

Chapter 14 Topics

- Introduction to Exception Handling
- Exception Handling in Ada
- Exception Handling in C++
- Exception Handling in Java
- Introduction to Event Handling
- Event Handling with Java

Copyright © 2008 Pearson Education, Inc. All rights reserved. 14-1

Introduction to Exception Handling

- In a language without exception handling: When an exception occurs, control goes to the operating system, where a message is displayed and the program is terminated
- In a language with exception handling: Programs are allowed to trap some exceptions, thereby providing the possibility of fixing the problem and continuing
- Many languages allow programs to trap input/output errors (including EOF)
- Def: An exception is any unusual event, either erroneous or not, detectable by either hardware or software, that may require special processing
- Def: The special processing that may be required after detection of an exception is called exception handling
- Def: The exception handling code unit is called an exception handler

Copyright © 2008 Pearson Education, Inc. All rights reserved. 14-2

Introduction to Exception Handling

- Def: An exception is raised when its associated event occurs
- A language that does not have exception handling capabilities can still define, detect, raise, and handle exceptions (user defined, software detected)
 1. Send an auxiliary parameter or use the return value to indicate the return status of a subprogram
 2. Pass a label parameter to all subprograms (error return is to the passed label)
 3. Pass an exception handling subprogram to all subprograms
- Advantages of Built-in Exception Handling:
 1. Error detection code is tedious to write and it clutters the program
 2. Exception propagation allows a high level of reuse of exception handling code

Copyright © 2008 Pearson Education, Inc. All rights reserved. 14-3

Introduction to Exception Handling

- Design Issues for Exception Handling:
 1. How and where are exception handlers specified and what is their scope?
 2. How is an exception occurrence bound to an exception handler?
 3. Where does execution continue, if at all, after an exception handler completes its execution?
 4. How are user-defined exceptions specified?
 5. Should there be default exception handlers for programs that do not provide their own?
 6. Can built-in exceptions be explicitly raised?
 7. Are hardware-detectable errors treated as exceptions that can be handled?
 8. Are there any built-in exceptions?
 9. How can exceptions be disabled, if at all?

Copyright © 2000 Pearson Education, Inc. All rights reserved. 11.1

Exception-Handling Control Flow

Copyright © 2000 Pearson Education, Inc. All rights reserved. 11.2

Exception Handling in Ada

- An exception handler in Ada can occur in either a subprogram body, a package body, a task, or a block
- Because exception handlers are usually local to the code in which the exception can be raised, they do not have parameters

Copyright © 2000 Pearson Education, Inc. All rights reserved. 11.3

Exception Handling in Ada

- Handler form:

```
exception
  when exception_name[|exception_name] =>
    statement_sequence
  ...
  when ...
  ...
  [when others =>
    statement_sequence]
```
- Handlers are placed at the end of the block or unit in which they occur

Copyright © 2000 Pearson Education, Inc. All rights reserved. 11.7

Exception Handling in Ada

- Binding Exceptions to Handlers
 - If the block or unit in which an exception is raised does not have a handler for that exception, the exception is propagated elsewhere to be handled
 - 1. Procedures - propagate it to the caller
 - 2. Blocks - propagate it to the scope in which it appears
 - 3. Package body - propagate it to the declaration part of the unit that declared the package (if it is a library unit (no static parent), the program is terminated)
 - 4. Task - no propagation; if it has a handler, execute it; in either case, mark it "completed"

Copyright © 2000 Pearson Education, Inc. All rights reserved. 11.8

Exception Handling in Ada

- Continuation
 - The block or unit that raises an exception but does not handle it is always terminated (also any block or unit to which it is propagated that does not handle it)

Copyright © 2000 Pearson Education, Inc. All rights reserved. 11.9
