION :

A Form may be taken to be a solid created by a number of layers of Planes. The process in which the layers are arranged to form the Form may be said to be the maneuvering process. With this basis understanding, it is possible to formulate the Principles for Coding various known and imagined Forms. The basic principles for Coding include 4 steps. These are :

Step 01	To specify **major appearance** of the Form.	Mention the rough **proportion** (Height to bulkiness, where bulkiness is the maximum extent of breadth) Example : 1:2, 1: 3 etc.
Step 02	To specify the First or lowest **Point, Line** or **Plane.**	Specify the **initial** (lowest) **plane.** Mention whether it is a Point or Line or a Plane. Code of plane is mentioned in terms of : (a) Type of line used (C for Curve line, S for Straight line, V for V-shapes etc.), (b) Number of lines used (like 1, 2, 3 etc.) and (c) Nature of line (Mild, Acute, Thin, Deep etc.)
Step 03	To specify the M**aneuvering process.**	The Codes of maneuvering Processes may be : PRO, TWIST, SPI and ROT, each of which may have STRT/DRUM, CON or DIV versions. Their surfaces may be normal (Nil), Bulged (b) or Depressed (d). In case of Twist, Spiraling or Rotation, these may be Clock-wise or Anti-clockwise.
Step 04	To specify the last or highest **Point, Line or Plane.**	Same as explained above in Step 02.

Others	(a) The term "and Reverse" after any Code would indicate that the previous Code would be repeated in the reverse way. (b) Similarly, the term "and Repeat" would mean that the previous Code would be repeated in the same way. (c) In case of compound forms, the Form first needs to be segmented by perceivable sections. The proportion of each section will be mentioned and the each segment would be described by Code.

So, the jobs to be done for preparing the Principles for Coding of Forms are : (01) Specifying major appearance of the form through Proportion of major dimensions, (02) Coding the various types of Planes and (03) Coding the various types of Maneuvering processes.

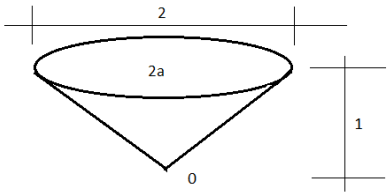
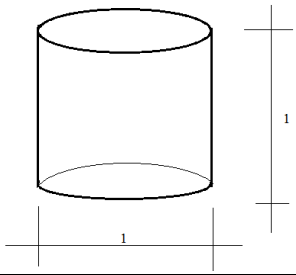
EXAMPLE OF CODING FORM :

Before going into specific discussion we shall present two simple examples to show how various Forms can be coded by following the above mentioned four steps. Let the two forms are : (a) An inverted Cone and (b) A Drum.

(a) (1) As for major appearance, let the proportion of the major dimensions of the Cone (Height : Bulkiness) is 1:2. (2) As for initial plane, the Cone starts from a Point and its Code is Point. (3) The process in which the point is transformed into a large circular plane includes –Progression and Divergence. The short name or Code of such process is, PRO-DIV. (4) The final plane is a circular plane. We shall later see that the Code for such a circular plane is C1. Combining all these, the Code of the inverted Cone is : **1:2, Point.PRO-DIV. C1.**

(b) In the same way (1) The major appearance or Proportion of the major dimensions of the Drum is 1:1. (2) It starts with a circular plane whose code is C1. (3) The process in which the circular plane reaches at the uppermost level is : Progression in a Straight-forward manner. The short name or Code of the maneuvering process is : PRO-STRT. (4) Finally, the final plane is a circular plane (C1). However, there is no need for mentioning this Code, because it is the same as the Starting plane. We shall later see that in case of Straight-forward (STRT) or Drum-like (DRUM) growth, the end plane is the same as the initial plane.

Figure No. 01: PRINCIPLES FOR CODING FORMS.

FORM	EXPLANATION	CODE
	(1) Proportion: Height : Bulkiness. (2) Code of the starting plane (Point). (3) Code of maneuvering process : PRO-DIV. (4) Code of and ending plane (C1).	**1:2, Point.PRO-DIV. C1.**
	(1) Proportion: Height : Bulkiness. (2) First plane : Circular plane (C1). (3) Maneuvering processes: PRO-DIV. (4) Code of and ending plane. (Not required, because it is the same)	**1:1, C1.PRO-STRT.**

STEP 01 : SPECIFYING THE MAJOR APPEARANCE OF THE FORM :

It has already been stated that the major appearance of the Form is expressed in terms of Proportion of its major dimensions, starting with the Height. The second dimension expressed as multiple of this one will be the maximum width, diameter etc. However, the smallest dimension will be expressed by 1. So, the Proportion will be - Height : Diameter = 1 : 2 or 3: 1 etc.

