CS5800 Homework 03
Subham Panda
017314921

[Link to GitHub Repo](#)

Class Diagrams
Part 1
Code

```plantuml
@startuml

abstract class Employee {
  -firstName : String
  -lastName : String
  -socialSecurityNumber : String
  +getFirstName() : String
  +setFirstName(firstName : String) : void
  +getLastName() : String
  +setLastName(lastName : String) : void
  +getSocialSecurityNumber() : String
  +setSocialSecurityNumber(socialSecurityNumber : String) : void
  +toString() : String
}

class SalariedEmployee {
  -weeklySalary : int
  +getWeeklySalary() : int
  +setWeeklySalary(weeklySalary : int) : void
  +toString() : String
}

class HourlyEmployee {
  -wage : int
  -hoursWorked : int
  +getWage() : int
  +setWage(wage : int) : void
  +getHoursWorked() : int
  +setHoursWorked(hoursWorked : int) : void
  +toString() : String
}

class CommisionEmployee {
  -commissionRate : int
  -grossSales : int
  +getCommissionRate() : int
  +setCommissionRate(commissionRate : int) : void
  +getGrossSales() : int
  +setGrossSales(grossSales : int) : void
  +toString() : String
}

class BaseEmployee {
```
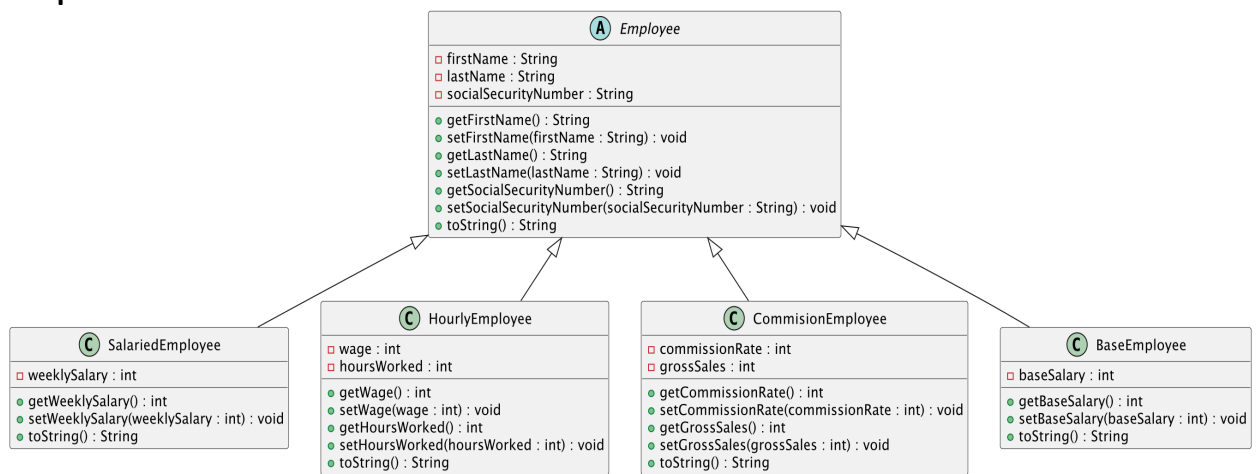
```
  -baseSalary : int
  +getBaseSalary() : int
  +setBaseSalary(baseSalary : int) : void
  +toString() : String
}

Employee <|-- SalariedEmployee
Employee <|-- HourlyEmployee
Employee <|-- CommisionEmployee
Employee <|-- BaseEmployee

@enduml
```

