

DeFi on Solana · Week 3

Speaker script: every slide with its notes. Facts checked 11 to 14 Oct 2026; re-check anything marked "verify".

1

Week 3 of 6

3

Liquidity, AMMs & yield

Task

Blog post

Provide liquidity on devnet, or to a pool the class builds

Learning DeFi: my experience with liquidity pools and yield on Solana

DeFi on Solana · Week 3

Week 3 is where the finance background pays off. Market-making, inventory risk, return measurement: they already know the vocabulary. Today we show what those look like when a formula replaces the market maker.

The original curriculum asked for Uniswap testnet or SushiSwap on Mumbai. Mumbai is gone (Polygon shut it in 2024). We use Orca or Raydium on devnet, or a pool the builders deploy.

Plan: 30 minutes of teaching (four parts), then the task live.

2

Today

1

How a market sets a price

2

$x \cdot y = k$, by hand

3

Impermanent loss

4

Smarter curves and routing

5

Where yield really comes from

Today

Five stops. The first two are mechanics, the third is the risk every LP takes, the fourth is how Solana DEXs improve on the basic pool, and the fifth is the habit we want them to leave with: every time you see a yield number, ask who is paying it.

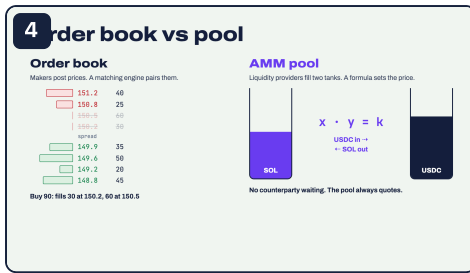
3

How a market sets a price

Two designs. One needs a counterparty. One doesn't.

How a market sets a price

No notes. Let the slide speak.



Order book vs pool

Click 1: the order book. This is what they know from stock exchanges. Makers post bids (green, people willing to buy) and asks (red, people willing to sell). The gap between best bid and best ask is the spread.

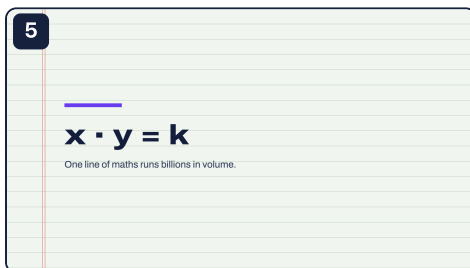
Click 2: a market buy for 90 SOL. It eats the cheapest asks first: 30 at 150.2, then 60 at 150.5. Their average price is worse than the best ask. That is price impact in an order book.

Click 3: the pool. No makers posting prices. Liquidity providers (LPs) deposit both tokens into two "tanks", and a formula sets the price from the ratio of the tanks.

Click 4: the same buy. USDC goes in, SOL comes out, and the formula moves the price. There is no one on the other side waiting; the pool always quotes.

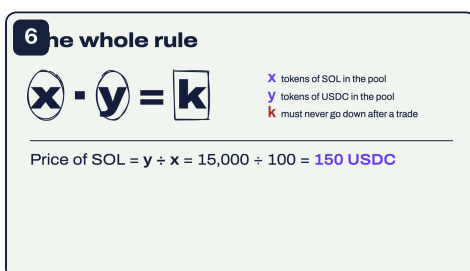
Why AMMs exist: on early blockchains, posting and cancelling thousands of orders was too slow and too expensive. A pool needs no active market maker. On Solana, fast slots and tiny fees made on-chain order books (Phoenix, OpenBook) and "prop AMMs" practical too, which is why the map in part 4 is wider than Ethereum's.

Finance link: an LP is a passive market maker. They earn the spread (as fees) and carry inventory risk (as impermanent loss). Ask the room where the analogy breaks: a real market maker can pull quotes and hedge; a passive LP cannot react.



$$x \cdot y = k$$

No notes. Let the slide speak.



The whole rule

This is the rule Uniswap v2 made famous in 2020. On Solana, Raydium CPMM uses the same maths.