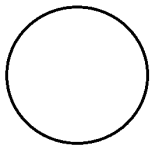
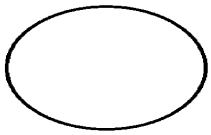
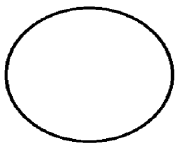
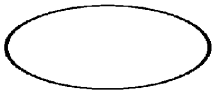
STEP 02 : CODING THE FIRST OR LOWEST POINT, LINE OR PLANE :

The physically conceivable Architectural Forms are generated from points. Points make lines, lines make Planes and Planes, Solids. The Planes and solids are represented on flat papers by drawing the peripheral lines. Naturally various types of planes may be categorized on the basis of (a) Type of lines (e.g. Straight, Curve etc.), (b) Number of lines (e.g. 1,2,3 etc.) and (c) Nature of the curve lines (e.g. Normal /Abnormal, Bulged/Depressed) etc. The terms mentioned above can be abbreviated as follows :

01. Straight (S), 02. Curve (C), 03. Normal (Nil), 04. Abnormal (a),
05. Bulged (b), 06. Depressed (d), 07. Mild (Mild), 08. Acute (Acute),
09. Thin (Thin), 10. Deep (Deep) etc.

Now to start with, only one Form can be generated by using one normal curve line, and it is a circle. So, the Code of a Circle can be : C1. The circle may be oval-shaped, which is abnormal. So an Oval may be coded as S1.a. The Oval shapes may be somewhat Mild or Acute and coded accordingly. These principles lead to the following codes of the shapes of the planes shown here under.

Figure No. 02: SHAPES GENERATED BY ONE CURVE LINE AND THEIR CODES.

Shape				
Code	**S1**	**S1.a**	**S1.a. Mild**	**S1.a. Acute**

In the same way, the various shapes generated by more than one number of Curve lines may be the following :

Figure No. 03: SHAPES GENERATED BY CURVE LINES AND THEIR CODES

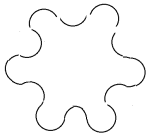
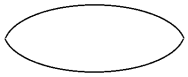
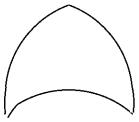
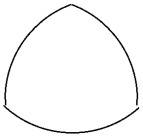
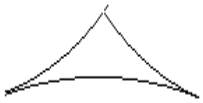
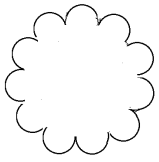
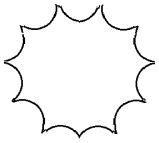
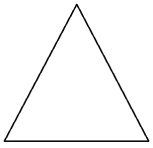
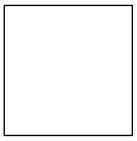
Shape				
Code	**C1.bd (6)**	Code: **C2.b**	**C2.bd.**	**C3.bbd**
How created	1 curve line, alternately bulged and depressed 6 times.	2 curve lines, bulged	2 curve line, bulged and depressed	3 curve lines, 2 bulged-1 depressed
Shape				
Code	**C3.b**	**C3.b**	**C11.b**	**C11.d**
How created	3 curve lines, all depressed	3 curve lines, bulged	11 curve lines, bulged outside.	11 curve lines, depressed inside.

Figure No. 04: SHAPES GENERATED BY STRAIGHT LINES AND THEIR CODES.

Shape				

Code	S3	S4	S22	S22.a
How created	3 Equal straight lines	4 equal straight lines	2 one size + 2 another size straight lines	Abnormal rectangle

Figure No. 05: STAR-SHAPES GENERATED BY V'S AND THEIR CODES.

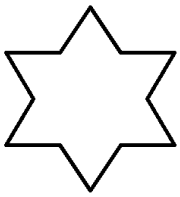
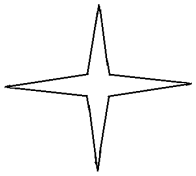
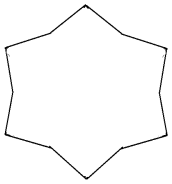
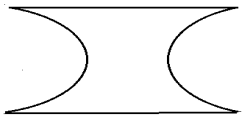
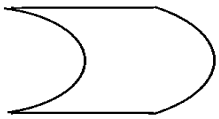
Shape				
Code	**V4**	**V6**	**V4.Acute**	**V6.Mild**
How created	Created by 4 V's	Created by 6 V's.	Created by 4 Acute-natured V's.	Created by 6 Mild-natured V's.

Figure No. 06: SHAPES GENERATED BY CURVE AND STRAIGHT LINES AND THEIR CODES.

Shape			
Code	**S2C2. b -Mild**	**S2C2d -Mild**	**S2C2bd- Mild**
How created	All curve lines bulged.	All curve lines depressed	Curve lines 1 bulged and 1 depressed
Acute versions of the above shapes.			
Shape			
Code	**S2C2b –Acute**	**S2C2d-Acute**	**S2C2bd- Acute**

STEP 03. CODING THE MANEUVERING PROCESS :

Maneuvering processes are the processes in which one shape (including point and line) transforms into another shape. In case of Forms these may take place in the following 4 major ways :

(A) PROGRESSING (abbreviated by PRO) i.e. by translational movement of the plane one direction. In this book only upward direction has been taken into consideration.

(B) TWISTING (abbreviated by TWIST) i.e. by simultaneous pulling and twisting the plane through certain distance.

