

Common Logic Mistakes

Yung-Hsiang Lu

Tom always eats pizza for lunch on Sundays. Which statement must be correct? You may choose multiple answers.

- A) Tom never eats pizza for dinner on Sundays.
- B) If Tom does not eat pizza for dinner, today must not be Sunday.
- C) If Tom does not eat pizza for lunch, today must not be Sunday.
- D) Tom never eats pizza for lunch on Monday.
- E) If Tom eats pizza for lunch today, he does not eat pizza for dinner today.
- F) When Tom eats pizza, he always drinks Pepsi.

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

If it is raining right now, the street is wet. Which statement must be correct? You can choose multiple answers.

- A) If the street is wet, it must be raining right now.
- B) If the street is dry, it must not be raining right now.
- C) If it is not raining right now, the street must be dry.
- D) If the street is dry, it must be sunny right now.

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

Logic

Your Score	{score}
Max Score	{max-score}
Number of Quiz Attempts	{total-attempts}

Question Feedback/Review Information Will Appear Here

Continue

Review Quiz

Tom always eats pizza for lunch on Sundays. Which statement must be correct?

Tom never eats pizza for dinner on Sundays. (We don't know.)

If Tom does not eat pizza for dinner, today must not be Sunday.
(We don't know.)

Tom never eats pizza for lunch on Monday. (We don't know.)

If Tom eats pizza for lunch today, he does not eat pizza for dinner today. (We don't know.)

When Tom eats pizza, he always drinks Pepsi. (We don't know.)

If Tom does not eat pizza for lunch, today must not be Sunday. (✓ correct)

If it is raining right now, the street is wet. Which statement must be correct?

If the street is wet, it must be raining right now. (It could be raining ten minutes ago. The street could be flooded. Maybe a water pipe broke.)

If it is not raining right now, the street must be dry. (The street could be flooded. Maybe a water pipe broke.)

If the street is dry, it must be sunny right now. (It may be cloudy.)

If the street is dry, it must not be raining right now. (✓ correct)

```
int x;  
int y;  
/* some code */  
...  
/* L1 */  
if (x == 0)  
{  
    y = 0;  
}  
/* L2 */
```

If x is not zero ($x \neq 0$) at L1,
what is the value of y at L2?
Can y be zero at L2?

```
int x;
int y;
/* some code */
...
/* L1 */
if (x == 0)
{
    y = 0;
}
/* L2 */
```

If x is zero ($x == 0$) at L1, y is zero at L2.
If it is raining, the street is wet.

If it is not raining, we do not know whether the street is wet or dry.

If x is not zero ($x != 0$) at L1, what is the value of y at L2?

Can y be zero at L2?

We don't know; y may already be zero at L1.

```
int x;  
int y;  
/* some code */  
...  
/* L1 */  
if (x == 0)  
{  
    y = 0;  
}  
/* L2 */
```

If y is zero at L2, what is x's
value at L1?
Must x be zero?

```
int x;
int y;
/* some code */
...
/* L1 */
if (x == 0)
{
    y = 0;
}
/* L2 */
```

If x is zero ($x == 0$) at L1, y is zero at L2.
If it is raining, the street is wet.

If the street is wet, is it raining?
No. Maybe a water pipe broke.

If y is zero at L2, what is x's value
at L1?

Must x be zero?

**We don't know; y
may already be
zero at L1.**

```
int x;  
int y;  
/* some code */  
...  
/* L1 */  
if (x == 0)  
{  
    y = 0;  
}  
/* L2 */
```

If y is not zero at L2, what is x's
value at L1?

Can x be zero?

```
int x;  
int y;  
/* some code */  
  
...  
/* L1 */  
if (x == 0)  
{  
    y = 0;  
}  
/* L2 */
```

If x is zero ($x == 0$) at L1, y is zero at L2.
If it is raining, the street is wet.

If the street is not wet, is it raining?

If y is not zero at L2, what is x's value at L1?

Can x be zero?

x must not be zero at L1.

brace { } after if

You should **always** add braces after if (or else or while)

```
int x = 0;
int y = 0;
int z = 0;
if (x == 1)
    y = 1;
    z = 1;

/* what are the values of x, y,
   z here? */
```

It is equivalent to

```
int x = 0;
int y = 0;
int z = 0;
if (x == 1)
{
    y = 1;
}
z = 1; /* not controlled by x */
x is 0, y is 0, z is 1 (not 0)
```

if - else pairs

else always follows the **closest** if

```
int x = 0;
int y = 0;
int z = 0;
if (x == 0)
    if (y == 1)
        z = 1;
else
    z = 2;
```

```
if (x == 0) /* true */
{
    if (y == 1) /* false */
    {
        z = 1;
    }
} Wrong
else
{
    z = 2;
}
/* z is 0 */
```

```
if (x == 0) /* true */
{
    if (y == 1) /* false */
    {
        z = 1;
    }
else
{
    z = 2;
}
}
/* z is 2 */
```

no default in switch

If you think the cases cover all possible scenarios, **print an error message** in default.

The “impossible” occurs frequently.