Output

Part 2
Code

```
@startuml

' Definition for the base Ship class
class Ship {
    #shipName : String
    #yearBuilt : String

    +Ship(shipName : String, yearBuilt : String)
    +setShipName(shipName : String) : void
    +getShipName() : String
    +setYearBuilt(yearBuilt : String) : void
    +getYearBuilt() : String
    +printShip() : void
}

' Definition for the CruiseShip class that extends Ship
class CruiseShip extends Ship {
    -maxNumberPassengers : int

    +CruiseShip(shipName : String, yearBuilt : String, maxNumberPassengers :
int)
    +setMaxNumberPassengers(maxNumberPassengers : int) : void
    +getMaxNumberPassengers() : int
    +printShip() : void
}

' Definition for the CargoShip class that extends Ship
class CargoShip extends Ship {
    -cargoCapacityInTonnage : int

    +CargoShip(shipName : String, yearBuilt : String, cargoCapacityInTonnage
: int)
    +setCargoCapacityInTonnage(cargoCapacityInTonnage : int) : void
    +getCargoCapacityInTonnage() : int
    +printShip() : void
}

Ship <|-- CruiseShip
Ship <|-- CargoShip

@enduml
```
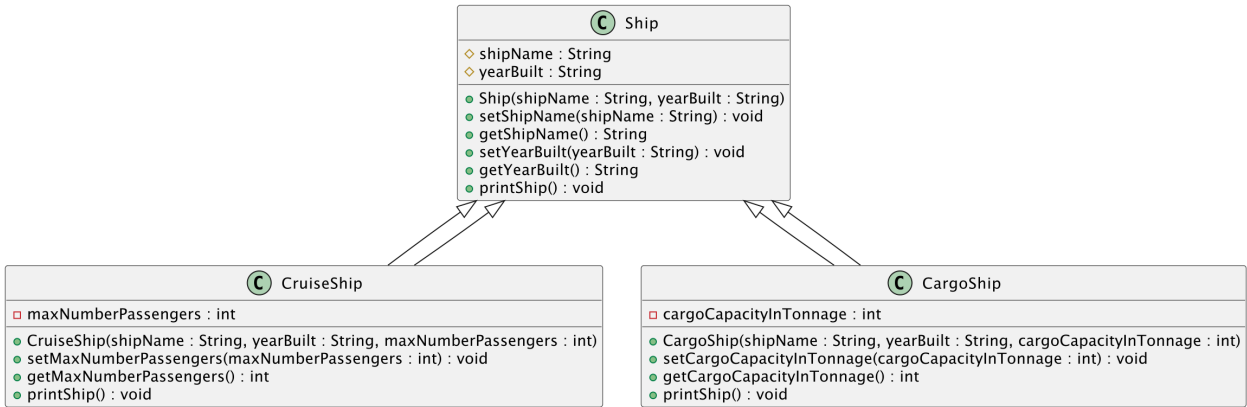
Output

```
                          ┌─────────────────────────────────────────┐
                          │           Ⓒ  Ship                       │
                          ├─────────────────────────────────────────┤
                          │ ◇ shipName : String                     │
                          │ ◇ yearBuilt : String                    │
                          ├─────────────────────────────────────────┤
                          │ ● Ship(shipName : String, yearBuilt : String) │
                          │ ● setShipName(shipName : String) : void │
                          │ ● getShipName() : String                │
                          │ ● setYearBuilt(yearBuilt : String) : void │
                          │ ● getYearBuilt() : String               │
                          │ ● printShip() : void                    │
                          └─────────────────────────────────────────┘
```

CruiseShip

- maxNumberPassengers : int

- CruiseShip(shipName : String, yearBuilt : String, maxNumberPassengers : int)
- setMaxNumberPassengers(maxNumberPassengers : int) : void
- getMaxNumberPassengers() : int
- printShip() : void

CargoShip

- cargoCapacityInTonnage : int

- CargoShip(shipName : String, yearBuilt : String, cargoCapacityInTonnage : int)
- setCargoCapacityInTonnage(cargoCapacityInTonnage : int) : void
- getCargoCapacityInTonnage() : int
- printShip() : void

