

Problem seeking

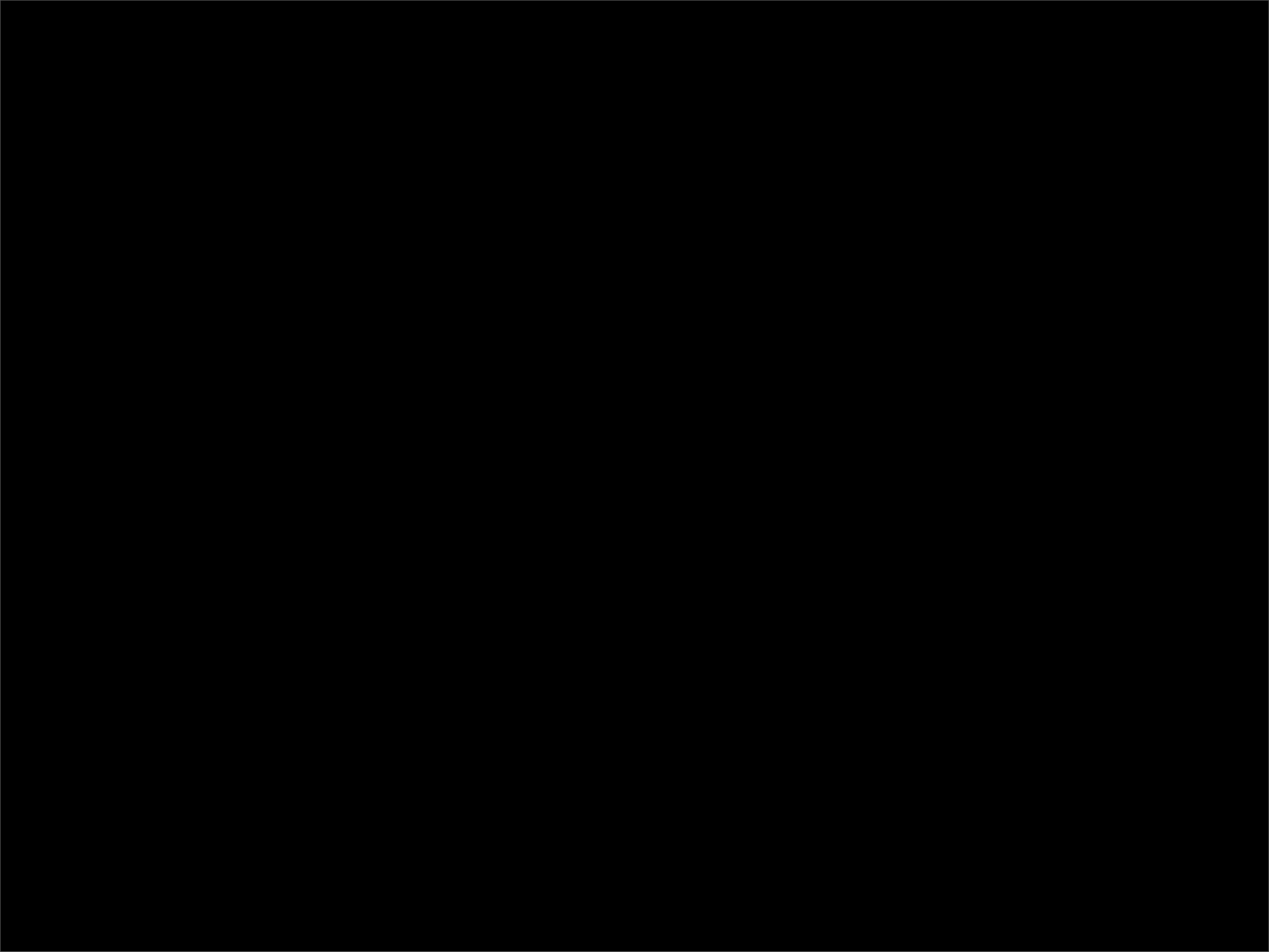
An Architectural Programming Primer

The Primer

- Good Good buildings don't just happen.
- They are planned to look good and perform well, and come about when good architects and good clients join in thoughtful, cooperative effort.
- Programming the requirements of a proposed building is the architect's first task, often the most important.
- There are a few underlying principles that apply to programming — whether the most complex hospital or a simple house.

Programming Process : five-steps process

- 1 Establish **Goals**.
- 2 Collect and analyze **Facts**.
- 3 Uncover and test **Concepts**.
- 4 Determine **Needs**.
- 5 State the **Problem**.



