

Exception:

Any unusual event, erroneous or not, that is detectable either by hardware or software that may require special processing.

Exception Handling:

The special processing required by the detection of an exception.

Exception Handler:

The code unit that performs the special processing.

Exception Handling

Error

- hardware
- software: subscript range error, divide by zero
overflow, underflow

Non-error

- end-of-file

Exception Handler

- built-in
- user-defined

PL/I

22 built-in exceptions

ON condition [SNAP]

BEGIN;

;

END;

CLU, C++, Java

Design Issues for Exception Handlers

1. How (program unit or a code segment) and where are exception handlers specified, and what is their scope?
2. How is the 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. How can exceptions be disabled, if at all?

```

GRADE_DISTRIBUTION: PROCEDURE OPTIONS (MAIN);
  DECLARE FREQ(1:10) FIXED INIT ((10) 0),
          NEW_GRADE FIXED,
          LIMIT_1 FIXED,
          LIMIT_2 FIXED,
          INDEX FIXED;
/* Exception Handlers */
  ON ENDFILE (SYSIN) GOTO FINISH;
  ON SUBSCRIPTRANGE
    BEGIN;
    IF NEW_GRADE = 100 THEN
      FREQ(10) = FREQ(10) + 1;
    ELSE
      DO;
      PUT LIST ('INPUT GRADE: ' || NEW_GRADE ||
              'NOT IN RANGE') SKIP;
      GOTO INPUT_LOOP;
    END;
  END;
/* Main program body */
INPUT_LOOP:
  DO;
  GET LIST (NEW_GRADE);
  INDEX = NEW_GRADE / 10 + 1;
(SUBSCRIPTRANGE):
  FREQ(INDEX) = FREQ(INDEX) + 1;
  END INPUT_LOOP;

FINISH:
  PUT LIST ('LIMITS  FREQUENCY') SKIP(2);
  DO INDEX = 0 TO 9;
    LIMIT_1 = 10 * INDEX;
    LIMIT_2 = LIMIT_1 + 9;
    IF INDEX = 9 THEN
      LIMIT_2 = 100;
    PUT LIST (LIMIT_1, LIMIT_2, FREQ(INDEX+1));
  END;
END GRADE_DISTRIBUTION;

```

Event Handler

Exception vs. Event

Exception – internal to the program in which it occurs

Event – external

Event – mouse click, keyboard stroke, data from sensor, etc.

Event handler

- code that is executed in response to an event.
- “Listener” in Java

-

```
while (1) {  
    if (!empty(event queue)) {  
        pop an event off the event queue;  
        pass the event to appropriate handler;  
    }  
}
```