Exception Handling Encyclopedia Article

Exception Handling

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

Exception Handling Encyclopedia Article.....1

Contents.....2

Exception Handling.....3

Exception Handling

Despite a programmer's best efforts, abnormal or illegal operations happen during the execution of program instructions. These deviations from the normal execution sequence are called exceptions. Processors detect exceptions when activities such as division by zero (impossible) or a request for **memory** allocation exceeds the available amount of storage occur. Exceptions can corrupt memory and create adverse side-effects if the program continues. The process of dealing with exceptions is called exception handling.

There is a wide variety of ways in which **programming** languages implement exception handling. Once an exception has been detected, it is *raised*. An exception is raised when an abnormal event occurs, information about that event is collected, and control is transferred to an exception handler. An exception handler performs a specified response to recover from the exception. The specified response can be to execute a corrective **action** or to ignore the exception.

Once the exception handler's action is executed, the program can resume normal processing, or it can terminate. Resumption of a process may require a statement to be reevaluated or it may return control to the point after which the exception occurred. Termination may involve the cessation of a portion of a program or the program in its entirety. If an exception handler does not exist, the run-time system can execute a default action, such as terminating the program.

Exception handling is used to improve the reliability of a program and make it robust. Robust programs handle exceptions gracefully, without undo interruption or bringing frustration upon the user. Exception handling can also be used to validate **data** and maintain the integrity of data or the environment.