Design Considerations : Four Considerations

Function

Form

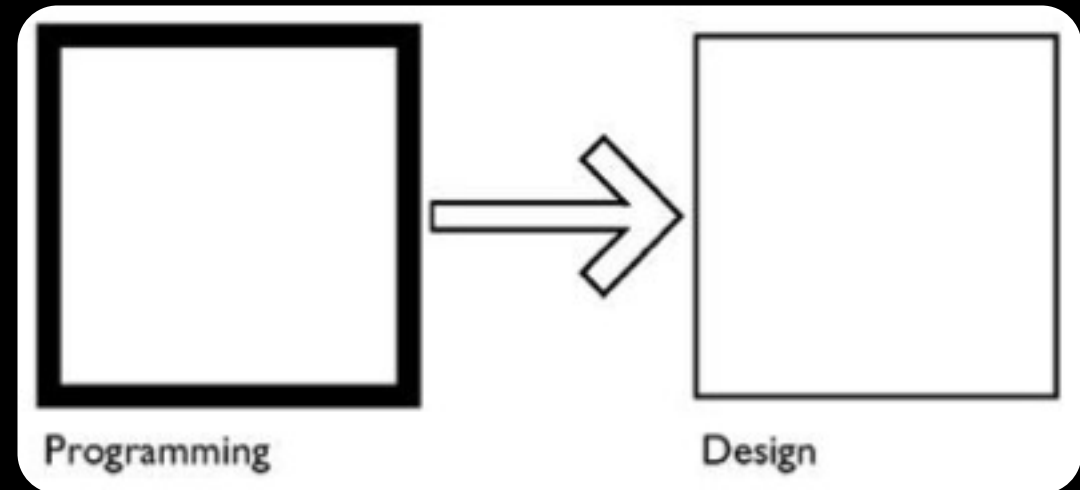
Economy

Time

Architectural programming, therefore, involves an organized method of inquiry... a five-step process... interacting with four considerations.

The Search

Programming is a process.



“A process leading to the statement of an architectural problem and the requirements to be met in offering a solution.” Webster

If programming is problem seeking, then design is problem solving.

These are two distinct processes, requiring different attitudes, even different capabilities. Problem solving is a valid approach to design when, indeed, the design solution responds to the client's design problem. Only after a thorough search for pertinent information can the client's design problem be started. “Seek and you shall define!”

Programmers and Designers

If I were given one hour to save the planet, I would spend 59 minutes defining the problem and one minute resolving it. — Einstein

Programmers and Designers

Programmers must be objective (to a degree) and analytical, at ease with abstract ideas, and able to evaluate information and identify important factors while postponing irrelevant material. Designers can't always do this.

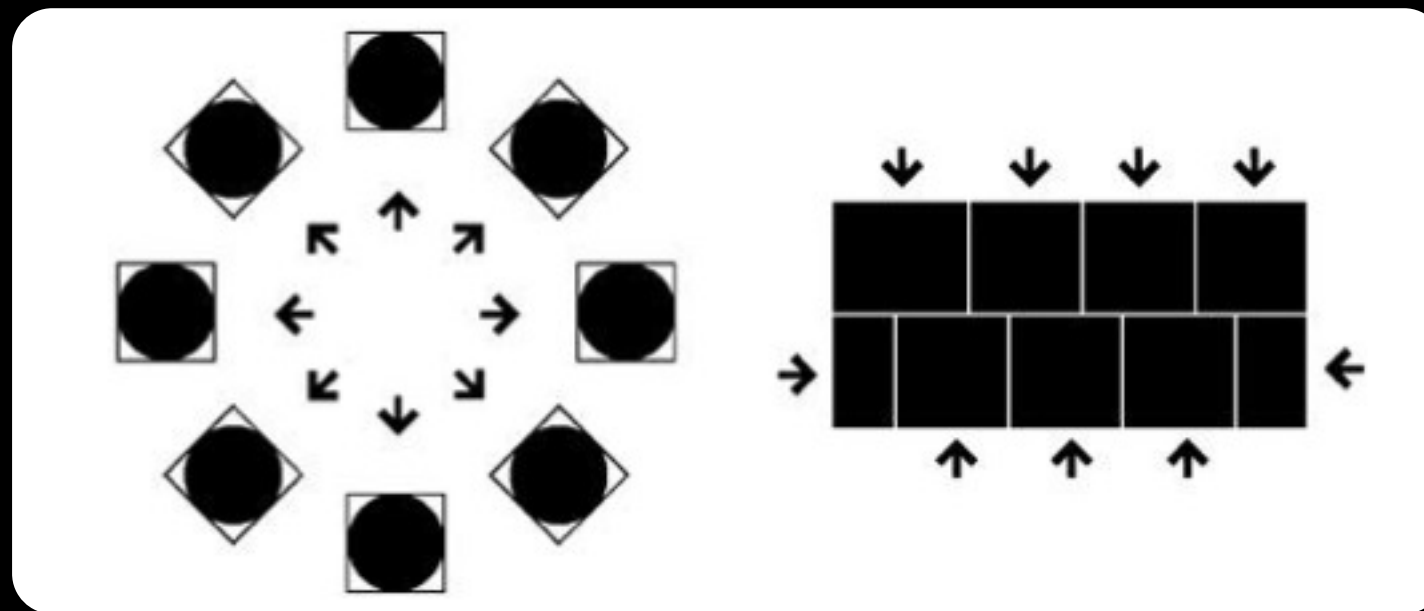
Designers generally are subjective, intuitive, and facile with physical concepts.