Click 1 and 2: x and y are just the two balances.

Click 4 (after k): the pool's price is the ratio. A pool with 100 SOL and 15,000 USDC prices SOL at $15,000 / 100 = 150$ USDC. Nobody typed that price in; it falls out of the balances.

Click 3: k is the product. A swap can add one token and remove the other, but the product afterwards must be at least k . That one constraint does three jobs at once:

1. It quotes a price for any size of trade.
2. It makes bigger trades more expensive (the pool can never be fully drained, because x or y would hit zero and k would collapse).
3. With a fee, k grows over time, and that growth belongs to the LPs.

Raydium's CPMM pools use exactly this rule. The concentrated and bin designs in part 4 are refinements of it.

7

ne swap, worked by hand

No fee	With a 0.3% LP fee
Pool before: $x = 100$ $y = 100$ $k = 100 \times 100 = 10,000$	0.3% fee laid back: 10×8.997 $= 9.97 \text{ X counts for pricing}$
You add 10 X: $x = 110$ $k \text{ must stay } 10,000$	Pool solves for y: $y = 10,000 / 109.97$ $= 90.934$
Pool solves for y: $y = 10,000 / 110$ $= 90.91$	You receive: $100 - 90.934$ $= 9.066 \text{ Y}$
You receive: $100 - 90.91$ $= 9.09 \text{ Y}$	Pool keeps all 10 X: $k = 110 \times 90.934$ $= 10,002.7 \text{ LPs got richer}$

One swap, worked by hand

This is the exact example from the teacher prep. Do it on the board too.

No fee (clicks 1 to 4):

- Pool holds 100 X and 100 Y. $k = 10,000$. Spot price is 1 Y per X.
- Trader adds 10 X. x becomes 110.
- k must stay 10,000, so $y = 10,000 / 110 = 90.909$.
- Trader receives $100 - 90.909 = 9.09$ Y.
They paid 10 X and got 9.09 Y, so their average price was 0.909 Y per X, about 9% worse than the spot price of 1. That gap is price impact. It is not a fee: nobody earned it. It is the curve.

With a 0.3% fee (clicks 5 to 8):

- The pool keeps 0.3% aside: only 9.97 X counts for pricing.
- $y = 10,000 / 109.97 = 90.934$.
- Trader receives 9.066 Y (slightly less than 9.09).
- But all 10 X went into the pool. New $k = 110 \times 90.934 = 10,002.7$. k grew by 2.7. That growth is the LPs' fee income, shared in proportion to their pool share.

Ask: what happens if the trade is 50 X instead of 10? ($y = 10,000 / 150 = 66.67$, they get 33.33 Y, average 0.667: a 33% worse price.) Then go to the next slide and let them drag it.

8

rag it

Bigger trade, steeper slide down the curve, worse price. Tick the fee and watch k grow.

Drag it

Live widget. Things to show:

1. It opens at 30 X so the legs are visible. Slide to 10: you get 9.09 Y, impact 9.09%. Matches the board.
2. Slide to 100: you get 50 Y for 100 X. Average price 0.5. The pool price after is 0.25 ($y/x = 50/200$). Impact 50%. Nobody sane does this; it shows why depth matters.
3. Tick the 0.3% fee: "You get" drops slightly, and k after goes above 10,000. That green number is the LPs' income.

Price impact vs slippage (students mix these up):

- Price impact: how much YOUR trade moves the price, known before you sign.
- Slippage: how much the price moves between your quote and your execution, because of OTHER trades landing first. Wallets and Jupiter let you set a slippage tolerance (say 0.5%). If the final amount out would be below "quoted $\times (1 - \text{tolerance})$ ", the transaction fails instead of filling badly. A failed transaction still pays its small network fee.

9

Being the pool

Pool fees

Deposit

Both tokens, at the pool's current ratio

Get a receipt

LP tokens (CPMM) or a position NFT (CLMM)

Earn

Every swap leaves a fee in the pool

Withdraw

Burn the receipt, take your share of both tokens

Being the pool

Click 1: Deposit. You must add both tokens at the pool's current ratio. If SOL is 150 USDC, \$300 in means 1 SOL plus 150 USDC.