(C) SPIRALING (abbreviated by SPI) i.e. by pulling the plane to create some thickness and then spiraling this thick plane around.

(D) ROTATING (abbreviated by ROT) i.e. by rotating the plane around its x or y (or, long or short) axes.

Each of the above processes again may take place in a number of special ways. For example :

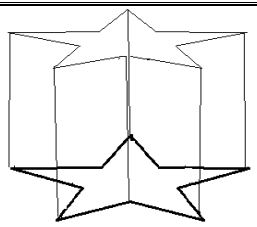
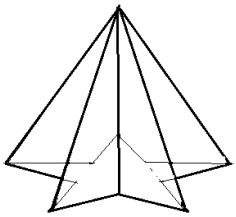
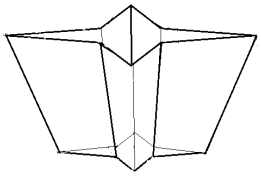
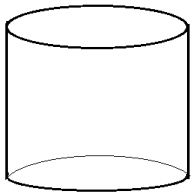
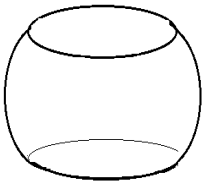
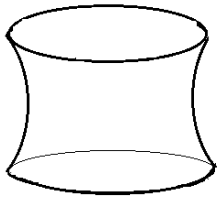
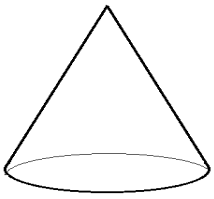
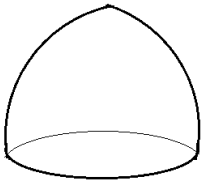
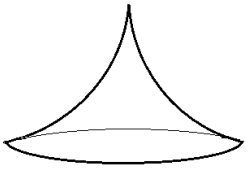
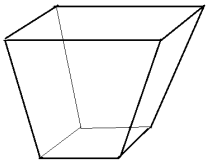
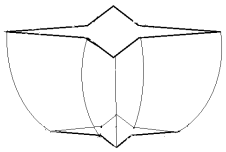
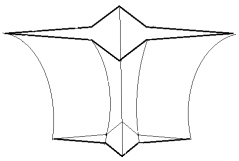
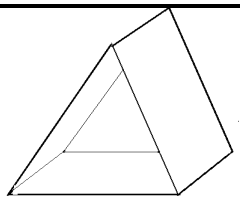
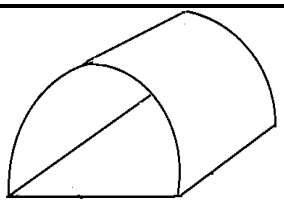
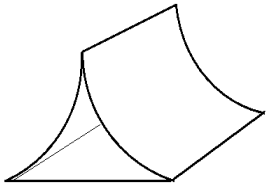
(A) PROGRESSING may take place in (i) Translational or Straight manner (abbreviated by PRO-STRT), (ii) Convergent manner (abbreviated by PRO-CON) or (iii) Divergent manner to end in a bigger plane (abbreviated by PRO-DIV).

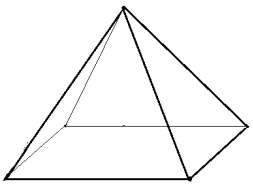
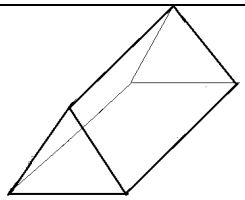
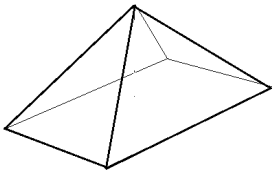
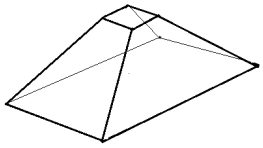
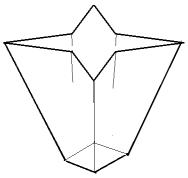
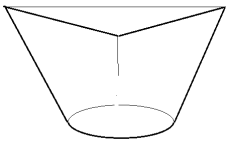
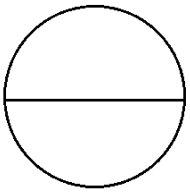
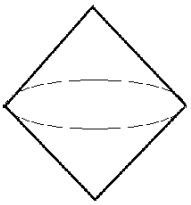
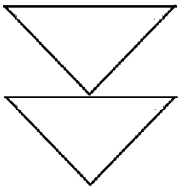
In the above development, the act may take place while the surface may remain normal or plane (abbreviated by nothing), may show bulging (abbreviated by b) or depression (abbreviated by'd'). Again, finally the bulging or depression may be both Mild or Acute.

