Chaitanya Ganpat Nalage
017300803

CS 5800.01 - Advanced Software Engineering
Homework – 03

Github Link: https://github.com/chaitanyanalage/CS5800/tree/main/Homework03

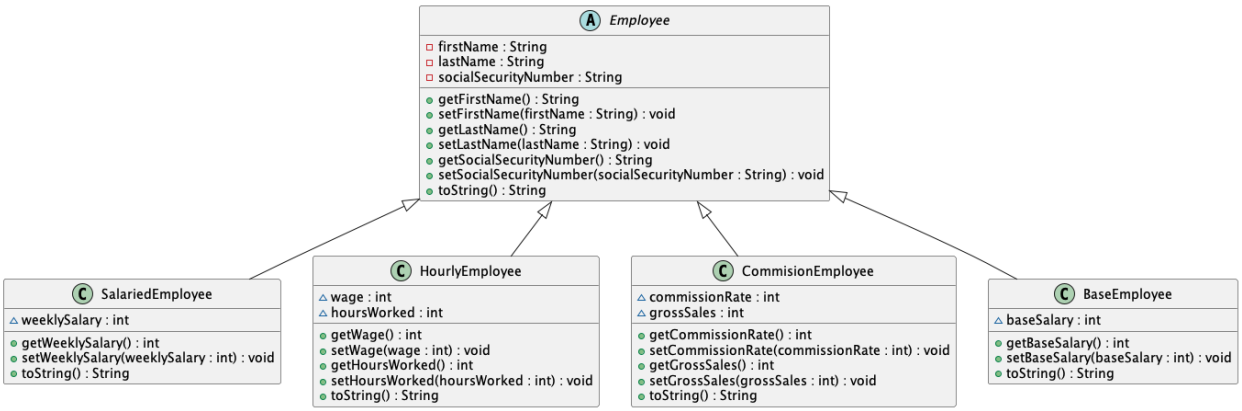
Class Diagrams:

Inheritance:

```
@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
```

```
37      +setCommissionRate(commissionRate : int) : void
38      +getGrossSales() : int
39      +setGrossSales(grossSales : int) : void
40      +toString() : String
41    }
42
43    class BaseEmployee {
44      ~baseSalary : int
45      +getBaseSalary() : int
46      +setBaseSalary(baseSalary : int) : void
47      +toString() : String
48    }
49
50    Employee <|-- SalariedEmployee
51    Employee <|-- HourlyEmployee
52    Employee <|-- CommisionEmployee
53    Employee <|-- BaseEmployee
54
55    @enduml
```

Output:



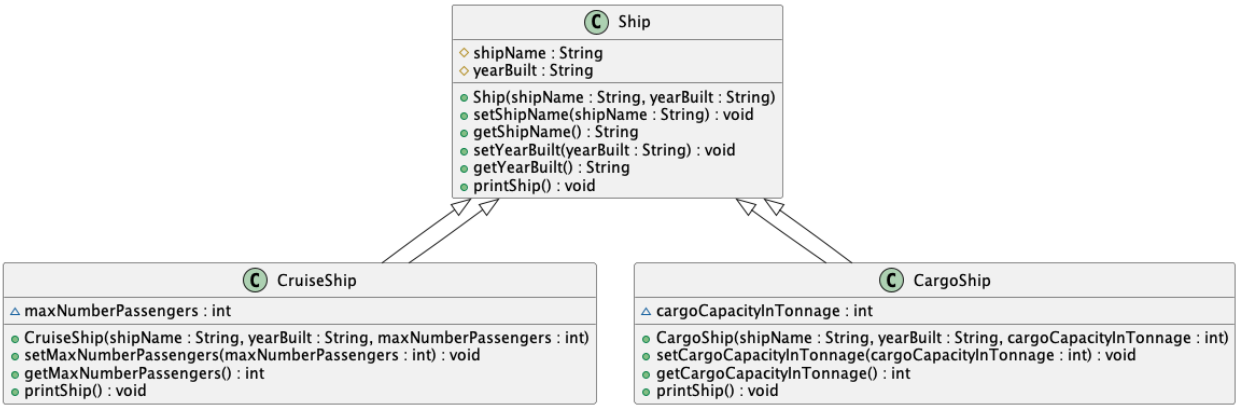
Polymorphism:

```
@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
```

Chaitanya Ganpat Nalage
017300803

Output:

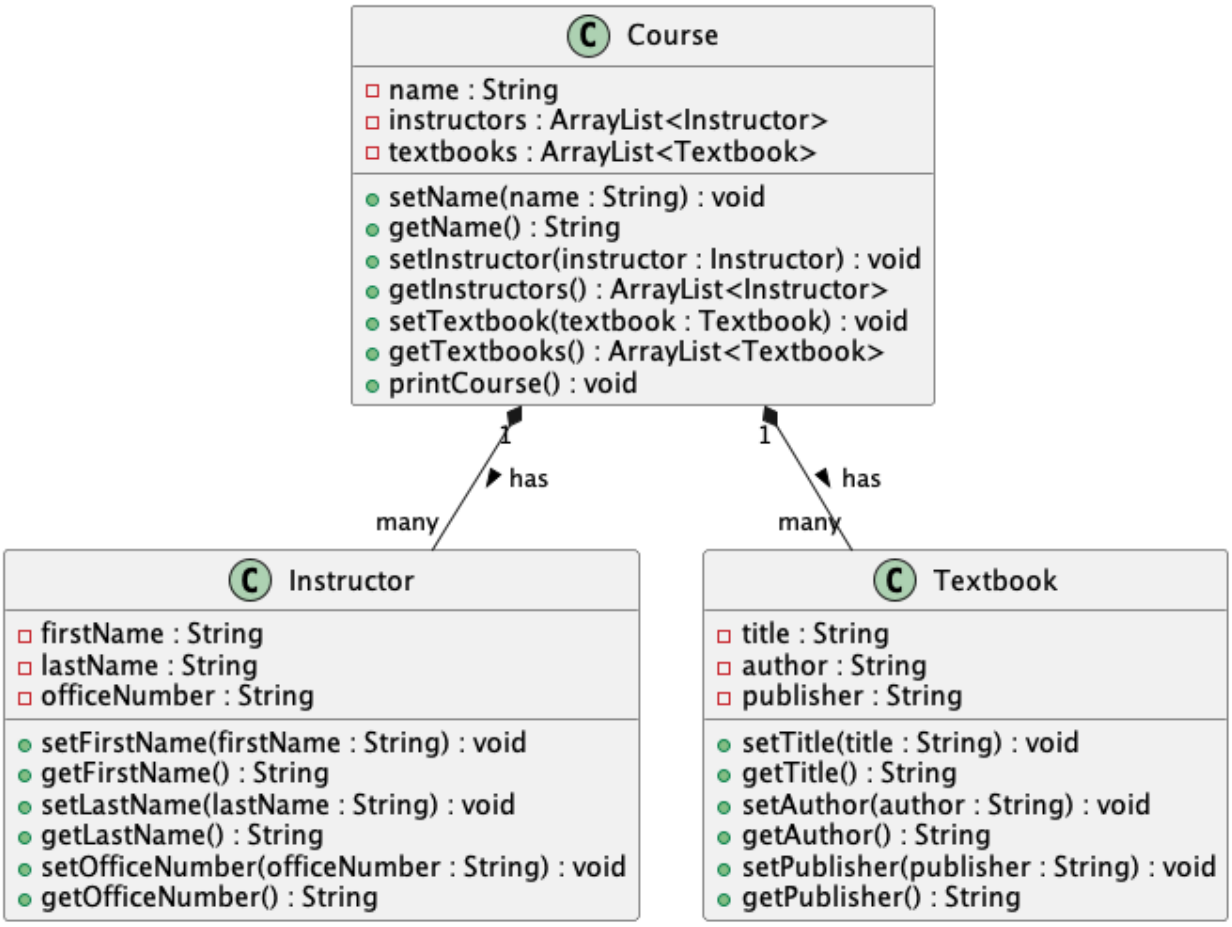


Aggregation:

```
@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:



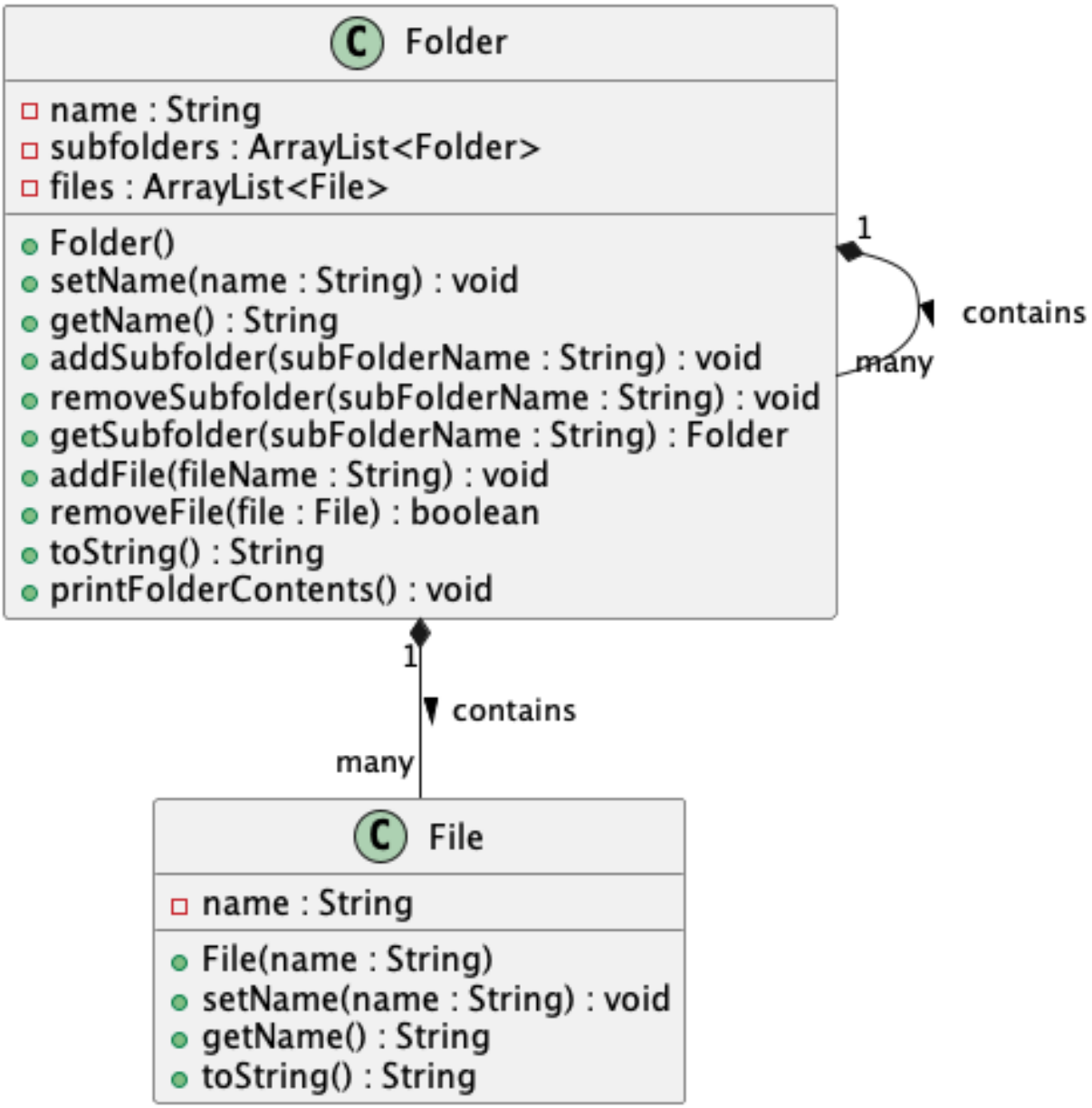
Chaitanya Ganpat Nalage
017300803

Composition:

```