Click 2: Receipt. A constant-product pool gives you fungible LP tokens representing your share of the pool. A concentrated pool (Orca Whirlpools, Raydium CLMM) gives you a position NFT instead, because every position has its own price range and is not interchangeable.

Click 3: Earn. Every swap leaves its fee in the pool, so the tanks grow. Your share of the tanks grows with them.

Click 4: Withdraw. Burn the receipt and take your share of BOTH tokens, at whatever ratio the pool holds now. That last part is the setup for impermanent loss: you don't get back what you put in, you get back a share of what the pool ended up holding.

Finance link: the LP token is like a unit in a fund. NAV per unit rises with fees and moves with prices.

10

Impermanent loss

The price of being the pool everyone trades against.

Impermanent loss

No notes. Let the slide speak.

11

Why it happens

Elsewhere

SOL jumps to \$200

Your pool

still says \$100

Arbitrageurs buy cheap SOL from the pool until it says \$200

Pool had

5 SOL + 500 USDC

Pool now has

3.54 SOL + 707 USDC

At \$200: holding = \$1,500 - your pool share = \$1,414 - gap = 6.72%

Why it happens

Set-up: you deposit \$1,000 as 5 SOL at \$100 plus 500 USDC. $k = 5 \times 500 = 2,500$.

Click 1: SOL's price jumps to \$200 on every other market.

Click 2: the pool doesn't know. It still prices SOL at \$100. Arbitrageurs buy SOL from your pool (cheap) and sell elsewhere (expensive) until the pool's price is \$200 too. They are the ones who keep pool prices honest, and they profit from it.

Click 3 and 4: where the pool ends up, and what it's worth. Price = $y/x = 200$ and $x \cdot y = 2,500$, so $x = \sqrt{(2,500/200)} = 3.54$ SOL and $y = 707$ USDC.

- Pool value now: $3.54 \times 200 + 707 = \$1,414$.
- If you had just held: $5 \times 200 + 500 = \$1,500$.
- Gap: \$86, or 5.72%. That gap is impermanent loss.

Key insight: the pool sold your SOL on the way up. It always sells the asset that is rising and buys the one that is falling. Finance students will recognise this: it's a short volatility position, like being short an option. Fees are the premium you collect for it.

"Impermanent" because if the price returns to \$100, the gap closes. It becomes permanent the moment you withdraw at a different price. Many people say "divergence loss" instead, which is more honest.

12 The table to memorise

SOL move	Hold	LP	Implement loss
1.25x	\$1125	\$1118	-0.62%
2x	\$1500	\$1414	-5.72%
4x	\$2500	\$2000	-20.00%
0.5x	\$750	\$707	-5.72%

Same loss for 2x and 0.5x. Direction doesn't matter, only distance.

The table to memorise

\$1,000 deposit, 5 SOL at \$100 + 500 USDC, no fees.

- 1.25x: hold \$1,125, LP \$1,118, IL -0.62%. Small moves barely hurt.
- 2x: hold \$1,500, LP \$1,414, IL -5.72%. The example from the previous slide.
- 4x: hold \$2,500, LP \$2,000, IL -20%. Big moves hurt a lot.
- 0.5x: hold \$750, LP \$707, IL -5.72%. A halving costs the same as a doubling.

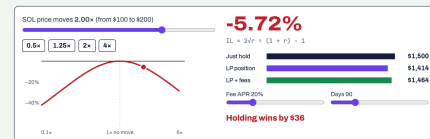
Formula, where r is the price ratio (new price ÷ old price):

$$IL = 2\sqrt{r} \div (1 + r) - 1$$

Students build this table in a spreadsheet in the task. Let them check each row against the formula.

Note the LP is still UP in dollars in the 2x and 4x cases. IL is measured against holding, not against your starting money. That trips people up in their blog posts.

13 Can fees win it back?



Can fees win it back?

Live widget. Same \$1,000 deposit.