Figure No. 07: TYPE OF GROWTH, SURFACE APPEARANCE / DEVIATION, EXTENT OF DEVIATION AND CODE OF THE MANEUVERING PROCESS - **PROGRESSION (PRO)**

TYPE OF GROWTH	**SURFACE APPEARANCE/ DEVIATION**	**EXTENT OF DEVIATION**	**CODE** (These Forms may end in another Plane, Point or Line)
(a) STRT. Straight or Parallel.	(i) Plane (Nil)	Nil	**PRO-STRT.**
	(ii) Bulged (b)	Mild	**PRO-STRT.b.Mild**
		Acute	**PRO-STRT.b.Acute**
	(iii) Depressed(d)	Mild	**PRO-STRT.d.Mild**
		Acute	**PRO-STRT.d.Mild**
(b) CON- Converging Growth.	(i) Plane (Nil)	Nil	**PRO-CON.**
	(ii) Bulged (b)	Mild	**PRO-CON.b.Mild**
		Acute	**PRO-CON.b.Acute**
	(iii) Depressed(d)	Mild	**PRO-CON.d.Mild**
		Acute	**PRO-CONd.Acute**
(c) DIV. Diverging Growth.	(i) Plane (Nil)	Nil	**PRO-DIV.**
	(ii) Bulged (b)	Mild	**PRO-DIV.b.Mild**
		Acute	**PRO-DIV.b.Acute**
	(iii) Depressed(d)	Mild	**PRO-DIV.d.Mild**
		Acute	**PRO-DIV.d.Acute**

Figure No. 08: EXAMPLES OF FORMS GENERATED BY PROGRESSION (PRO) AND THEIR CODES. (Mild and Acute versions not shown)

Form			
Man. process	PRO-STRT	PRO-CON.	PRO-DIV.
Code	**1:1. V5.PRO-STRT.**	**1:1. V-5.PRO-CON.Point**	**1:1. PRO-DIV.**
Form			
Man. process	PRO-STRT	PRO- STRT.b	PRO- STRT.d
Code	**1:1. C1. PRO-STRT.**	**PRO- STRT.b**	**PRO- STRT.d**
Form			
Man. process	PRO-CON	PRO-CON.b	PRO-CON.d
Code	**1:1. C1.PRO-CON.Point**	**1:1. C1.PRO-CON.b.Point**	**1:1. PRO-CON.d.Point**
Form			
Man. process	PRO-DIV.	PRO-DIV.b	PRO-DIV.d
Code	**1:1. S4.PRO-DIV.**	**1:1. V4.PRO-DIV.b**	**1:1. V4.PRO-DIV.d**
Form			
Man. process	PRO-CON.Line	PRO- CON.b.Line	PRO- CON.d.Line
Code	**1:1. S4. PRO-STRT.**	**1:1. S4.PRO-CON.b. Line**	**1:1.PRO- CON.d.Line**

Form			
Man. process	PRO-CON.Point	PRO-CON.Line	PRO-CON.Point
Code	**1:1. S4. PRO-CON.Point**	**1:2. S22.PRO-CON.Line**	**1:2.S22.PRO-CON.Point**
Form			
Man. process	**PRO-CON.**	**PRO-DIV.**	**PRO-DIV.**
Code	**1:2. S22. PRO-CON.S4**	**1:1. S4.PRO-DIV.V4**	**1:1.C1.PRO-DIV. S3**
Form			
Man. process	**PRO-DIV. b and Reverse**	**PRO-DIV. and reverse**	**PRO-DIV.**
Code	**1:2. Point. PRO-DIV.b. C1.and Reverse** It is a Sphere. However, a shorter Code for Sphere is **C1.ROT.**	**1:1. Point.PRO-DIV.C1. and Reverse**	**1:1. Point.PRO-DIV.C1. and Reverse**

(B) TWISTING

Twisting may be considered as the act of pulling the initial plane through some distance and then twisting it. Twisting can take place either in the (a) Clock-wise (abbreviated as 'Clock') or in the (b) Anti-clockwise (abbreviated as 'Anti') directions.

There happens no straight or parallel movement in cases of Twisting and Spiraling and the increase resembles a "Drum". We shall use the term 'Drum' in place of 'Strt'. Other than drum-like growth, these may be of Converging (abbreviated by CON) or Diverging (abbreviated by DIV) types. Also the surface may be normal (abbreviated by nothing), 'Bulged' (abbreviated as 'b') or 'Depressed' (abbreviated as 'd'). Also the deviations may be 'Mild' or 'Acute'. In addition the Twisting may take place either in the Clockwise (abbreviated by Clock) or Anti-

clockwise (abbreviated by Anti-clock) directions. Incorporating all these, we may get the following Codes for various maneuvering processes related to Twisting.