Programmers and designers are separate specialists because the problems of each are very complex and require two different mental capabilities, one for analysis, another for synthesis.

It may well be that one person can manage both analysis and synthesis.

If so, he or she must be of two minds and use them alternately. However, for clarity, these different qualifications will be represented by different people— programmers and designers.

Analysis and Synthesis



The total design process includes two stages: analysis and synthesis. In analysis, the parts of a design problem are separated and identified. In synthesis, the parts are put together to form a coherent design solution. The difference between programming and design is the difference between analysis and synthesis.

Analysis and Synthesis

Programming IS analysis. Design IS synthesis.

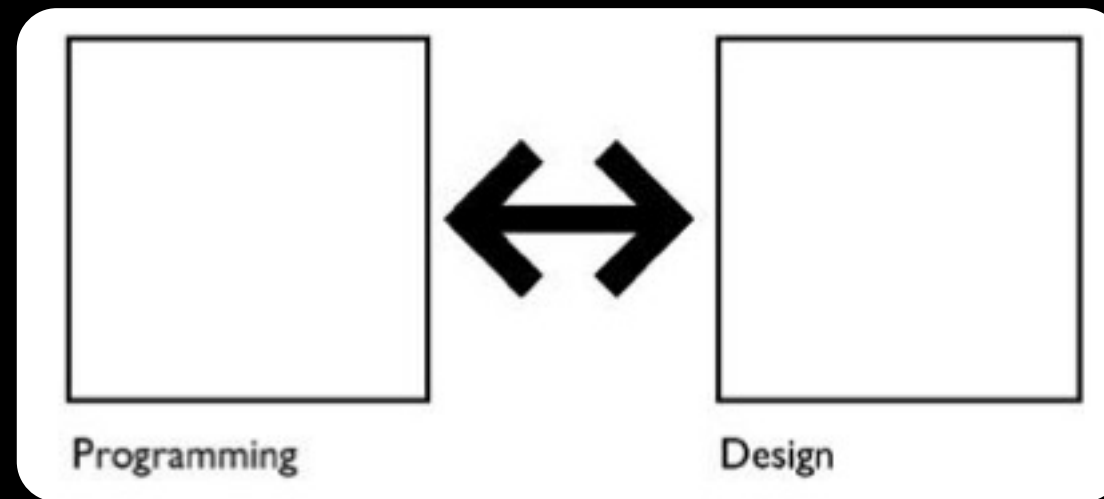
Sometimes I think we arrive at a solution before we know what the problem is. We say: “My next design will be **Round!**,” without logic or analysis. — William Peña

You may not perceive the design process in terms of analysis and synthesis. You may even question problem solving as an approach. You may think of the design process as a creative effort. It is.

But the creative effort includes similar stages: analysis becomes preparation or exposure, and synthesis becomes illumination or insight. The total design process is, indeed, a creative process.

Does programming inhibit creativity? Definitely not! Programming establishes the considerations, the limits, and the possibilities of the design problem. (We prefer “considerations” to “constraints” to avoid being petulant.) Creativity thrives when the limits of a problem are known.

The Separation



- The problem-seeking method described in this book requires **a distinct separation of programming and design.**
- They believe in thorough analysis before synthesis. They know that programming is the prelude to good design — although it does not guarantee it.