Try:

1. Click 2x, fee APR 20%, 90 days: fees are about \$49, IL gap is \$86. Holding wins.
2. Push days to 365: fees about \$200. LP wins.
3. Click 4x: gap is \$500. Even a year of 20% fees doesn't cover it.
4. Click 1.25x: gap is \$7. Fees win almost immediately.

The lesson: LPing is a bet that fees will outrun how far prices drift apart. Stable pairs (USDC/USDT) have almost no IL, so lower fees are enough. Volatile pairs need high fees and high volume to be worth it.

Simplifications in this widget (say them out loud): fee APR is assumed constant and applied to the starting \$1,000; real fee income depends on daily volume and your share of the pool, and it compounds into the position. The point is the shape, not the exact dollars.

Do not let anyone conclude "positive fees means positive returns". That is the exact sentence the curriculum warns about.

14 Quiz: SOL doubles. You were an LP in a SOL/USDC pool with no fees. Compared with just holding, you have:

- ☐ A More dollars
- ☐ B The same
- ☐ C About 5.7% fewer dollars
- ☐ D You lost everything

Quiz: SOL doubles. You were an LP in a SOL/USDC pool with no fees. Compared with just holding, you have:

Most will get this now. If someone picks A, ask them whether their LP value went up in dollars (yes, from \$1,000 to \$1,414) and whether it went up as much as holding (\$1,500). Then they see the distinction.

15

Beyond $x \cdot y = k$

How Solana's DEXs put liquidity where trades happen.

Beyond $x \cdot y = k$

No notes. Let the slide speak.

16 Four ways to hold liquidity



Four ways to hold liquidity

Family tree, because mentees from Ethereum will ask: Uniswap v2 (2020) made $x \cdot y = k$ famous; Raydium CPMM uses the same constant-product maths. Uniswap v3 (2021) introduced concentrated liquidity; Orca Whirlpools and Raydium CLMM follow that design. Meteora DLMM's bins are closer to Trader Joe's Liquidity Book. Uniswap itself has no pools on Solana: its app routes Solana swaps through Jupiter (week 1).

Each mini chart: x axis is price, height is how much liquidity sits at that price, red dashed line is the current price.

Click 1, constant product (Raydium CPMM): liquidity is spread evenly from zero to infinity. Most of it sits at prices that will never trade. Simple, permissionless, but capital-inefficient.

Click 2, concentrated liquidity (Orca Whirlpools, Raydium CLMM): each LP chooses a price range. Depth piles up near the current price, so trades get much less price impact for the same total dollars. The catch is on the next slide.

Click 3, bins (Meteora DLMM, "dynamic liquidity market maker"): liquidity sits in discrete price buckets. Inside one bin, the price is fixed, so a trade that fits inside the active bin (green) has zero slippage. Meteora also uses dynamic fees that rise with volatility.

Click 4, prop AMMs: no passive curve at all. A professional market maker runs its own program and updates its quotes on-chain, often every slot (~400ms). Humidifi led Solana spot DEX volume in late 2025 (roughly 30 to 47%), then BisonFi, another prop AMM, passed it in January 2026. This only works because Solana is fast and cheap enough to update quotes constantly. Ask: is that still "decentralized"? It is permissionless to trade against, but one firm sets the price. Good blog-post material.

For the teacher: be ready to explain that LPs in Whirlpools and CLMM get position NFTs, not LP tokens.

17 Concentrated: deeper, until it isn't



Concentrated: deeper, until it isn't

Live widget. Default: SOL at \$150, range \$120 to \$185.

1. In range: the violet block is much taller than the grey full-range bar. That height multiple is how much deeper your dollars are than a full-range LP's (about 9.7x here). You earn roughly that multiple more fees per dollar.
2. Narrow the range (move low to 140, high to 160): depth jumps to about 30x (the bar is capped so it stays on screen). More fees, but...
3. Drag SOL price to 100: out of range. You now hold 100% SOL (the asset that fell), and you earn nothing until price comes back. Drag to 220: 100% USDC, you sold all your SOL on the way up and missed the rest of the rally.