Part 3
Code

```
@startuml

' Definition of the Instructor class
class Instructor {
    -firstName : String
    -lastName : String
    -officeNumber : String

    +setFirstName(firstName : String) : void
    +getFirstName() : String
    +setLastName(lastName : String) : void
    +getLastName() : String
    +setOfficeNumber(officeNumber : String) : void
    +getOfficeNumber() : String
}

' Definition of the Textbook class
class Textbook {
    -title : String
    -author : String
    -publisher : String

    +setTitle(title : String) : void
    +getTitle() : String
    +setAuthor(author : String) : void
    +getAuthor() : String
    +setPublisher(publisher : String) : void
    +getPublisher() : String
}

' Definition of the Course class, which aggregates Instructor and Textbook
class Course {
    -name : String
    -instructors : ArrayList<Instructor>
    -textbooks : ArrayList<Textbook>

    +setName(name : String) : void
    +getName() : String
    +setInstructor(instructor : Instructor) : void
    +getInstructors() : ArrayList<Instructor>
    +setTextbook(textbook : Textbook) : void
    +getTextbooks() : ArrayList<Textbook>
    +printCourse() : void
}

Course "1" *-- "many" Instructor : has >
Course "1" *-- "many" Textbook : has >

@enduml
```
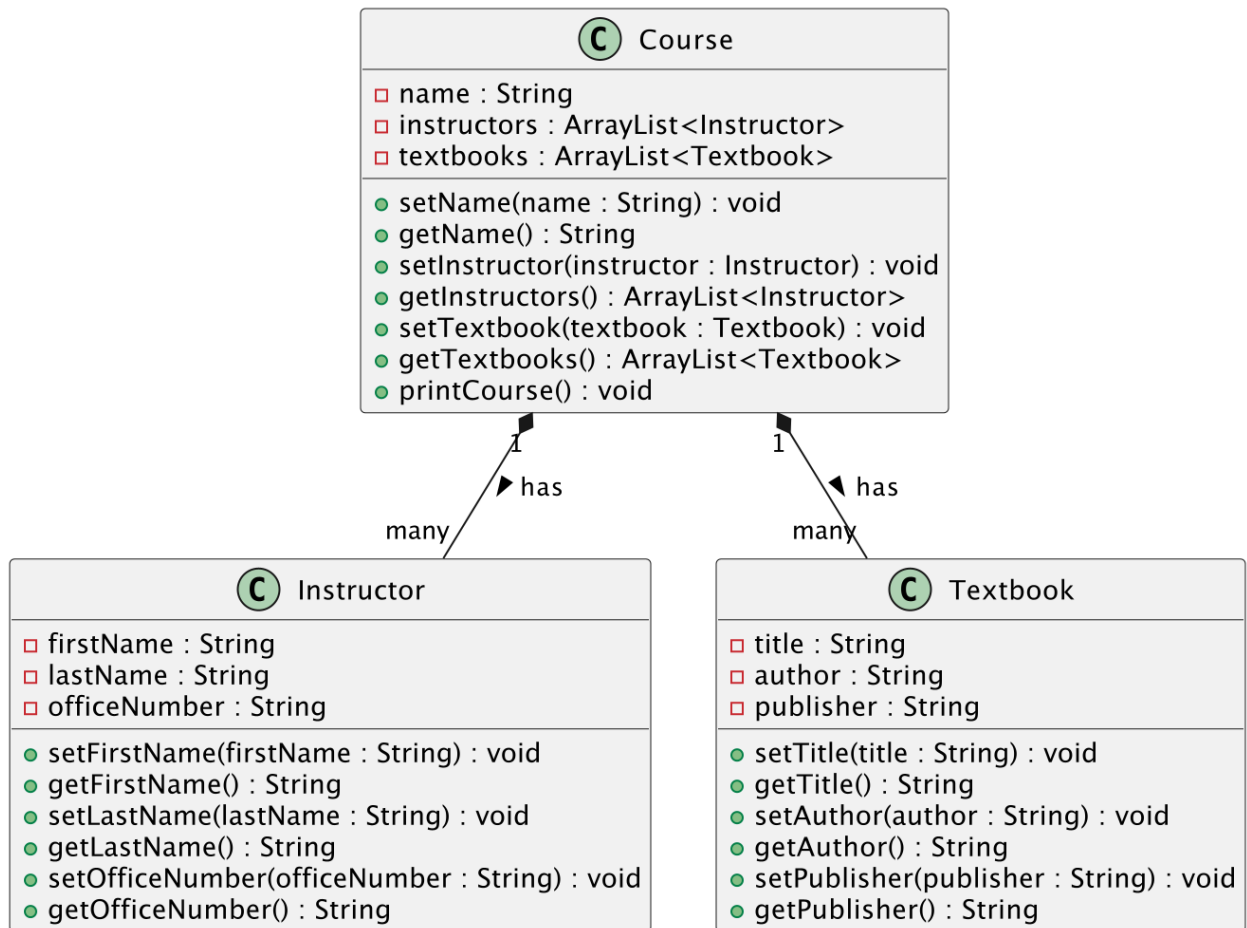
Output

```
                    ┌─────────────────────────────────────────┐
                    │         Ⓒ  Course                        │
                    ├─────────────────────────────────────────┤
                    │ □ name : String                          │
                    │ □ instructors : ArrayList<Instructor>    │
                    │ □ textbooks : ArrayList<Textbook>        │
                    ├─────────────────────────────────────────┤
                    │ ○ setName(name : String) : void          │
                    │ ○ getName() : String                      │
                    │ ○ setInstructor(instructor : Instructor) : void │
                    │ ○ getInstructors() : ArrayList<Instructor> │
                    │ ○ setTextbook(textbook : Textbook) : void │
                    │ ○ getTextbooks() : ArrayList<Textbook>    │
                    │ ○ printCourse() : void                    │
                    └─────────────────────────────────────────┘
                         1                          1
                          ▶ has                      ◀ has
                   many                          many
┌──────────────────────────────────────────┐   ┌──────────────────────────────────────────┐
│         Ⓒ  Instructor                      │   │         Ⓒ  Textbook                        │
├──────────────────────────────────────────┤   ├──────────────────────────────────────────┤
│ □ firstName : String                       │   │ □ title : String                           │
│ □ lastName : String                        │   │ □ author : String                          │
│ □ officeNumber : String                    │   │ □ publisher : String                       │
├──────────────────────────────────────────┤   ├──────────────────────────────────────────┤
│ ○ setFirstName(firstName : String) : void  │   │ ○ setTitle(title : String) : void          │
│ ○ getFirstName() : String                   │   │ ○ getTitle() : String                      │
│ ○ setLastName(lastName : String) : void     │   │ ○ setAuthor(author : String) : void        │
│ ○ getLastName() : String                     │   │ ○ getAuthor() : String                     │
│ ○ setOfficeNumber(officeNumber : String) : void │ ○ setPublisher(publisher : String) : void │
│ ○ getOfficeNumber() : String                 │   │ ○ getPublisher() : String                  │
└──────────────────────────────────────────┘   └──────────────────────────────────────────┘
```

