

# Garbage In–Garbage Out

The quality of system output is dependent on the quality of system input.<sup>1</sup>

The garbage in–garbage out principle is based on the observation that good inputs generally result in good outputs, and bad inputs, barring design intervention, generally result in bad outputs. The rule has been generalized over time to apply to all systems, and is commonly invoked in domains such as business, education, nutrition, and engineering, to name a few. The “garbage in” metaphor refers to one of two kinds of input problems: problems of type and problems of quality.<sup>2</sup>

*Problems of type* occur when the incorrect type of input is fed into a system, such as entering a phone number into a credit card number field. Problems of type are serious because the input provided could be radically different from the input expected. This can be advantageous in that problems of type are relatively easy to detect, but problematic in that they represent the maximum form of garbage if undetected. Problems of type are generally caused by a class of errors called mistakes—incorrect actions caused by conscious actions. The primary strategies for minimizing problems of type are affordances and constraints. These strategies structure input and minimize the frequency and magnitude of garbage input.

*Problems of quality* occur when the correct type of input is fed into a system, but with defects, such as entering a phone number into a phone number field but entering the wrong number. Depending on the frequency and severity of these defects, problems of quality may or may not be serious. Mistyping one letter in a name may have minor consequences (e.g., search item not found); trying to request a download of fifty records but typing five thousand might lock up the system. Problems of quality are generally caused by a class of errors called slips—incorrect actions caused by unconscious, accidental actions. The primary strategies for minimizing problems of quality are previews and confirmations. These strategies allow the consequences of actions to be reviewed and verified prior to input.

The best way to avoid garbage out is to prevent garbage in. Use affordances and constraints to minimize problems of type. Use previews and confirmations to minimize problems of quality. When input integrity is critical, use validation tests to check integrity prior to input, and consider confirmation steps that require the independent verification of multiple people. Consider mechanisms to automatically flag and, in some cases, autocorrect bad input (e.g., automatic spelling correction in word processors).

See also Errors, Feedback Loop, and Signal-to-Noise Ratio.

<sup>1</sup> Also known as *GIGO*.

<sup>2</sup> While the garbage in–garbage out concept dates back to Charles Babbage (1864), or earlier, the term is attributed to George Fuechsel, a programming instructor who used it as a teaching device in the late 1950s. It should be noted that Fuechsel used the principle to emphasize that “garbage out” is not the inevitable result of “garbage in,” but rather a condition that should be addressed through design.

## Original Form

**Order Form: Billing and Shipping Information** page 2 of 2

**Shipping Address:**

Name

Street Address

Street Address

City, State and Zip Code

**Billing Address:**

Name

Street Address

Street Address

City, State and Zip Code

**Credit Card Information:**

Name on Credit Card  Type of Credit Card  Credit Card Number  Exp. Date

**Shipping Method:**

**Date to Ship:**

○ Unconstrained fields increase the probability of garbage input.

## Redesigned Form

**Order Form: Billing and Shipping Information** page 2 of 2

☐ click here to use the information saved with your account

**Shipping Address:**

First Name  Last Name

Street Address

City  State  Zip Code

**Billing Address**

☐ click here if Billing Address is the same as Shipping Address

First Name  Last Name

Street Address

City  State  Zip Code

**Credit Card Information:**

Name on Card  Type of Card

Credit Card Number

Expiration Date

**Shipping Method:**

Standard Shipping \$7.00

**Date to Ship:**

Month  Day  Year

○ Allow users to automate input by accessing stored information.

○ Constrain input when a specific amount of information is required.

○ Constrain input using menus of options.

Your order will not be placed until you review the information you entered and click the "submit order" button.

March 21, 2003

1 dozen  
chocolate chip cookies

Ship to:  
Randy Williams  
101 Main Street  
Houston, TX 90990

Ship on:  
March 30, 2003

Bill to:  
Kristen Johnson  
211 Elm Blvd.  
Columbus, OH 44356

VISA: \*\*\*\* \* 3041

Exp. Date 5/2006

Name on Card: Kristen J. Johnson

○ Allow users to preview information before they complete transactions.

These two different designs of the same order form illustrate how designers can influence the amount of garbage entered into a system.