That is the trade-off in one line: concentrated liquidity means more fees per dollar AND more impermanent loss when price leaves your range. Active LPs rebalance their ranges; Kamino's vaults automate exactly that (back in part 5).

Formula used for depth (Uniswap v3 maths, same as Whirlpools): for a range $[a, b]$ at price p , the multiple vs full range is $2\sqrt{p} \div (2\sqrt{p} - p/\sqrt{b} - \sqrt{a})$.

18 Why Jupiter splits your order



Why Jupiter splits your order

Made-up pool sizes, real constant-product maths.

Click 1: send all 100 SOL through the deepest pool (1,000 SOL deep). Impact = $100 \div 1,100 = 9.1\%$.

Click 2: an aggregator splits the order in proportion to depth: 50 / 30 / 20.

Click 3: each pool now moves only $50 \div 1,050 = 30 \div 630 = 20 \div 420 = 4.8\%$. Same total, about half the impact.

This is what Jupiter does on every quote, across Raydium, Orca, Meteora, prop AMMs and others. Most Solana users never touch a DEX directly; they swap through Jupiter. Week 1's task already had them read a Jupiter route. Remind them of the route they traced.

Builders: the Swap API (GET <https://api.jup.ag/swap/v2/order>, free key from portal.jup.ag) returns the route plan. quote-api.jup.ag/v6 is being shut down. lite-api.jup.ag still works for now (its retirement was postponed with no new date), but teach api.jup.ag with a key.

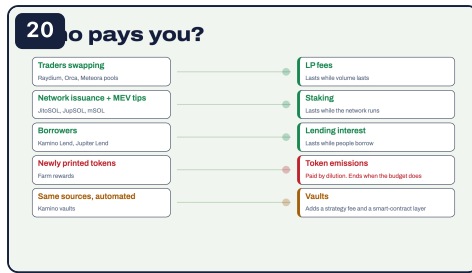
19

Where yield comes from

Every yield has a payer. Find them.

Where yield comes from

No notes. Let the slide speak.



Who pays you?

The analyst habit for this whole course: for every yield number, ask who pays it, what activity generates it, and what makes it stop.

Click 1, LP fees: paid by traders. Real and sustainable as long as volume lasts. Minus impermanent loss.

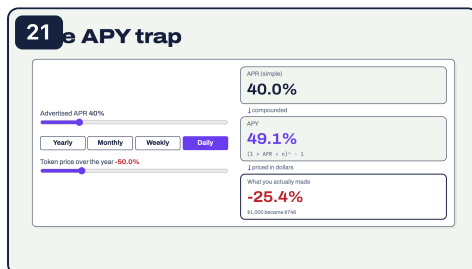
Click 2, staking: SOL stakers earn network issuance (inflation rewards) plus a share of MEV tips. Liquid staking tokens like JitoSOL pass both through and stay tradeable; JitoSOL is the one known for adding MEV tips on top of staking. Others: JupSOL, mSOL (Marinade), Sanctum's family. Risk: the LST can trade below its SOL value in a panic, and there is smart-contract risk.

Click 3, lending interest: paid by borrowers. Next week in detail.

Click 4, token emissions: the protocol prints its own token and hands it out to attract deposits. This is the "yield farming" of DeFi Summer 2020. Paid by dilution of everyone holding the token. It ends when the budget runs out, and the token's price often falls as farmers sell. Red, deliberately.

Click 5, vaults (Kamino): no new source of money. A vault automates one of the strategies above (for example, rebalancing a concentrated LP range) and takes a fee. You add a strategy layer and another contract to trust.

Finance link: separate protocol revenue (real fees from real users) from incentives (subsidies). The curriculum asks for this distinction explicitly in the week 3 report.



The APY trap

Live widget. \$1,000 of a farm token, rewards paid in the same token, one year.

