# Exceptions Question Bank

CSCI 40: Computing for the Web

**How current LLMs scored on this practice bank** *2026-08-26*

| Model | Reasoning | Score |
|---|---|---|
| claude-haiku-4.5 | low effort | 35/36 (97%) |
| claude-sonnet-5 | low effort | 34/36 (94%) |
| claude-fable-5 | low effort | 35/36 (97%) |
| claude-opus-4.8 | low effort | 36/36 (100%) |
| gpt-5.6-luna | low reasoning | 33/36 (92%) |
| gpt-5.4 | low reasoning | 36/36 (100%) |
| gpt-5.6-terra | low reasoning | 32/36 (89%) |
| gpt-5.5 | low reasoning | 35/36 (97%) |
| gpt-5.6-sol | low reasoning | 36/36 (100%) |
| gemini-3.1-flash-lite | minimal thinking | 26/36 (72%) |
| gemini-3.5-flash-lite | minimal thinking | 32/36 (89%) |
| gemini-3.6-flash | minimal thinking | 34/36 (94%) |
| gemini-3.7-flash | low thinking | 36/36 (100%) |

One independent, tool-free prompt per question; deterministic executable ground truth.

Your exceptions quiz will contain 5 questions that follow the pattern in this question bank.

Recall that you are responsible for knowing the following exceptions: AssertionError, AttributeError, IndexError , KeyError, NameError, UnboundLocalError, TypeError, ZeroDivisionError.

**Problem 1.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
total = 0
while xs:
    total += 1
print('total=', total)
```

**Problem 2.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
while xs:
    total += 1
    assert(xs)
print('total=', total)
```

**Problem 3.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
while len(xs)<5:
    xs.append('test')
print('len(xs)=',len(xs))
```

**Problem 4.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = []
while len(xs)<5:
    xs.append('test')
assert(xs)
print('len(xs)=',len(xs))
```

**Problem 5.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
while xs:
    xs.append('test')
print('len(xs)=',len(xs))
```

**Problem 6.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
total = 0
while xs:
    total //= total
print('len(xs)=',len(xs))
```

**Problem 7.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
while xs:
    xs.append('test')
    assert('t' in xs)
print('len(xs)=',len(xs))
```

**Problem 8.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
while xs:
    xs += 'test'
print('len(xs)=',len(xs))
```

**Problem 9.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
total = 10
for x in xs:
    total %= x
print("total=",total)
```

**Problem 10.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [0, 1, 2]
xs.replace(1, 2)
print("xs=", xs)
```

**Problem 11.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
s = 'hello world'
i = s.find('-')
print("i=", i)
```

**Problem 12.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(v['hw1'])
```

**Problem 13.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(k['hw1'])
```

**Problem 14.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(v[0])
```

**Problem 15.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
for k,v in sorted(grades.items()):
    print(k[0])
```

**Problem 16.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = "grade=" + grades['alice']['hw1']
print('output=', output)
```

**Problem 17.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = "grade=" + grades['charlie']['hw1']
print('output=', output)
```

**Problem 18.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = "grade=" + grades['bob']['hw2'][91]
print('output=', output)
```

**Problem 19.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
output = alice['hw1']
print('output=', output)
```

**Problem 20.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
total = 0
for i in grades:
    for j in i:
        total += 1
print('total=',total)
```

**Problem 21.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
try:
    result = xs[3]
except IndexError:
    result = -1
print('result=', result)
```

**Problem 22.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
try:
    result = xs[3]
except:
    result = -1
print('result=', result)
```

**Problem 23.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
xs = [1, 2, 3]
try:
    result = xs[3]
except NameError:
    result = -1
print('result=', result)
```

**Problem 24.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
try:
    output = "grade=" + grades['charlie']['hw1']
except KeyError:
    output = 'oops'
print('output=', output)
```

**Problem 25.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
grades={
    'alice':{'hw1':99,'hw2':88},
    'bob':{'hw1':82,'hw2':91},
}
try:
    output = "grade=" + grades['charlie']['hw1']
except IndexError:
    output = 'oops'
print('output=', output)
```

**Problem 26.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)
foo()
```

**Problem 27.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)

try:
    foo([1,2,3])
except AssertionError:
    pass
```

**Problem 28.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)

result = 0
try:
    result += foo([1,2,3])
except AssertionError:
    result -= 1
print('result=', result)
```

**Problem 29.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)

example = 0
try:
    example += foo([1,2,3])
    example += 1
except ValueError:
    pass
print('example=', example)
```

**Problem 30.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def foo(xs):
    assert(len(xs) > 0)

example = 0
try:
    example += foo([1,2,3])
    example += 1
except AssertionError:
    pass
print('example=', example)
```

**Problem 31.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    assert(len(xs) > 0)
    return len(xs)*2

result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar([])
    result += bar([5])
except AssertionError:
    pass

print('result=', result)
```

**Problem 32.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    assert(len(xs) > 0)
    return len(xs)*2

result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar([])
    result += bar([5])
except ValueError:
    pass

print('result=', result)
```

**Problem 33.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    assert(len(xs) > 0)
    return len(xs)*2

result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar()
    result += bar([5])
except AssertionError:
    result += 1
except TypeError:
    result += 5

print('result=', result)
```

**Problem 34.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    if not len(xs) > 0:
        raise ValueError('input list must be non-empty')
    return len(xs)*2

result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar()
    result += bar([5])
except AssertionError:
    result += 1
except TypeError:
    result += 5

print('result=', result)
```

**Problem 35.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    if not len(xs) > 0:
        raise ValueError('input list must be non-empty')
    return len(xs)*2

result = 0
try:
    result += bar([1,2,3])
    result += bar([2,3])
    result += bar()
    result += bar([5])
except ValueError:
    result += 1
except TypeError:
    result += 5

print('result=', result)
```

**Problem 36.** The following code (circle one)

terminates without error  throws an exception  runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
def bar(xs):
    if not len(xs) > 0:
        raise ValueError('input list must be non-empty')
    return len(xs)*2

result = 0
result += bar

print('result=', result)
```