
General Case Programming

Contributors: Robert H. Horner
Editors: Michel Hersen & Johan Rosqvist & Alan M. Gross & Ronald S. Drabman & George Sugai & Robert Horner
Description of the Strategy

Generalization is a core goal of nearly all instruction. When a student learns to read, calculate math equations, cross streets, or engage in social interactions in proscribed training contexts, it is expected that she or he will be able to perform those same skills in new contexts with new examples (e.g., new words, different numerical amounts, different streets, different greetings). Effective teaching should lead to skills that are performed with precision across the full range of trained and novel situations where the skill is appropriate. In effect, skills learned in a carefully defined instructional context are expected to generalize to the typical situations the student encounters on a daily basis. Generalization maximizes the value of good teaching and brings tremendous practical value to the acquisition of new skills. General case programming is an instructional approach used to build generalized skills.

General case programming combines (a) the social expectation that teaching should produce skills of direct, practical value; (b) the conceptual theory of behavior analysis; and (c) the instructional logic from direct instruction. The social values behind general case instruction emphasize development of practical skills that the student can use in typical situations to achieve socially valued outcomes. The theory of behavior analysis provides core concepts, such as stimulus control, stimulus generalization, reinforcement, and shaping, that are essential for organizing effective teaching, and the technology of direct instruction.

The major contribution of general case instruction for educators has been to focus attention on the impact of (a) selecting instructional examples and (b) sequencing instructional examples to encourage generalization. From behavior analysis, we learn how stimuli (e.g., letters on a page, social greetings from peers, lights on a street corner) can set the occasion for adaptive skills (e.g., reading, greeting, crossing). Further, we learn that with effective teaching, novel stimuli that are similar to those used during instruction (e.g., the different sounds of the letter b, different forms of greeting, different placements of street lighting) will also occasion appropriate responding.
Prior to the emergence of general case procedures, the approach to building generalized skills focused on simply adding new performance opportunities after initial skills were learned. The general case approach argues that generalization is most influenced by the initial design of instruction. Teaching generalized skills requires attention to instructional design decisions that are made before active teaching ever begins. Six key steps are emphasized in general case instruction.

Define the instructional universe. The instructional universe is the full set of situations (set of letters, set of streets, set of social situations) in which the new skills should be used. The instructional universe often is defined by curriculum demands or the unique learning needs of the student. The instructional universe for phonics-based reading for a third grader may be all third-grade reading materials in English that present any of the 44 primary letter (and combination) sounds regardless of font or context. The instructional universe for a student with severe intellectual disabilities who is working on street crossing may be all traffic intersections in town that have pedestrian walk lights. The value of defining the instructional universe is twofold. First, all members of the instructional universe must share the relevant stimulus features that signal that the new skill is appropriate. The members of an instructional universe will vary with respect to irrelevant features (e.g., size, color, position), but if they are members of the instructional universe, they will contain the relevant feature (e.g., the presence of a pedestrian walk light). Second, the designation of an instructional universe allows identification of situations in which the new skills should not occur (e.g., those situations outside the instructional universe), as well as those situations in which the new skill should occur. The acquisition of a “crossing the street” skill should be used at all controlled intersections but should not be used at freeway intersections. The acquisition of the very narrow skill of saying /b/ when presented with the letter b should not result in saying /b/ when presented with the letter d.

By defining the instructional universe and the relevant stimulus features that make situations or examples members of the instructional universe, the general case approach emphasizes that generalization goals are achieved by building very precise stimulus control. The student must learn not simply what skill to perform (e.g., saying /b/) but when that skill is appropriate (e.g., “b”) and not appropriate (e.g., “d”). The acquisition of generalized skills is not achieved by training loosely but by training with elegant precision.
The instructional universe is an essential concept for general case instruction. By defining the instructional universe, the teacher defines, up front, the targeted generalization range. The instructional universe should be clearly defined in any formal curriculum used to teach generalized skills, and the instructional universe should be overtly defined in any instructional individualized education program (IEP) goal.

Select teaching examples that sample the range of stimulus and response variation in the instructional universe. The second step in general case instruction involves the selection of teaching examples (e.g., what the student will be asked to do during instructional sessions). The two major messages from the general case approach are (a) always teach with multiple examples and (b) select examples to teach the student what not to do, as well as what to do. If the only goal is to document that a student is able to perform a skill, one teaching example might be acceptable, but if the goal is to demonstrate that a student both is able to perform a skill and performs that skill only under appropriate conditions, then multiple teaching examples are needed.