1. APR vs APY: APR is simple interest. APY includes compounding: $(1 + \text{APR}/n)^n - 1$. At 40% APR compounded daily, APY is 49.2%. Protocols love quoting APY because it's the bigger number. Set it to Yearly: they match.
2. The trap: drag token price to -50%. You end the year with 1,492 tokens worth \$746. A "49% APY" farm lost you 25% in dollars.
3. Drag APR to 200% and token price to -70%: still a loss. This is exactly how many 2020 to 2022 farms played out.

Questions to ask the room:

- Is the yield denominated in the token you care about?
- Is the APY a historical average or the last 24 hours annualised?
- Is it gross or net of fees and IL?

The curriculum's line: a high advertised yield does not necessarily mean a good investment. This widget is the proof.

22 A company advertises 300% APY, paid in its own new token. What is the first question to ask?

- A** Is it audited?
- B** Who is paying this, and in what?
- C** Daily or weekly compounding?
- D** What is the TVL?

All four are reasonable questions, and B comes first. Audits and TVL come back in week 5 due diligence.

23

for the task: two "USDC"s

USDC

Issued by Circle's devnet faucet

`4z9PCVY...3DnGDU`

What you got in week 2

+

Same name, different mint. A program only checks the mint.

devUSDC

Minted by Orca for its devnet pools

`BRj5Ck...3hokXk`

What the Orca devnet pool wants

On mainnet, anyone can mint a token called "USDC". [Check the mint address.](#)

Teaching moment from the handbook, and the best possible proof of week 2's rule "the mint address is the identity".

Click 2: Orca's devnet examples use their own devUSDC, mint
BRjpCHtyQLNCo8gqRUr8jtdAj5AjPYQaoqbvcZiHok1k.

Mainnet lesson: scammers mint tokens named USDC all the time. Always check the mint address against the issuer's official one.

24 Task: be a liquidity provider

Users

Pick the route your class can run:

- A Raydium devnet UI: swap, add liquidity
- C Add liquidity to the class pool with your two week-2 tokens

Then build the impermanent-loss table: 1.25x, 2x, 4x.

Test the route yourself first

Raydium: devnet setting on raydium.io, program IDs from docs raydium.io only. Orca pool: 3MBZLL...c3kP5

Builders

- B Orca SDK on devnet:
- `solana-cli config --url https://api.devnet.10101.com`, open a position in the SOL/devUSDC Whirlpool
- C Deploy the constant-product pool from `program-examples/tokens/token-swap` or Blueshift's Pinocchio AMM

Task: be a liquidity provider

Three routes from the handbook. Pick the one that works in your class, after testing it yourself 48 hours before.

A. Raydium devnet. Raydium announced a working devnet UI through its infra account (@0xINFRA) on 15 July 2025: swap, create pools, test tokens. It is a devnet setting inside raydium.io, not a separate site; find the toggle yourself before class. Use the devnet program IDs from docs.raydium.io/reference/program-addresses: CPMM DRaycpLY18LhpbydsBWbVJtxpNv9oXPgJRSfpF2bWpYb, CLMM DRayAUgENGQBKVaX8owNhgzkEDyoHTGVEGHVJT1E9pfH, AMM v4 DRaya7Kj3aMWQSy19kSjvmuwq9docCHofyP9kanQGaav. The raydium-cpi-example README lists a different CPMM ID (CPMDW...) that does not match the UI; that issue (#16) was still open on 14 Oct 2026. Devnet APIs: api-v3-devnet.raydium.io and transaction-v1-devnet.raydium.io.

B. Orca SDK on devnet. Orca's docs use an existing SOL/devUSDC Whirlpool on devnet and setWhirlpoolsConfig('solanaDevnet'). Whirlpools program: whirLbMiicVdio4qvUfM5KAg6Ct8VwpYzGff3uuctyCc. Pool: 3KBZiL2g8C7tiJ32hTv5v3KM7aK9htpqTw4cTXz1HvPt. Package: @orca-so/whirlpools. Remember it needs Orca's devUSDC, not Circle's.