Part 4
Code

```
@startuml

' Definition of the File class
class File {
    -name : String

    +File(name : String)
    +setName(name : String) : void
    +getName() : String
    +toString() : String
}

' Definition of the Folder class, which owns File objects and other Folder
objects
class Folder {
    -name : String
    -subfolders : ArrayList<Folder>
    -files : ArrayList<File>

    +Folder()
    +setName(name : String) : void
    +getName() : String
    +addSubfolder(subFolderName : String) : void
    +removeSubfolder(subFolderName : String) : void
    +getSubfolder(subFolderName : String) : Folder
    +addFile(fileName : String) : void
    +removeFile(file : File) : boolean
    +toString() : String
    +printFolderContents() : void
}

Folder "1" *-- "many" Folder : contains >
Folder "1" *-- "many" File : contains >

@enduml
```
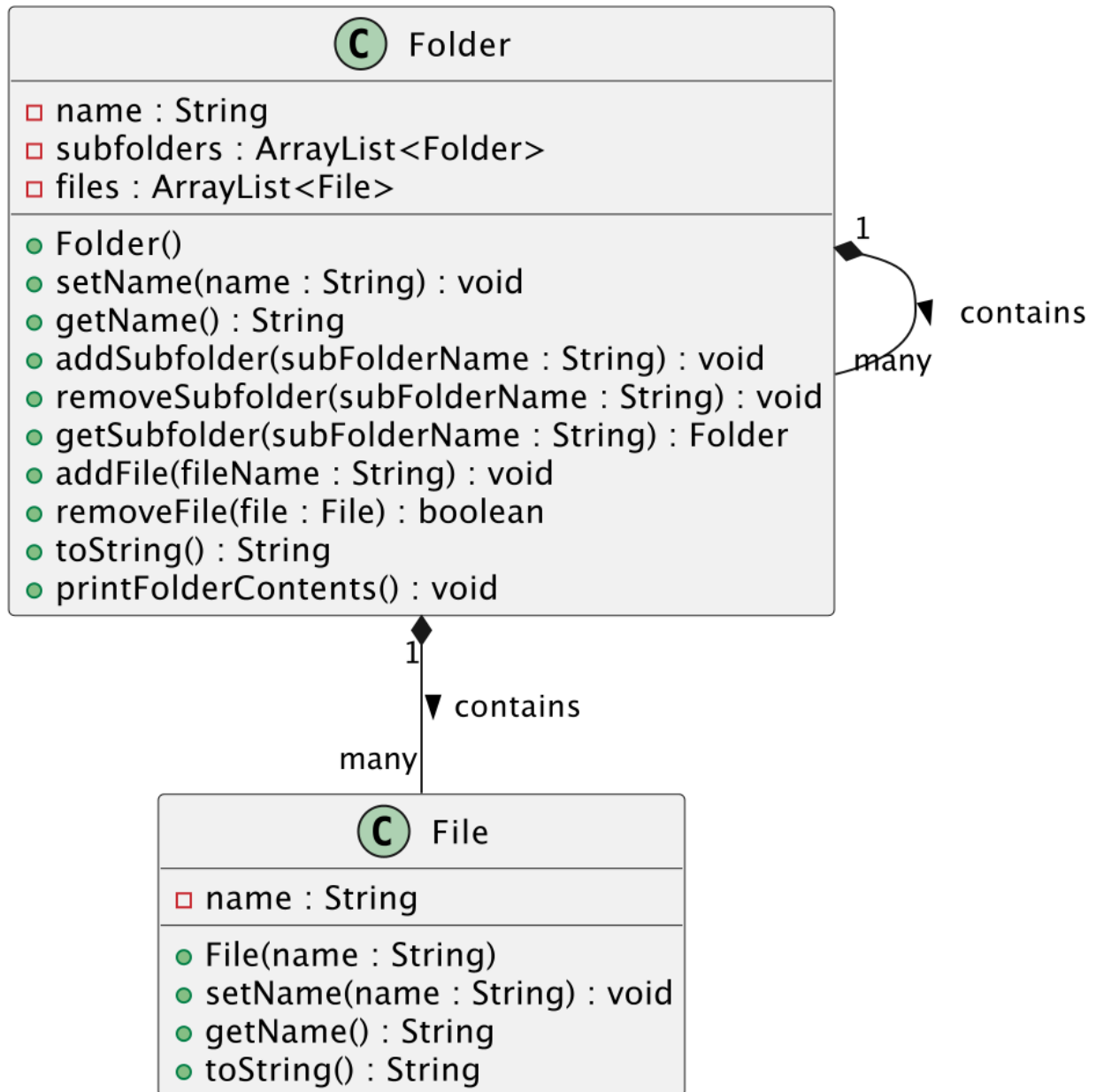
Output