Initial teaching examples (e.g., words, math problems, streets, social greetings) will be selected from the instructional universe. The general guideline is to teach with a minimum of three examples from the instructional universe and select examples that are as different from each other as possible while still being part of the instructional universe. With some skills (e.g., math problems), the selection of many teaching examples is easy, and the number of teaching examples can be expanded from the minimum of three to many more. With other skills (e.g., street crossing), the expense and logistics of providing a teaching example make it important to select a smaller number of examples with care. The basic idea is to select a small number of teaching examples that represent the range of stimulus and response variation across the instructional universe.

Once a set of teaching examples has been selected from the instructional universe, it often is appropriate to select some nonexamples that are outside the instructional universe. Present nonexamples when it is as important to teach the student when or where not to perform the new skill as it is to teach him or her when or where it is appropriate to perform the skill. It would be recommended, for example, to select a range of situations in which social greetings are appropriate. A nonexample, however, may be when a student reencounters someone she or he just greeted. This would be
a situation outside the instructional universe and should not result in a student using a newly learned social greeting. Using a more academic example, calculation of the Pythagorean theorem is appropriate for any right triangle but would not be appropriate for use with an equilateral triangle. To teach this general case skill, the teacher would teach students the formal steps used in solving the Pythagorean theorem and would use multiple examples of right triangles that varied in size, orientation, and color. Then the teacher would contrast these examples with nonexamples (e.g., triangles of the same size, orientation, and color but without a right angle). The goal would be for the student to (a) learn the steps used in solving a problem with the Pythagorean theorem, (b) learn to use that skill with any right triangle, and (c) learn not to apply the skill with triangles that do not have a right angle.

Part of the power of general case programming is the development of the ability to discriminate between situations when it is appropriate and when it is not appropriate to perform a skill. The process of defining the instructional universe and selecting positive and negative teaching examples is at the heart of building generalized skills that are performed with precision.

Sequence the presentation of teaching examples to help the learner acquire key discriminations. There are two messages about the sequencing of teaching examples that characterize general case programming. The first focuses on building precise stimulus control by juxtaposing positive and negative teaching examples. If a student learns that “b” represents /b/, then juxtapose “d” and “l” as negative examples (e.g., “not /b/”) to help the student focus on the critical features of “b” that are relevant for effective reading.

The second sequencing message is to teach the general case, or most common, form of a skill to fluency before introducing exceptions. Many skills have a standard form but vary under certain situations. A common mantra learned to teach spelling in elementary school, “i before e except after c,” is an example. When exceptions are taught early in a program, the student may easily become confused. If, however, the general case is taught to fluency and then the exceptions are introduced, learning occurs more efficiently.
Teach the examples. The fourth step in general case programming involves the active instruction of a skill. Here, general case programming draws directly from applied behavior analysis and direct instruction guidelines. To the extent possible, teaching should be done in the context in which the skill will be performed. Appropriate pacing of instructional trials, immediate feedback, reinforcement of correct responding, fading of instructional prompts, and shaping of complex skills are incorporated into the general case approach. The general case approach offers guidance for determining what will be presented during teaching sessions and incorporates existing strategies for actively teaching selected examples.

Test with novel probe examples. The final step in general case programming is to test intermittently with novel positive and negative examples. It is not adequate to simply meet criteria with the teaching examples. Successful instruction occurs when the student (a) performs the new skill with technical precision, (b) performs the skill in novel appropriate situations, and (c) does not perform the skill in inappropriate situations. General case programming has been successful when a student documents that she or he is able to perform the targeted skill with precision, performs the skill across appropriate trained and novel examples and situations, and does not perform the skill in inappropriate situations.

Research Basis

General case programming evolved from the behavioral principles of applied behavior analysis and the teaching practices of direct instruction. A strong empirical foundation exists in applied behavior analysis documenting the process by which stimulus control develops and the link between stimulus similarity and generalization of acquired skills. This work was done initially with rats and pigeons and then confirmed with human learners. Additional research from applied behavior analysis and direct instruction provides strong support for the use of instructional procedures emphasizing (a) multiple presentations of instructional opportunities, (b) instructional prompts that guide the student’s attention to relevant stimulus features, (c) contingent rewards, (d) shaping of a response to achieve precision, and (e) fading of prompts to ensure appropriate stimulus control.
During the last 20 years, general case programming has become a curriculum design format within the direct instruction approach. The specific emphases on defining an instructional universe and using general case guidelines for selecting and sequencing teaching examples have been validated in multiple research studies. Academic curricula addressing reading, writing, and math in particular have documented the value of the general case procedures. General case programming also has been applied to social skills and self-help skills with students who have severe disabilities. General case programming has proven superior to other instructional methods for teaching skills such as table bussing, grocery store shopping, street crossing, vending machine use, social greetings, and personal hygiene to students with more significant intellectual disabilities.