The Interface

The product of programming is a statement of the problem. Stating the problem is the last step in problem seeking (programming), and it is also the first step in problem solving (design).

The problem statement, then, is the interface between programming and design.

Since this statement is the first step in design as well as the last step in programming, its composition must be the joint effort of the designer and the programmer.

Programming Five Steps

1 Establish Goals

2 Collect and Analyze Facts

3 Uncover and Test Concepts

4 Determine Needs

5 State the Problem

primarily the search for
pertinent information

feasibility test

distilling what has been found

Programming is based on a combination of interviews and work sessions. Interviews are used for asking questions and collecting data, particularly during the first three steps.

Work sessions are used to verify information and to stimulate client decisions — particularly during the fourth step.

The five steps pose these questions:

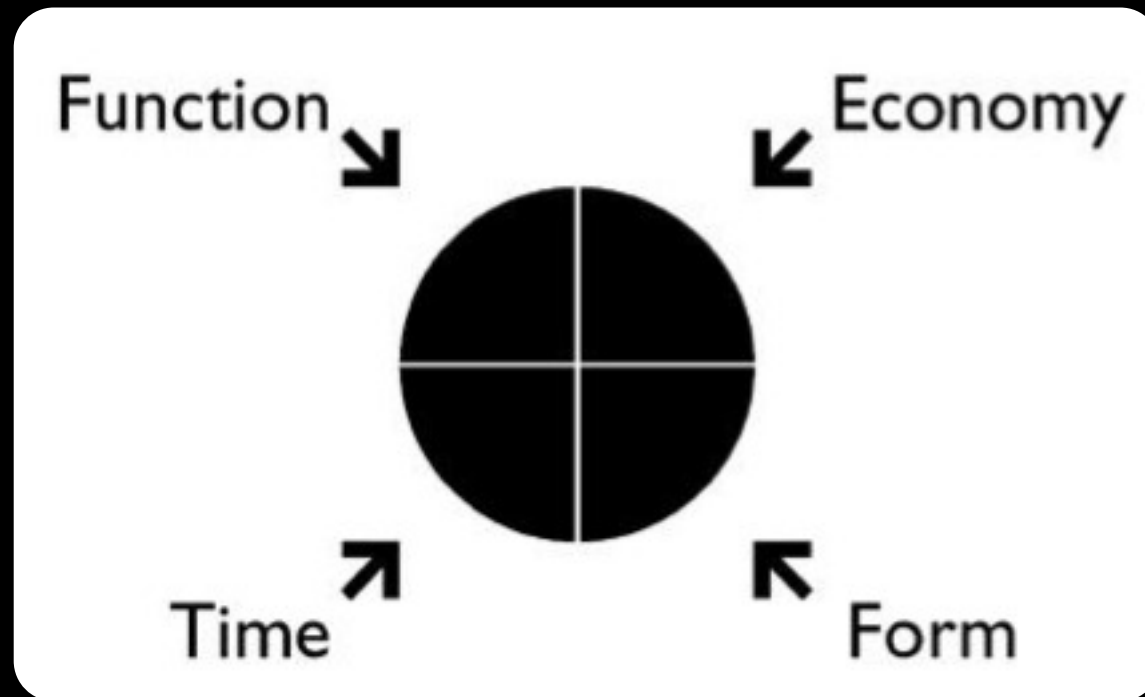
- 1. Goals** — *What does the client want to achieve, and Why?*
- 2. Facts** — *What do we know? What is given?*
- 3. Concepts** — *How does the client want to achieve the goals?*
- 4. Needs** — *How much money and space? What level of quality?*
- 5. Problem** — *What are the significant conditions affecting the design of the building? What are the general directions the design should take?*

1	4	3	2	5
2	3	4	1	5
4	1	2	3	5

Working through the steps in numerical sequence is preferable; theoretically, this is the logical order. But, in actual practice, **steps may be taken in a different order or at the same time**— all but the last step. It is frequently necessary, for example, to start with a given list of spaces and a budget (fourth step) before asking about Goals, Facts, and Concepts (first, second, and third steps).

The fifth step is taken only after marshalling all the previous information, extracting, abstracting, and getting to the very essence of the problem.

The Whole Problem



- It's important to search for and find the whole problem. To accomplish this, the problem must be identified in terms of **Function, Form, Economy, and Time**. Classifying information accordingly simplifies the problem while maintaining a comprehensive approach. A wide range of factors makes up the whole problem, but all can be classified in the four areas that serve later as design considerations.