C Folder

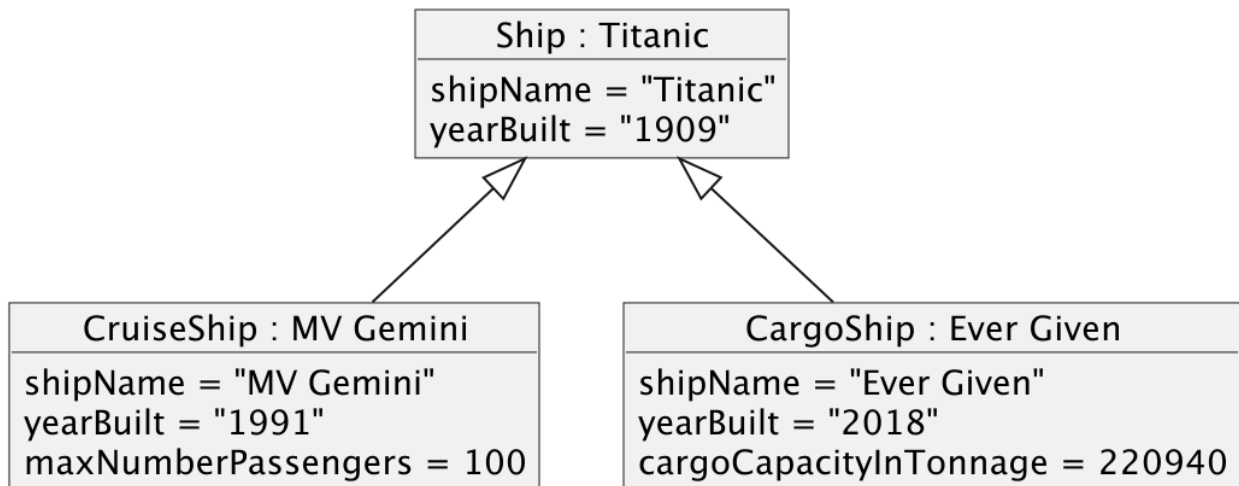☐ name : String
☐ subfolders : ArrayList<Folder>
☐ files : ArrayList<File>

○ Folder()
○ setName(name : String) : void
○ getName() : String
○ addSubfolder(subFolderName : String) : void
○ removeSubfolder(subFolderName : String) : void
○ getSubfolder(subFolderName : String) : Folder
○ addFile(fileName : String) : void
○ removeFile(file : File) : boolean
○ toString() : String
○ printFolderContents() : void