Figure No. 09: TYPE OF GROWTH, SURFACE APPEARANCE / DEVIATION, EXTENT OF DEVIATION AND CODE OF THE MANEUVERING PROCESS - TWISTING **(TWIST)**

TYPE OF GROWTH	**SURFACE APPEARANCE/ DEVIATION**	**DIRECTION**	**EXTENT**	**CODE**
(a) DRUM type growth	(i) Plane (Nil)			**TWIST.DRUM.**
	(ii) Bulged (b)	Clock	Mild	**TWIST.DRUM.b.Clock.Mild**
			Acute	**TWIST.DRUM.b.Clock.Acute**
		Anti	Mild	**TWIST.DRUM.b.Anti.Mild**
			Acute	**TWIST.DRUM.b.Anti.Acute**
	(iii) Depressed(d)	Clock	Mild	**TWIST.DRUM.d.Clock.Mild**
			Acute	**TWIST.DRUM.d.Clock.Acute**
		Anti	Mild	**TWIST.DRUM.d.Anti.Mild**
			Acute	**TWIST.DRUM.d.Anti.Acute**
(b) CON-Converging Growth.	(i) Plane (Nil)			**TWIST-CON**
	(ii) Bulged (b)	Clock	Mild	**TWIST-CON.b.Clock.Mild**
			Acute	**TWIST-CON.b.Clock.Acute**
		Anti	Mild	**TWIST-CON.b.Anti.Mild**
			Acute	**TWIST-CON.b.Anti.Acute**
	(iii) Depressed(d)	Clock	Mild	**TWIST-CONd.Clock.Mild**
			Acute	**TWIST-CONd.Clock.Acute**
		Anti	Mild	**TWIST-CONd.Anti.Mild**
			Acute	**TWIST-CONd.Anti.Acute**
(c) DIV. Diverging Growth.	(i) Plane (Nil)			**TWIST-DIV.**
	(ii) Bulged (b)	Clock	Mild	**TWIST-DIV.b.Clock.Mild**
			Acute	**TWIST-DIV.b.Clock.Acute**
		Anti	Mild	**TWIST-DIV.b.Anti.Mild**
			Acute	**TWIST-DIV.b.Anti.Acute**
	(iii) Depressed(d)		Mild	**TWIST-DIV.d.Mild**
		Clock	Mild	**TWIST-DIV.d.Clock.Mild**
			Acute	**TWIST-DIV.d.Clock.Acute**
		Anti	Mild	**TWIST-DIV.d.Anti.Mild**
			Acute	**TWIST-DIV.d.Anti.Acute**

(C) SPIRALING :

Spiraling may be defined as 'pulling the lower plane through some distance and then to spiral (i.e. rotate and pull) this thickness around. All the natures and characteristics of movements of Spiraling are the same or similar to those of Twisting. Naturally the Codes are also similar.

Figure No. 10: TYPE OF GROWTH, SURFACE APPEARANCE / DEVIATION, EXTENT OF DEVIATION AND CODE OF THE MANEUVERING PROCESS - SPIRALING **(SPI)**

TYPE OF GROWTH	**SURFACE APPEARANCE/ DEVIATION**	**DIRECTION**	**EXTENT**	**CODE**
(a) DRUM type growth	(i) Plane (Nil)			**SPI-DRUM.**
	(ii) Bulged (b)	Clock	Mild	**SPI-DRUM.b.Clock.Mild**
			Acute	**SPI-DRUM.b.Clock.Acute**
		Anti	Mild	**SPI-DRUM.b.Anti.Mild**
			Acute	**SPI-DRUM.b.Anti.Acute**
	(iii) Depressed(d)	Clock	Mild	**SPI-DRUMd.Clock.Mild**
			Acute	**SPI-DRUMd.Clock.Acute**
		Anti	Mild	**SPI-DRUMd.Anti.Mild**
			Acute	**SPI-DRUMd.Anti.Acute**
(b) CON. Converging Growth.	(i) Plane (Nil)			**SPI-CON.**
	(ii) Bulged (b)	Clock	Mild	**SPI-CON.b.Clock.Mild**
			Acute	**SPI-CON.b.Clock.Acute**
		Anti	Mild	**SPI-CON.b.Anti.Mild**
			Acute	**SPI-CON.b.Anti.Acute**
	(iii) Depressed(d)	Clock	Mild	**SPI-CONd.Clock.Mild**
			Acute	**SPI-CONd.Clock.Acute**
		Anti	Mild	**SPI-CONd.Anti.Mild**
			Acute	**SPI-CONd.Anti.Acute**
(c) DIV. Diverging Growth.	(i) Plane (Nil)			**SPI-DIV.**
	(ii) Bulged (b)	Clock	Mild	**SPI-DIV.b.Clock.Mild**
			Acute	**SPI-DIV.b.Clock.Acute**
		Anti	Mild	**SPI-DIV.b.Anti.Mild**
			Acute	**SPI-DIV.b.Anti.Acute**
	(iii) Depressed(d)		Mild	**SPI-DIV.d.Mild**
		Clock	Mild	**SPI-DIV.d.Clock.Mild**
			Acute	**SPI-DIV.d.Clock.Acute**
		Anti	Mild	**SPI-DIV.d.Anti.Mild**
			Acute	**SPI-DIV.d.Anti.Acute**

Figure No. 11: SOME EXAMPLES OF TWISTED AND SPIRALLED FORMS.