Four Considerations

Function	1 People 2 Activities 3 Relationships
Form	4 Site 5 Environment 6 Quality
Economy	7 Initial budget 8 Operating costs 9 Life cycle costs
Time	10 Past 11 Present 12 Future

- **Function** implies ‘what’s going to happen in their building.’ It concerns activities, relationship of spaces, and people—their number and characteristics. Key words are (1) people, (2) activities, and (3) relationships.
- **Form** relates to the site, the physical environment (psychological, too) and the quality of space and construction. Form is what you will see and feel. It’s “what is there now” and “what will be there.” Key words are (4) site, (5) environment, and (6) quality.
- **Economy** concerns the initial budget and quality of construction, but also may include consideration of operating and life cycle costs. Key words are (7) initial budget, (8) operating costs, and (9) life cycle costs.
- **Time** has three classifications — past, present, and future — which deal with the influences of history, the inevitability of changes from the present, and projections into the future. Key words are (10) past, (11) present, and (12) future.

Framework

	1 Goals	2 Facts	3 Concepts	4 Needs	5 Problem
Function					
Form					
Economy					
Time					

- **All four considerations interact at each step.**

Information Index

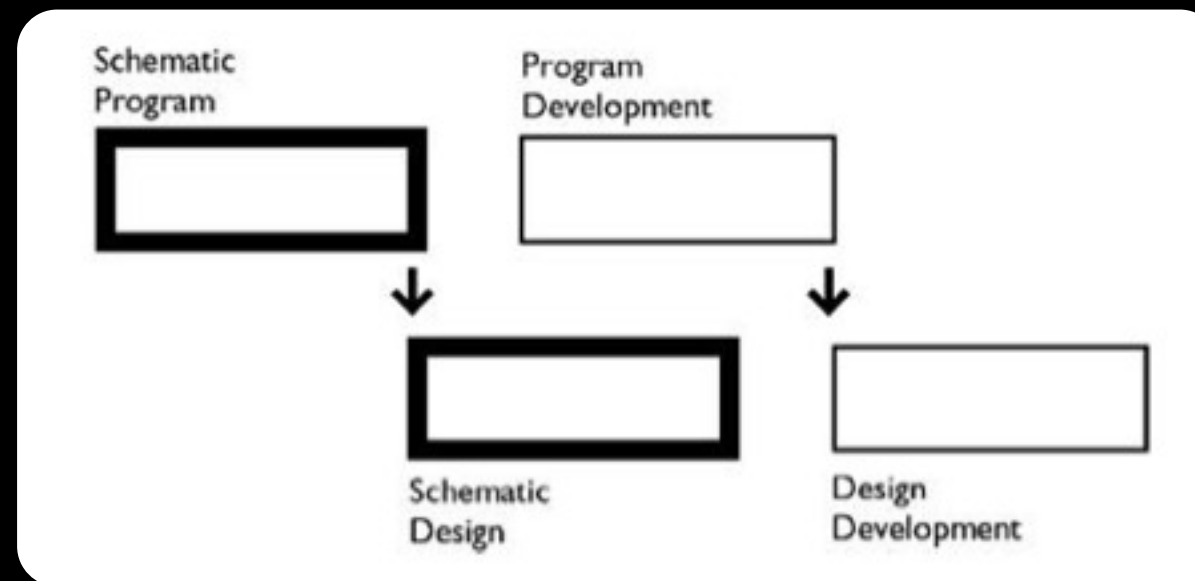
	1 Goals	2 Facts	3 Concepts	4 Needs	5 Problem
Function People Activities Relationships	Mission Maximum number Individual identity Hierarchy of values Security Progression Segregation Encounters Efficiency	Statistical data Area parameters Manpower/workloads User characteristics Community characteristics Value of loss Time-motion study Traffic analysis Behavioral patterns Space adequacy	Service grouping People grouping Activity grouping Priority Security controls Sequential flow Separated flow Mixed flow Relationship	Space requirements Parking requirements Outdoor space requirements Building efficiency Functional alternatives	Unique and important performance requirements which will shape building design
Form Site Environment Quality	Site elements(tree, water, open space, existing facilities, utilities) Efficient land use Neighbors Individuality Direction Entry Projected image Level of quality	Site analysis Climate analysis Code survey Soil analysis F.A.R. and G.A.C. Surroundings Psychological implications Cost/sq. ft. Building efficiency Functional support	Enhancement Climate control Safety Spacial foundations Density Interdependence Home base Orientation Accessibility Character Quality control	Quality(cost sq. ft.) Environmental and site influences on cost	Major form considerations which will affect building design
Economy Initial Budget Operating Costs Life Cycle Costs	Extent of funds Cost effectiveness Maximum return Return on investment Minimize operating costs Maintenance and operating costs	Cost parameters Maximum budget Time-use factors Market analysis Energy source-costs Activities and climate factors Economic data	Cost control Efficient allocation Multi-function Merchandising Energy conservation Cost control Cost control	Cost estimate analysis Entry budget(if required) Operating costs(if req.) Life cycle costs(if req.)	Attitude towards the initial budget and its influence on the fabric and geometry of the building
Time Past Present Future	Historic preservation Static/Dynamic Change Growth Occupancy date	Significance Space parameters Activities Projections Linear schedule	Adaptability Tailored/Loose fit Convertibility Expansibility Concurrent scheduling	Phasing Escalation	Implications of change/growth on long-range performance

