ERC1363 Tokens and Bonding Curves

This material is copyrighted by RareSkills. External distribution is prohibited.

For the next two weeks we will build some intermediate to advanced level token applications so that we can test them in week 3

What is Due at the End of the Week

□ Read up on ERC777 (code) and ERC1363 (code). See discussion on this issue and this tweet thread. What is the design motivation behind these EIPs and the pros and cons of both tokens?

For the following assignments, you are encouraged to use ERC1363, but you can use whichever standard you like.

□ Token with sanctions. Create a fungible token that allows an admin to ban specified addresses from sending and receiving tokens.

□ Token with god mode. A special address is able to transfer tokens between addresses at will.

□ (**hard**) Token sale and buyback with bonding curve. The more tokens a user buys, the more expensive the token becomes. To keep things simple, use a linear bonding curve. When a person sends a token to the contract with ERC1363 or ERC777, it should trigger the receive function. If you use ERC20, you need to use the approve and send workflow. This should support fractions of tokens.

You can use this article as a starting point, but you are encouraged to do your own research on how bonding curves work works.
https://www.linumlabs.com/articles/bonding-curves-the-what-why-and-shapes-behind-it

Where students commonly mess up

- Not putting proper access controls on the functions

- Using magic numbers (numbers instead of constant variables)
- Not assigning the keyword constant or immutable to variables that are never updated.
- Using public function modifiers when external is sufficient
- Ignoring the decimal place
- Not using the OpenZeppelin implementation of ERC20
- Rebuilding useful functions OpenZeppelin already provides like _beforeTokenTransfer()
- Writing large numbers as 1000000 instead of 1_000_000
- Floating pragma (learn more)
- Natspec
- Using virtual where you shouldn’t
- **Please use a formatter to make your code look nice**