Form			
Process	TWIST-DRUM	TWIST-CON.	TWIST-DIV.
Code	**4:1. S4.TWIST-**DRUM**.Clockwise**	**3:1. C1.TWIST-**CON.Clockwise	**4.1. S22.TWIST.DIV. Clockwise**
Form			
Process	SPI-DRUM	SPI-CON.	SPI-DIV.
Code	**2:1. C1.SPI-DRUM.Anticlock**	**3:1. C1.SPI-CON.Anticlock.Point**	**1:1. C1.** SPI-**DIV.Clockwise**

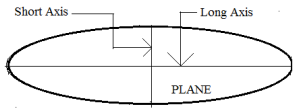
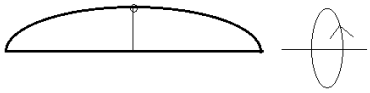
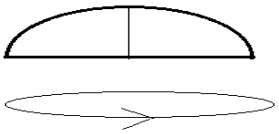
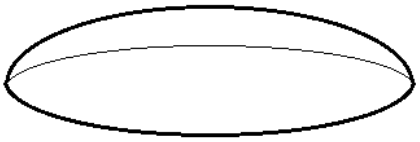
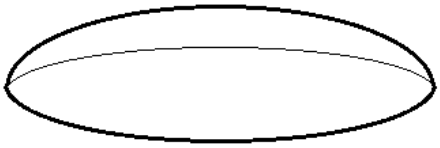
(D) ROTATING :

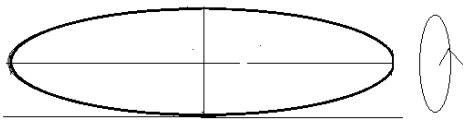
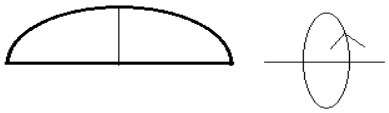
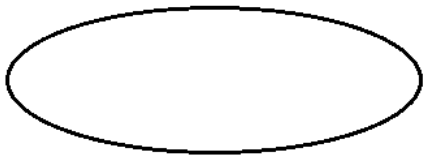
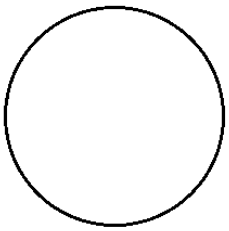
Rotation may take place around two axes, i.e. (a) x-axis or LONG axis (abbreviated as ROT-LONG) and (ii) y-axis or SHORT axis (abbreviated by ROT-SHORT) of the plane. Rotation

around z-axis is meaningless, because all such rotations result in the creation of a C1, i.e. a circular plane.

Since the plane is supposed to stay on ground, it can rotate through 180 degrees only. So, it would be meant by Code - HALF.ROT (Half Rotation). Full rotation can take place at certain height from the ground. It will be known as ROT (i.e. Full rotation). Generation of Forms by various types of rotations have been shown in Figure No. 12.

Figure No. 12: GENERATION OF FORMS BY ROTATION

		
PLANE (S1.a)	Code : **HALF-ROT-LONG** (Rotation against long axis)	Code : **HALF-ROT-SHORT** (Rotation against short axis)
		
(01) Form generated by **ROT-LONG** (Similar to half-cut Rugby-ball) Code : **1:4.C1a. ROT.LONG.**	(02) Form generated by **ROT- SHORT** (Similar to half-cut Tyre) Code : **1:4.C1a. ROT.SHORT.**	

	
Code : **ROT-SHORT** (Rotation against long axis)	Code : **ROT-LONG** (Rotation against short axis)
	
Form generated by **ROT-SHORT** (Similar to Rugby-ball) Code : **1:4.C1a. ROT.LONG.**	(02) Form generated by **ROT-LONG** (or SHORT (Sphere) Code : **1:1.C1. ROT.SHORT.**

STEP 04 : CODING THE LAST OR HIGHEST POINT, LINE OR PLANE :