1

▼ contains

many

1

▼ contains

many

C File

☐ name : String

○ File(name : String)
○ setName(name : String) : void
○ getName() : String
○ toString() : String

Object Diagrams

Part 2

Code

```
@startuml

object "Ship : Titanic" as Ship1 {
  shipName = "Titanic"
  yearBuilt = "1909"
}

object "CruiseShip : MV Gemini" as CruiseShip1 {
  shipName = "MV Gemini"
  yearBuilt = "1991"
  maxNumberPassengers = 100
}

object "CargoShip : Ever Given" as CargoShip1 {
  shipName = "Ever Given"
  yearBuilt = "2018"
  cargoCapacityInTonnage = 220940
}

Ship1 <|-- CruiseShip1
Ship1 <|-- CargoShip1

@enduml
```
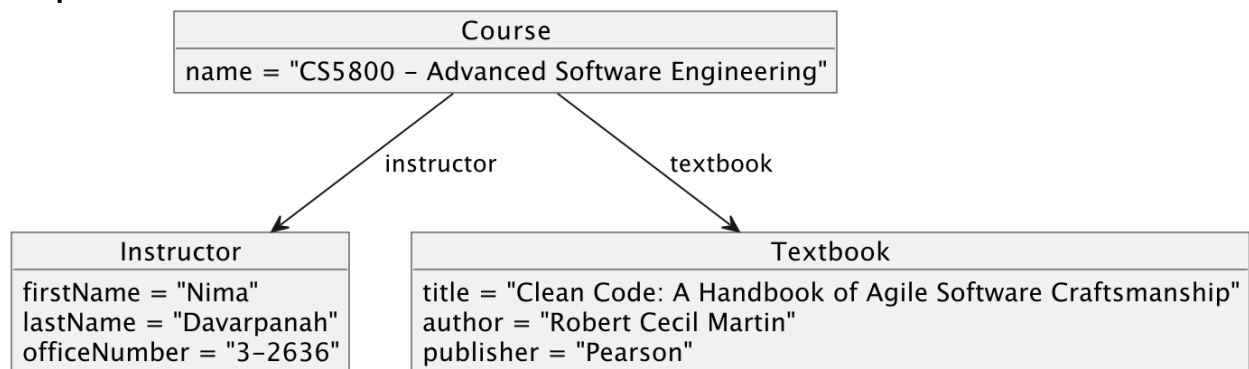
Output

Part 3
Code

```
@startuml

object Instructor {
  firstName = "Nima"
  lastName = "Davarpanah"
  officeNumber = "3-2636"
}

object Textbook {
  title = "Clean Code: A Handbook of Agile Software Craftsmanship"
  author = "Robert Cecil Martin"
  publisher = "Pearson"
}

object Course {
  name = "CS5800 - Advanced Software Engineering"
}

Course --> Instructor : instructor
Course --> Textbook : textbook

@enduml
```

Output

Part 4

Code

```
@startuml

' Define instances of Folder and File after the 'app' folder deletion

object Folder {
  name = "php_demo1"
  files = []
  subfolders = ["Source Files", " Include Path", "Remote Files"]
}

object SubFolder1 {
  name = "Source Files"
  files = []
  subfolders = [".phalcon", "cache", "public"]
}

object SubFolder1_1 {
  name = ".phalcon"
  files = []
  subfolders = []
}

object SubFolder1_3 {
  name = "cache"
  files = []
  subfolders = []
}

object SubFolder1_4 {
  name = "public"
  files = [".htaccess", ".htrouter.php", "index.html"]
  subfolders = []
}

object File1_4_1 {
  name = ".htaccess"
}

object File1_4_2 {
  name = ".htrouter.php"
}

object File1_4_3 {
  name = "index.html"
}

object SubFolder2 {
  name = "Include Path"
  files = []
  subfolders = []
}

object SubFolder3 {
  name = "Remote Files"
```

```
    files = []
    subfolders = []
}

' Define the links representing the composition relationships
Folder  *-- SubFolder1 : subfolder
SubFolder1 *-- SubFolder1_1 : subfolder
SubFolder1 *-- SubFolder1_3 : subfolder
SubFolder1 *-- SubFolder1_4 : subfolder
SubFolder1_4 *-- File1_4_1 : file
SubFolder1_4 *-- File1_4_2 : file
SubFolder1_4 *-- File1_4_3 : file
Folder *-- SubFolder2 : subfolder
Folder *-- SubFolder3 : subfolder

@enduml
```

Output