C. Build the pool. Builders deploy the constant-product example from solana-developers/program-examples/tokens/token-swap (Anchor; create pool, deposit, withdraw, swap; fee in basis points) or do Blueshift's Pinocchio AMM challenge. A third reference: the constant-product-curve crate (github.com/deanmlittle/constant-product-curve) that Turbin3's Anchor AMM class builds on; it handles the curve maths, LP minting and slippage so the program can focus on accounts. builders/anchor/programs/cp-amm in this repo is a smaller, fully tested version. Users then provide liquidity to it with two tokens minted in week 2. This is the most reliable route because you control every piece.

Everyone: the impermanent-loss spreadsheet with at least three price scenarios, and a comparison of four yield sources.

25

Blog post

"Learning DeFi: my experience with liquidity pools and yield on Solana"

Include your impermanent-loss table, and one yield you found and who pays it.

Blog post

Required: the impermanent-loss table. Encourage one honest paragraph on whether they would LP with real money, and why.

The curriculum's week 3 report also asks for: order book vs AMM comparison, a worked AMM trade, an LP return model with at least three scenarios, four sources of yield compared, an assessment of one real Solana exchange, and why advertised yield is not a guaranteed return. The blog post can be the short version.

26

Teacher prep

For you, before class. Skip this live.

Teacher prep

No notes. Let the slide speak.

27 You, before class

1. **Work the swap by hand.** 100 / 100 pool, swap 10 X in: 9.09 Y out. Redo with 0.3% fee: 9.066.
2. **Build the IL spreadsheet.** 1.25x, 2x, 4x LP vs holding. Reuse it in class.
3. **Learn three curve families.** CPMM, concentrated (Whirlpools, CLMM), bins (DLMM). Then how a prop AMM differs.
4. **Skim Blueshift's Pinocchio AMM** (deposit, swap) so you can point builders to it.
5. **Run your chosen task route** on a fresh wallet, 48 hours before.

You, before class

Straight from the handbook's "You, before class" lane for week 3.

The swap: $k = 10,000$. Swap 10 X in: $y = 10,000 / 110 \approx 90.91$, trader gets ≈ 9.09 Y. With a 0.3% fee: 9.97 X counts, $y = 10,000 / 109.97 \approx 90.934$, trader gets ≈ 9.066 Y, k rises to $\approx 10,002.7$.

The IL sheet: columns for price ratio r , hold value, LP value (= starting value $\times \sqrt{r}$ for a 50/50 pool), $IL = 2\sqrt{r}/(1+r) - 1$. Values: $1.25x \rightarrow -0.62\%$, $2x \rightarrow -5.72\%$, $4x \rightarrow -20.0\%$.

Curve families: constant product (Raydium CPMM), concentrated liquidity (Orca Whirlpools, Raydium CLMM), bins (Meteora DLMM). A prop AMM is a market maker quoting its own prices on-chain instead of a passive curve.

Blueshift: learn.blueshift.gg, Pinocchio AMM challenge (constant-product implementation with deposit and swap).

28 Addresses and checks

What	Value
Orca Whirlpools program (devnet)	9WzDfXkHvG8t8Zs4nFf85qgCt8Wp7z8FF3a1vGQ
Orca SOLdevUSDC pool	3BZLL3gK7CL33ZnT5d5u9W7uW1p2t4u7G13HAPL
Orca devUSDC mint	8E3uChyVGLKc8p8R9H415d35APYDn8pvcZ3H4K5K
Circle devnet USDC mint	4z9RCFur15R15X14dAgXaHh13d9FAERFP3g23n2d8E
Raydium devnet APs	ap0-v3-devnet.raydium.io , transaction-v3-devnet.raydium.io
Pool example	solana-devnet/solanaprogram-examples/tokens/solana-swap

Verify before class

Raydium's devnet is a setting on raydium.io (announced 15 Jul 2026). Use program IDs from docs.raydium.io, not the api-example README (CPMM mismatch, issue #18, still open).

Addresses and checks

All values from the course handbook, checked 11 October 2026. Re-check the Raydium route the week you teach. If it fails, fall back to route C (the class pool), which only depends on devnet itself.