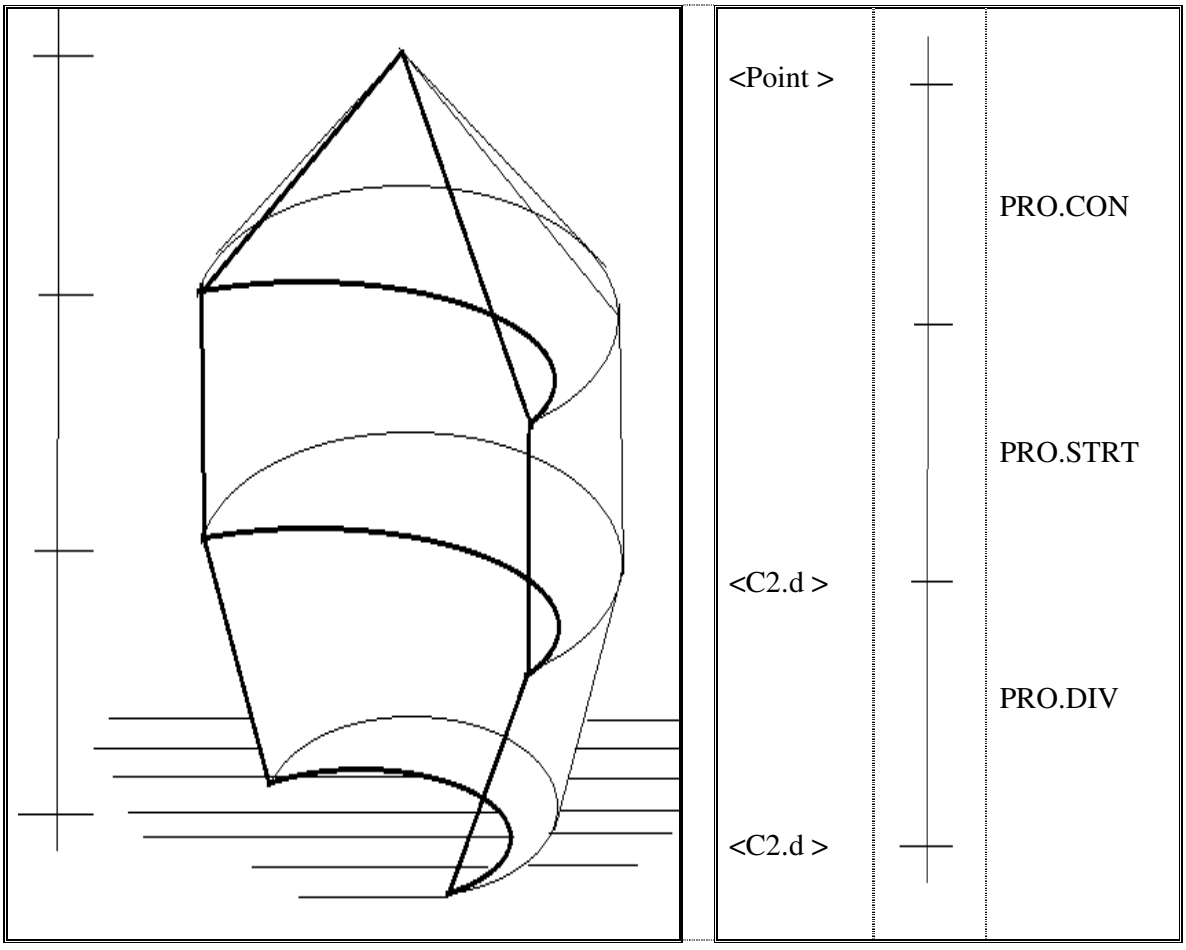
This is exactly the same as Step 02.

EXAMPLES OF PRESENTING FORMS:

We present hereunder two examples of how real or imagined complex Forms can be presented to the students. The examples show the conventional method and also the proposed methods. The proposed methods, however, can be introduced only after the introduction of the system of Coding. These are Proposed Method-01 and Proposed Method 02 as shown below.

Figure No. 13: THREE METHODS OF PRESENTING A FORM TO THE STUDENTS.

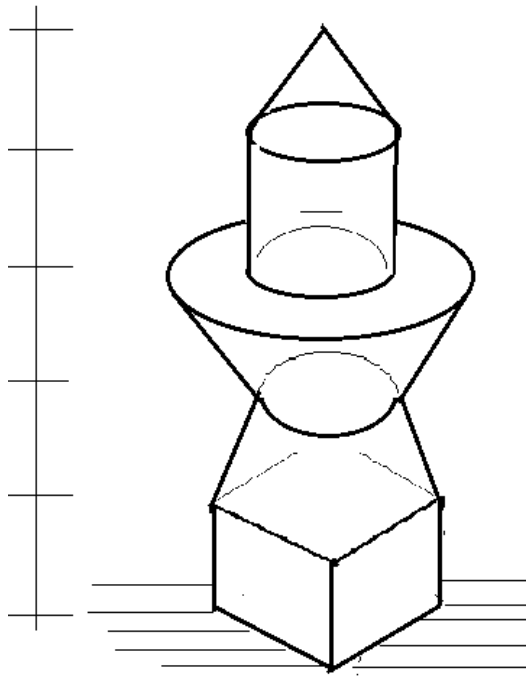
The Form shown above is composed of a small crescent shape at the bottom, it gradually grows with increasing cross-section up to certain level. From here it has got parallel growth with no change in size or shape up to another level. From this level the form converges and finally ends in a point. At present the only way before the teacher to describe this Form is to draw a sketch as shown in Conventional Method below. After Coding of Forms is introduced, the teacher would get two more methods mentioned as Proposed Method 01 and 02 below.