Organizing Information

	1	2	3	4	5
Function	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Form	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Economy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Time	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

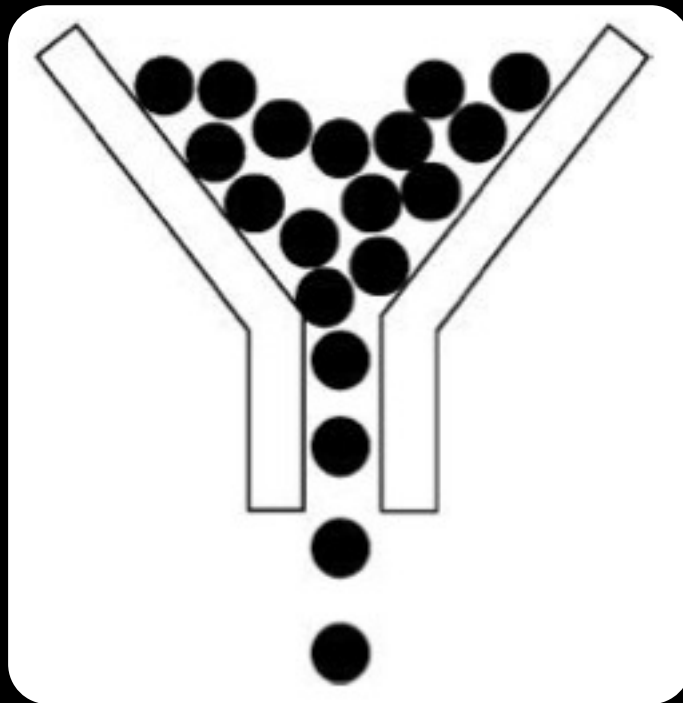
- **Programmers organize and classify information.**

Two-Phase Process



- **Programming is a two-phase process related to the two phases of design**— schematic design and design development.

Data Cog



- **One can assimilate any amount of information as long as it is pertinent, meaningful, and well organized for effective use.**

- Programming is five-step procedure(Goals, Facts, Concepts, Needs, Problem) involving four considerations(Function, Form, Economy, Time).
- Programming is a process leading to the statement of an architectural problem and the requirements to be met in offering a solution.
- Programming is the process of probing for sufficient information to understand and define the problem.
- Programming is problem seeking; design is problem solving.
- Programming is providing a sound basis for responsive design.
- Programming is analysis; design is synthesis.

- Programming is not design.
- Programming is distinct and separate from design.
- Programming is the prelude to good design.
- Programming is not merely asking questions.
- Programming is base on a combination of interviews for data gathering and work sessions for decision making.
- Programming is not an algorithmic process; it is a heuristic process.
- Programming is finding out what the whole problem is.
- Programming is determining present and future needs.

- Programming is the basis for a more comprehensive solution.
- Programming is a two-phase process.
- Programming is the establishment of limits and the scope of possibilities.
- Programming is processing raw data into useful and essential information.
- Programming is getting to the essence.
- Programming is the same process for any building type.
- Programming is essential regardless of size of project or the size of the firm.
- Programming is a cooperative process emphasizing client decision making.

- Programming is a rational and explicit process in which decisions and information are displayed for close scrutiny.
- Programming is an opportunity to raise the client's appreciation and aspiration for better buildings.
- Programming is a process requiring a high degree of communication.
- Programming is not just making a list of wanted spaces.
- Programming is the process of distinguishing between wants and needs.