



Certified Uniswap Developer (CUND)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working Uniswap developer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: Uniswap and AMM Foundations

DATA

Emma Wilson maps each AMM term to its meaning, with one meaning missing.

Term	Meaning
Constant product	x times y equals k
Liquidity pool	(to choose)
LP token	Share of a pool
Swap fee	Charge per trade

Table 1.1 AMM terms and meanings.

What is a liquidity pool?

- ☐ A Reserves in a pool
- ☐ B A single trade order
- ☐ C An order book queue
- ☐ D A price oracle feed

Correct answer: A

A liquidity pool holds paired token reserves that traders swap against, and the constant product rule keeps the product of the reserves near a constant. A single order, an order book queue, and an oracle feed describe other market designs, not the pooled reserves that an AMM uses.

Question 2

Domain 2: Liquidity Pools and Pricing

DATA

Ava Thompson maps each pool concept to what it describes, with one description missing.

Concept	What it describes
Impermanent loss	(to choose)
Slippage	Price move during trade
Swap fee	Reward to providers
LP token	Claim on the pool

Table 2.1 Pool concepts and meaning.

What does impermanent loss describe?

- ☐ A permanent token burn
- ☐ Loss versus holding
- ☐ Gas paid on deposit
- ☐ A failed transaction

Correct answer: B

Impermanent loss describes how a liquidity position can be worth less than simply holding the two tokens when their relative price moves. It is not a token burn, a gas charge, or a failed transaction, and it can shrink again if prices return toward the entry ratio.

Question 3

Domain 3: Swaps and Routing

CONFIG

Liam Anderson reviews a route built across two pools using an intermediate token.

```
path = [USDC, WETH, DAI]
route = new Route(pools, in, out)
trade = Trade.exactIn(route, amt)
```

Listing 3.1 A route across two pools.

What kind of swap does this route describe?

- ☐ A single-pool swap
- ☐ A liquidity deposit
- ☐ A multi-hop swap
- ☐ A limit order fill

Correct answer: C

A route that passes through two pools using an intermediate token is a multi-hop swap, which lets a trade reach a pair that has no direct pool. A single-pool swap uses one pool, while a deposit and a limit order are different operations entirely.

Question 4

Domain 4: Integrating the Uniswap SDK

CONFIG

David Brown sends a swap through a router but the transaction keeps reverting.

```
const amt = CurrencyAmount.fromRaw(x)
// token spend was never approved
router.swap(amt) // reverts
```

Listing 4.1 A reverting swap call.

Why does the swap revert?

- ☐ A Gas price too high
- ☐ B Wrong chain id set
- ☐ C Deadline in the past
- ☐ D Token not approved

Correct answer: D

An ERC20 swap through a router first needs the token approved for the router to spend, so an unapproved token makes the transfer and the swap revert. Gas price, chain id, and deadline can cause other failures, but the missing approval is what stops the spend here.

Question 5

Domain 5: Concentrated Liquidity and Hooks

CONFIG

Sarah Miller inspects a concentrated liquidity position defined by a tick range.

```
position = {
  tickLower: -60,
  tickUpper: 60,
  liquidity: L }
```

Listing 5.1 A position with a tick range.

Where must the price sit for this position to earn fees?

- ☐ A Outside all ticks
- ☐ B At tick zero only
- ☐ C Within the range
- ☐ D Above the upper tick

Correct answer: C

A concentrated liquidity position earns swap fees only while the market price sits within its chosen tick range, between the lower and upper ticks. Outside the range the position holds a single token and earns nothing, so tick zero or prices above the upper tick do not apply.

Question 6

Domain 6: Security, Testing and Best Practices

CONFIG

James Carter finds a swap that passes zero as the minimum output amount.

```
// swap with no minimum out
router.exactInput(amt, 0)
// amountOutMin = 0
```

Listing 6.1 A swap with no minimum out.

What risk does an amountOutMin of zero create?

- ☐ A No slippage guard
- ☐ B A compiler warning
- ☐ C Higher gas cost
- ☐ D Chain reorg fail

Correct answer: A

Setting amountOutMin to zero removes the slippage guard, so a swap will accept any output and can be sandwiched for a large loss. It does not raise gas, trigger a compiler warning, or cause a reorg, and a sensible minimum output protects the trader.

Question 7 Domain 3: Swaps and Routing

A large swap moves the pool price far against the trader because the trade is big relative to the reserves. What term names this effect?

- ☐ A Impermanent loss
- ☐ B Price impact
- ☐ C Block reward
- ☐ D Gas refund

Correct answer: B

Price impact is the change in the pool price caused by a trade that is large relative to the reserves, and it grows with trade size. Impermanent loss affects providers, while block rewards and gas refunds are unrelated to how a swap moves the pool price.

Question 8 Domain 2: Liquidity Pools and Pricing

A trader sets a very low slippage tolerance during a volatile moment, and the swap keeps reverting. What is the trade-off of raising the tolerance?

- ☐ A Lower gas limit
- ☐ B Fewer confirmations
- ☐ C Smaller pool fee
- ☐ D More MEV exposure

Correct answer: D

Raising slippage tolerance lets a swap succeed during volatility, but a wider bound gives searchers more room to sandwich the trade, increasing MEV exposure. It does not change the gas limit, confirmation count, or pool fee, so the trade-off is protection against reverts versus price safety.

Question 9 Domain 5: Concentrated Liquidity and Hooks

A liquidity provider sets a very narrow price range to earn more fees, but the market price soon moves outside it. What happens to that position?

- ☐ A It earns no fees
- ☐ B It doubles rewards
- ☐ C It burns the tokens
- ☐ D It moves the ticks

Correct answer: A

A narrow range earns more fees while price stays inside it, but once price moves outside the range the position sits fully in one token and earns no fees until price returns. It does not double rewards, burn tokens, or shift its own ticks.

Question 10 Domain 6: Security, Testing and Best Practices

A team deploys a pool integration straight to mainnet without forking or simulating swaps first, and users hit unexpected reverts. What practice would have caught this?

- ☐ A Skipping the audit
- ☐ B Fork testing
- ☐ C Higher gas limit
- ☐ D A bigger pool fee

Correct answer: B

Forking mainnet state and simulating swaps before deployment catches integration reverts that only appear against real pool data. Marketing, a higher gas limit, and a larger fee do not test behavior, so fork based testing is the practice that would have surfaced the problem.