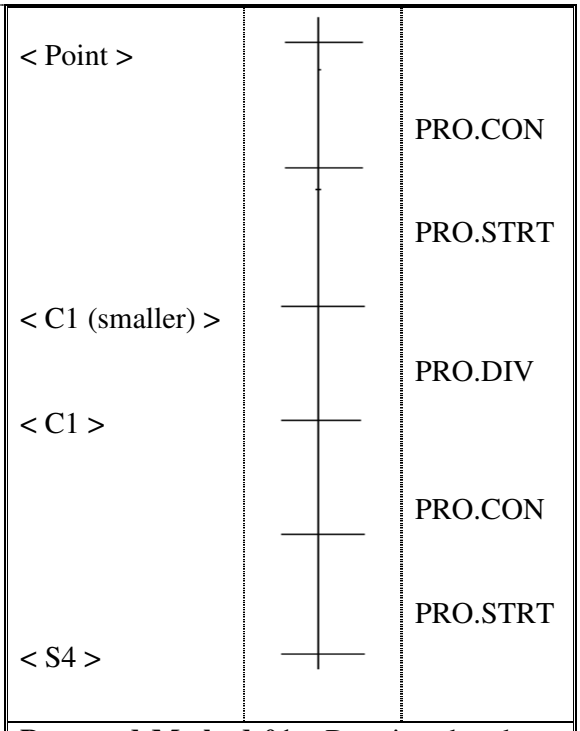
Conventional Method : Drawing the FORM on board.	**Proposed Method 01** : Drawing and writing the above on board.

Proposed Method 02 : Writing or saying the Code :
"Three equal segments. First **- C2.d.PRO.DIV.** Second**- PRO.STRT.** Last**- PRO.CON.Point".**

Another similar example in given in the following figure.



Conventional Method : Drawing the FORM on board



Proposed Method 01 : Drawing the above sketch and writing the Codes on board

Proposed Method 02 : Writing or saying the Code :
"Five equal segments. First- **S4. PRO.STRT.** Second- **PRO.CON. C1.** Third- **PRO.DIV.** Fourth- **C1** (smaller) **PRO.STRT** and Last **- PRO.CON.Point".**

USING CODES IN ARCHITECTURE :

In Architecture the use of Codes for denoting Forms may find ample scopes ion pedagogy and the profession. Quite often the teachers need to explain seen or imagined forms to their students. Similar is the case with the professional architects with their clients or co-workers. In describing such Forms, they use the well known terms like : SPHERE, DRUM or BARREL, DOME, PYRAMID, TETREHEDRON, CUBE etc. These can be also be expressed by drawing a sketch. Drawings at times create confusion. For example:

A Circular plane can often be confused with a sphere, a pyramid with tetrahedron, a cube with a square plane etc.

It is a common experience that only a minute portion of the innumerable number of real and imaginary Forms has got names. Those having no name can be described only through sketch, presentation of pictures or models etc., all of which are hazardous and time consuming. In comparison, Coding on logical basis may be considered as the easiest means of presenting innumerable real and imagined forms. In this article the authors have not endeavored to assign "Codes" to all Forms. On the other hand they have endeavored to explain the principles on which these Codes can be created. After these principles are understood, it will be possible for the teachers to prepare Codes by themselves. Also, since these are based on universal logic, the students or co-workers would be able to instantly visualize the Form from its code.

When the name of the popular object of Form is mentioned by the teacher, it instantly appears as a 'flush' in the minds of the students. However, when the Code of the known or unknown form will be pronounced the student would visualize it in a step by step manner. The process may help in improving the ability of visualization and imagination of the students.

COMPUTER DRAWING OF FORMS FROM CODES :

At present various Shapes or Forms are drawn in computer by using various drawing packages. In all cases, such packages use the constituting lines of the Forms. This takes considerable time because, quite often the lines appear with longer or shorter length and those need to be edited. Also the nature of lines (e.g. curvature, change of angle, direction etc.) need to be manipulated.

In comparison, the Forms can be made readily available from the package by using their Code names. This may save considerable quantity of time. Once the basic form is available, (which may be possible within a second) it may be edited to suit the need. By these considerations the Forms available by their code names from the computer seems much more advantageous than creating those by using various drawing packages.

CONCLUSION :

The professional architects are creating wonderful objects every day for men's use and enjoyment and also for enriching the civilizations of this world. The origin of this creation lies in the generation of Forms. The limitations with the generation and presentation of Forms lies with (a) limited names for identifying Forms and (b) hazards associated with the drawing or

presentation of Forms on boards.

In this age it is possible to prepare a catalogue of all possible forms and assign them names. But that would not solve the problem. It is not humanly possible to remember so many forms. In such a context “A system of Coding the Architectural Forms” seems to be a better solution. What has been explained in this book are not the ‘Codes of Forms”, but the principle to be used for such Coding. Once the principles are known, it will be possible for the experts to prepare the Codes.

At present the teachers mention or discuss only those forms which are convenient to mention, draw, or display. Naturally numerous Forms which do not come under conveniences stay beyond the arena of their discussion.

Needless to mention that the more new Forms are introduced in the academy, the better will be the state of creation in Architecture. In addition to adding vast number of new Forms to Architecture, the system of generating and identifying Architectural Forms by codes may help to increase the capability of imagination of the students of Architecture.

After the codes are made known, in place of just 'seeing' or 'remembering' the known forms, the students would go for imagining new, never-heard and never-seen Forms from their codes. They may initiate and continue discussions on forms among themselves only through codes. This may be advantageous because it would eliminate the time consuming drawings and sketches.

After new and unknown Forms enter the arena of Architecture through Codes, it may be expected that those would gradually enter in the professional works. And finally these would find their ways and uses in the real world. Definitely their materialization would enrich the world of Architecture and of civilization.

THE END