Invariant and Variant Assertions Encyclopedia Article

Invariant and Variant Assertions

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.



BOOKRAGS

Contents

Invariant and Variant Assertions Encyclopedia Article.....1

Contents.....2

Invariant and Variant Assertions.....3

Invariant and Variant Assertions

An assertion is a statement about a part of a program that is either true or false. Two types of **assertions**, invariant and variant, function in the **verification** process of **programming**. Verification consists of proving logically that the programming sequence will lead to the desired objective, no matter what the **data** are, or how many programming steps are necessary.

The verification process involves a mathematical property called induction. Induction asserts, or assumes to be true, that the state after each programming step is suitable for the start of the next step. The assertion that the state remains suitable from step to step is the invariant assertion. Maintaining suitable states does not guarantee the satisfactory achieving of the programming objective. It must also be demonstrated that each programming step incrementally advances towards the **object**, and that only a finite number of increments are possible. Otherwise, a program could run indefinitely without giving an answer.

In contrast to an invariant assertion, a variant assertion is a statement that can start out as true or false but will be changed to the opposite **value** by the time the programming sequence ends. A variant assertion is more complex than an invariant assertion, and deals with the programming logic that ensures that a programming sequence will eventually end. The variant statement is typically used when the sequence of **commands** does not repeat, or iterate.

Usually the true form of an assertion, whether invariant or variant, is used in programming verification.