Py §14.5, M2 §10.8-10.13: Exceptions

5 Nov 2008
CMPT14x
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Quiz06: ch8-9 (10 mins, 20 pts)

- Define: bit, byte, nibble, word.
- Describe: cylinder, head, sector.
- Name and describe the four set theory operators we learned.
- A = {1, 3, 5}, B = {2, 3, 4}.
  (set theory, not Python)
  - **A** ∩ **B** = ?
  - **A** ∪ **B** = ?
- Let three permission flags be: r=4, w=2, x=1.
  - What bitset value corresponds to a file with both read (r) and execute (x) permission?

Quiz06: ch8-9 answers #1

- Define: bit, byte, nibble, word.
  - Bit: smallest unit of information, only two possible values: 0/1, true/false, high/low, charge/uncharge
  - Byte: 8 bits. Can represent one ASCII character.
  - Nibble: 4 bits. One hexadecimal digit.
  - Word: Unit of data processed by CPU. Most CPUs have 32-bit or 64-bit words.

Quiz06: ch8-9 answers #2

- Describe: cylinder, head, sector.
  - Cylinder: concentric tracks across all heads in a hard disk
  - Head: read/write head on a hard disk. Number of heads is number of useable surfaces, usually twice the number of platters
  - Sector: portion of surface under the head for a fixed rotational angle of the platter

Quiz06: ch8-9 answers #3-4

- Name and describe the four set theory operators we learned.
  - Union: everything in either A or B
  - Intersection: everything in both A and B
  - Set difference: everything in A but not in B
  - Symmetric set difference: in exactly one of A or B, not both
- A = {1, 3, 5}, B = {2, 3, 4}. (set theory, not Python)
  - **A** ∩ **B** = {3}
  - **A** ∪ **B** = {1, 2, 3, 4, 5}

Quiz06: ch8-9 answers #5

- Let three file permission flags be: r = 4, w = 2, x = 1.
  - What is the bitset value corresponding to a file with both read (r) and execute (x) permission?
  - 4 + 1 = 5

Options for error handling

- Use a combination of these:
  - Ask the user to be nice:
    - User manual, precondition comments, prompts
  - Print an error message to screen
  - Set a result flag:
    - e.g., return False upon error
  - Panic and die: sys.exit()
  - Raise an exception: ZeroDivisionError

Exceptions

- Exceptions are a way of terminating execution of the current context
- When an exception is raised (thrown),
  - execution of the current procedure stops, and
  - Control jumps to the nearest exception handler (catches the exception)
- The exception handler can cleanup
- Execution then continues after that block
- If the exception reaches outermost level, an error message is automatically generated

try / except

- If an exception is raised within a try block,
- Execution of the block terminates and control jumps to the except clause:

```
try:
    while True:
        numer = input('Numerator: ')
        denom = input('Denominator: ')
        print '%d / %d = %d' % (numer, denom, numer /
          denom)
except:
    print 'Oops!'
```

Catching specific exceptions

- We can opt to catch only specific exceptions:

```
try:
    while True:
        numer = input('Numerator: ')
        denom = input('Denominator: ')
        print '%d / %d = %d' % (numer, denom, numer /
          denom)
except ZeroDivisionError:
    print 'Oops! Divide by zero!'
```

- Any other exception falls through to the next exception handler