@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
```

Chaitanya Ganpat Nalage
017300803

Output :



Chaitanya Ganpat Nalage
017300803

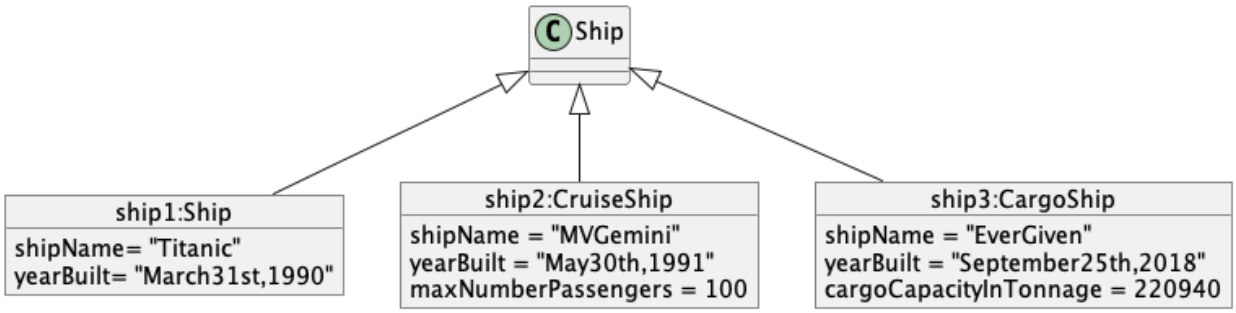
Object Diagrams:

Polymorphism:

