

[Multiple Inheritance Guide](#)[Abstract Contracts](#)[Abstract Contracts Guide](#)

Exercise

[Imports >](#)[Errors >](#)[The new Keyword >](#)[Contract to Contract Interactions >](#)[Events >](#)[Address and Payable >](#)

Development with Foundry

[Deploying a smart contract using Foundry](#)[Foundry: Setting up Foundry with Base](#)

Inheritance

Inheritance Exercise

 [Copy page](#)

Exercise - Demonstrate your knowledge of inheritance.

Create contracts that adhere to the following specifications.

Contracts

Employee

Create an `abstract` contract called `Employee`. It should have:

A public variable storing `idNumber`

A public variable storing `managerId`

A constructor that accepts arguments for and sets both of these variables

A `virtual` function

Ask a question...

Ctrl+I

uint

Salaried

A contract called `Salaried` . It should:

Inherit from `Employee`

Have a public variable for `annualSalary`

Implement an `override` function for `getAnnualCost` that returns `annualSalary`

An appropriate constructor that performs any setup, including setting `annualSalary`

Multiple Inheritance Guide

Abstract Contracts

Abstract Contracts Guide

[Exercise](#)

Hourly

Implement a contract called `Hourly` . It should:

Inherit from `Employee`

Have a public variable storing `hourlyRate`

Include any other necessary setup and implementation

 The annual cost of an hourly employee is their hourly rate * 2080 hours.

Development with Foundry

Deploying a smart contract
using Foundry

Foundry: Setting up Foundry
with Base

Manager

Implement a contract called `Manager` . It should:

Have a public array storing employee Ids

Include a function called `addReport` that can add id numbers to that array

Include a function called `resetReports` that can reset that array to empty

Multiple Inheritance Guide

Abstract Contracts

Abstract Contracts Guide

[Exercise](#)

Salesperson

Implement a contract called `Salesperson` that inherits from `Hourly`.

Engineering Manager

Implement a contract called `EngineeringManager` that inherits from `Salaried` and `Manager`.

Deployments

You'll have to do a more complicated set of deployments for this exercise.

Deploy your `Salesperson` and `EngineeringManager` contracts. You don't need to separately deploy the other contracts.

Use the following values:

Salesperson

Development with Foundry

Deploying a smart contract
using Foundry

Foundry: Setting up Foundry

with Base

Hourly rate is 20 dollars an hour

Id number is 55555

Manager Id number is 12345

Multiple Inheritance Guide

Abstract Contracts

Abstract Contracts Guide

[Exercise](#)

Manager

Annual salary is 200,000

Id number is 54321

Manager Id is 11111

Inheritance Submission

Copy the below contract and deploy it using the addresses of your `Salesperson` and `EngineeringManager` contracts.

```
contract InheritanceSubmission {
    address public salesPerson;
    address public engineeringManager;

    constructor(address _salesPerson, address _engineeringManager) {
        salesPerson = _salesPerson;
        engineeringManager = _engineeringManager;
    }
}
```



Development with Foundry

Deploying a smart contract
using Foundry

Foundry: Setting up Foundry

with Base

Submit your Contracts and Earn an NFT Badge! (BETA)

⚠ Hey, where'd my NFT go!?

Testnets are not permanent! Base Goerli will soon be sunset, in favor of Base Sepolia.

As these are separate networks with separate data, your NFTs **will not** transfer over.

Don't worry! We've captured the addresses of all NFT owners on Base Goerli and will include them when we release the mechanism to transfer these NFTs to mainnet later this year! You can also redeploy on Sepolia and resubmit if you'd like!

⚠ Submit your address for your copy of the `InheritanceSubmission` contract that contains your other contract addresses.

Multiple Inheritance Guide

Abstract Contracts

Abstract Contracts Guide

Exercise

Development with Foundry

Deploying a smart contract using Foundry

Foundry: Setting up Foundry with Base

Was this page helpful?

Yes

No

Suggest edits

Raise issue

Multiple Inheritance Guide

Abstract Contracts

 basedocs
Abstract Contracts Guide

base.org

Blog

Privacy Policy

Terms of Service

Cookie Policy

Exercise

Development with Foundry

Deploying a smart contract
using Foundry

Foundry: Setting up Foundry
with Base