

Value Parameter Encyclopedia Article

Value Parameter

The following sections of this BookRags Literature Study Guide is offprint from Gale's For Students Series: Presenting Analysis, Context, and Criticism on Commonly Studied Works: Introduction, Author Biography, Plot Summary, Characters, Themes, Style, Historical Context, Critical Overview, Criticism and Critical Essays, Media Adaptations, Topics for Further Study, Compare & Contrast, What Do I Read Next?, For Further Study, and Sources.

(c)1998-2002; (c)2002 by Gale. Gale is an imprint of The Gale Group, Inc., a division of Thomson Learning, Inc. Gale and Design and Thomson Learning are trademarks used herein under license.

The following sections, if they exist, are offprint from Beacham's Encyclopedia of Popular Fiction: "Social Concerns", "Thematic Overview", "Techniques", "Literary Precedents", "Key Questions", "Related Titles", "Adaptations", "Related Web Sites". (c)1994-2005, by Walton Beacham.

The following sections, if they exist, are offprint from Beacham's Guide to Literature for Young Adults: "About the Author", "Overview", "Setting", "Literary Qualities", "Social Sensitivity", "Topics for Discussion", "Ideas for Reports and Papers". (c)1994-2005, by Walton Beacham.

All other sections in this Literature Study Guide are owned and copyrighted by BookRags, Inc.

Contents

Value Parameter Encyclopedia Article.....	1
Contents.....	2
Value Parameter.....	3



Value Parameter

A **value parameter** is a **variable** that is passed to a **function** when it is called. Value parameters are also called actual parameters. Given the function **call**:

- `pow(i, j);`

`i` and `j` are both value parameters. The values of `i` and `j` at the time this call is made to the `pow()` function are the values passed into the formal parameters of the function. The formal parameters are the variables used in the definition of the function. Given the function definition:

- `int pow(int base, int exp) {...}`

`base` and `exp` are the formal parameters of the function. The type of any value parameter needs to be the same type as the formal parameter, or a subtype of the formal parameter's type. The order in which value parameters are passed to a function call will correspond to the order in which the formal parameters will be assigned values when the function is evaluated. The function's definition determines its formal parameters. However, every time a function is called, it can have different value parameters.