**Relevant Population and Cautions**

General case programming represents a theory-driven approach to teaching skills that will be performed across the full range of practical situations. The research support for this approach is strongest for the development of academic skills in elementary school and the development of self-help, vocational, and community integration skills by individuals with severe disabilities. Current research efforts are applying general case programming to the development of early language with language-delayed young children and the development of more complex social skills with students in elementary and middle school who carry emotional or behavioral disorder and severely emotionally disturbed labels.

It is important, however, to note that the general case approach addresses most directly the phenomenon of stimulus generalization. There remain patterns of learning related to response generalization and the emergence of stimulus equivalence skills that are not predicted by the general case programming approach. The integration of the different generalization phenomena will be an important area for future study.

**Case Illustration**

The benefits of general case programming were demonstrated by Jeff Sprague and his colleagues when they tested the comparative effects of using (a) a single, excellent
teaching example; (b) multiple, similar teaching examples; and (c) the general case guidelines for selecting multiple examples that sampled the range of variability in the instructional universe. The six participating students were high school students with moderate to severe intellectual disabilities, and the task was to use quarters to purchase items from vending machines. Sprague identified 10 vending machines that sampled the range of stimulus and response variability for machines in his town (e.g., different levers and buttons, different amounts of money). Sprague first presented each student with the opportunity to purchase desired items from each of the 10 machines and demonstrated that none of the students were competent vending machine users. These 10 machines were then kept only to probe for generalization and were never presented in training. Sprague next provided training for each of the students with one vending machine, tested to see the effect, then provided training with either three similar machines or three general case machines (machines that were maximally different but part of the same instructional universe). After learning to operate one training machine, the six students were more likely to perform correctly with the one probe machine that was similar to the one they were trained with. Unfortunately, even though the students had mastered the skill of purchasing with the one training machine, they were not successful with the range of probe machines. Similarly, when students 4, 5, and 6 were taught to use the three similar machines, they used their new skill with precision on the one or two probe machines that were very similar but were unsuccessful when the levers and buttons, the amount of money, or the mode of obtaining the item varied. Only after training with a set of multiple teaching examples that sampled the range of relevant stimulus and response variation (e.g., use of general case training examples) did the students perform successfully across the full range of probe examples. Most important, only after the general case training did the use of vending machines become a regular part of each student's daily activities.

Robert H. Horner

http://dx.doi.org/10.4135/9781412950534.n3071

See also

- Applied Behavior Analysis (Vol. III)
- Discrimination Training (Vols. II & III)
- Fading (Vol. III)
- Generalization (Vols. I & III)
- Phases of Learning (Vol. III)
- Stimulus (Vol. II)
- Stimulus Control (Vols. II & III)
- Transfer of Stimulus Control (Vol. II)

Suggested Readings


General case programming is defined as using multiple training examples (stimulus exemplars) that sample the range of relevant stimulus situations and response variations (i.e. subjects with mental retardation who were trained to operate a variety of washing machines were more successful in operating a novel machine than were the subjects who received training on only one machine). There was more generalization of the skills with the use of general case programming. Upper_case_snake_case. Which case type should I use? What convention should I use when naming files? Go Conventions. A major factor of being successful in naming is knowing the type of case you want to use so that you can have a consistent convention per project/workspace. If you're writing software, you'll come across at least one of these in a languages specification for how it's written. Some languages (Go, particularly) rely heavily on you knowing the difference between two of them and using them correctly! In this tutorial, you will learn to create a switch statement in C programming with the help of an example. The switch statement allows us to execute one code block among many alternatives. You can do the same thing with the if...else...if ladder. However, the syntax of the switch statement is much easier to read and write. Syntax of switch...case. switch (expression) { case constant1: // statements. break; case constant2: // statements. break; . . . default: // default statements }. How does the switch statement work?