```
@startuml
'https://plantuml.com/class-diagram

object "ship1:Ship" as s1{
    shipName= "Titanic"
    yearBuilt= "March31st,1990"
}

object "ship2:CruiseShip" as s2{
    shipName = "MVGemini"
    yearBuilt = "May30th,1991"
    maxNumberPassengers = 100
}

object "ship3:CargoShip" as s3{
    shipName = "EverGiven"
    yearBuilt = "September25th,2018"
    cargoCapacityInTonnage = 220940
}

class Ship{
}

Ship <|-- s1
Ship <|-- s2
Ship <|-- s3

@enduml
```

Chaitanya Ganpat Nalage
017300803

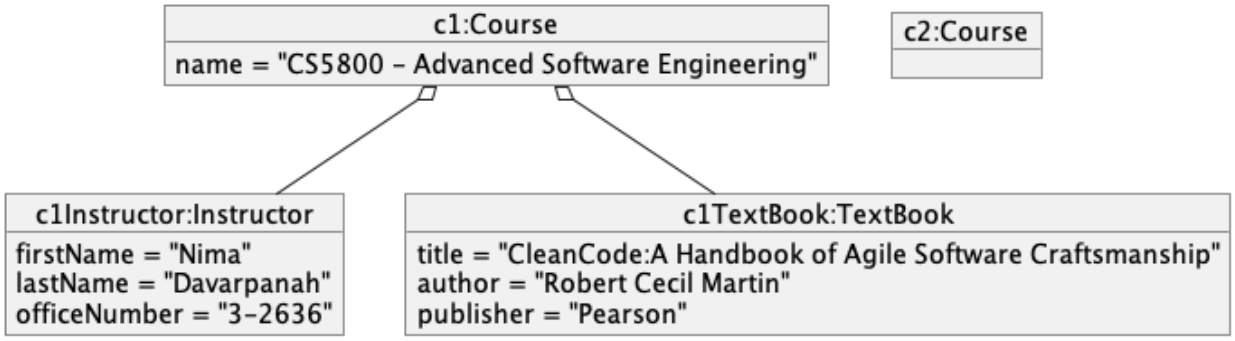
Output:



Aggregation:

```
@startuml

object "c1:Course" as c1 {
    name = "CS5800 - Advanced Software Engineering"
}
object "c2:Course" as c2 {

}

object "c1Instructor:Instructor" as c1ins{
    firstName = "Nima"
    lastName = "Davarpanah"
    officeNumber = "3-2636"
}

object "c1TextBook:TextBook" as c1tb{
    title = "CleanCode:A Handbook of Agile Software Craftsmanship"
    author = "Robert Cecil Martin"
    publisher = "Pearson"
}

c1 o-- c1ins
c1 o-- c1tb

@enduml
```

Output:


c1:Course
name = "CS5800 – Advanced Software Engineering"
c2:Course
c1Instructor:Instructor
firstName = "Nima"
lastName = "Davarpanah"
officeNumber = "3-2636"
c1TextBook:TextBook
title = "CleanCode:A Handbook of Agile Software Craftsmanship"
author = "Robert Cecil Martin"
publisher = "Pearson"

Composition:

```
@startuml

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

object "phpDemo1: Folder" as mainFolder {
  name = "php_demo1"
  files = []
  subfolders = ["Source Files", " Include Path", "Remote Files"]
}

object "Source Files: Folder" as sf1 {
  name = "Source Files"
  files = []
  subfolders = [".phalcon", "cache", "public"]
}

object  ".phalcon : Folder" as sf4 {
  name = ".phalcon"
  files = []
  subfolders = []
}

object "cache : Folder" as sf5 {
  name = "cache"
  files = []
  subfolders = []
}

object "public : Folder" as sf6 {
  name = "public"
  files = [".htaccess", ".htrouter.php", "index.html"]
  subfolders = []
}

object ".htaccess: File" as f1 {
  name = ".htaccess"
```

```
object ".htrouter.php : File" as f2 {
  name = ".htrouter.php"
}

object "index.html : File" as f3 {
  name = "index.html"
}

object "Include Path: Folder" as sf2 {
  name = "Include Path"
  files = []
  subfolders = []
}

object "Remote Files" as sf3 {
  name = "Remote Files"
  files = []
  subfolders = []
}

' Define the links representing the composition relationships
mainFolder *-- sf1
mainFolder *-- sf2
mainFolder *-- sf3

sf1 *-- sf4
sf1 *-- sf5
sf1 *-- sf6

sf6 *-- f1
sf6 *-- f2
sf6 *-- f3

@enduml
```

Chaitanya Ganpat Nalage
017300803

Output: