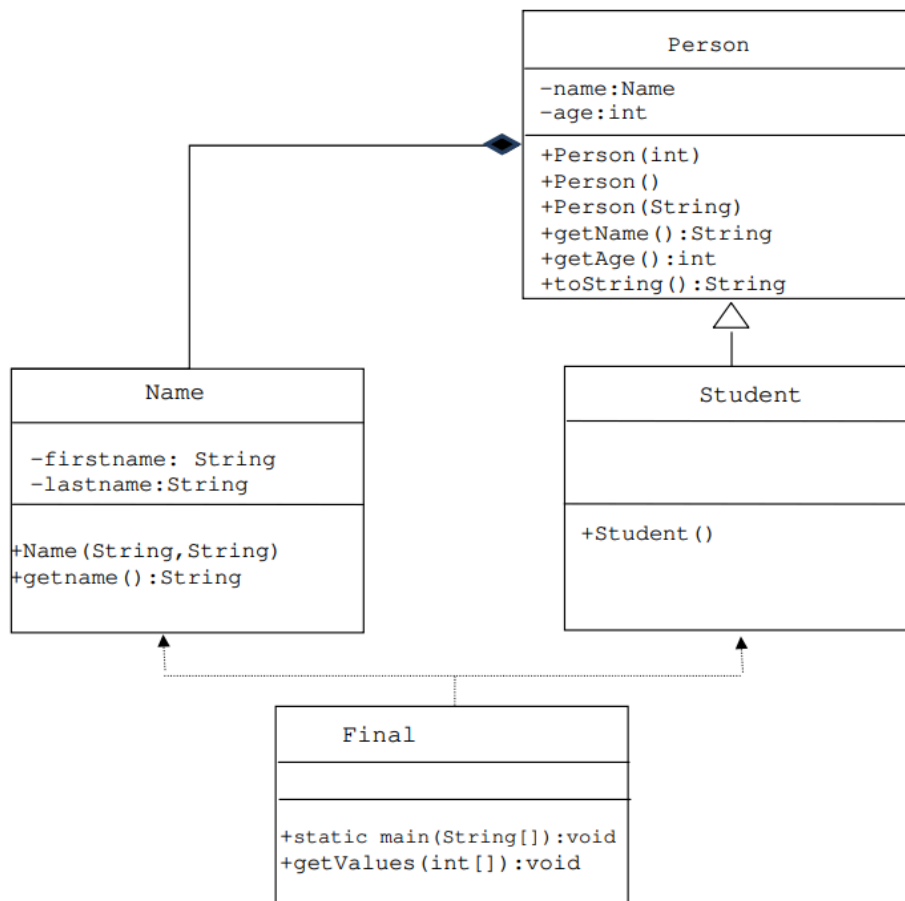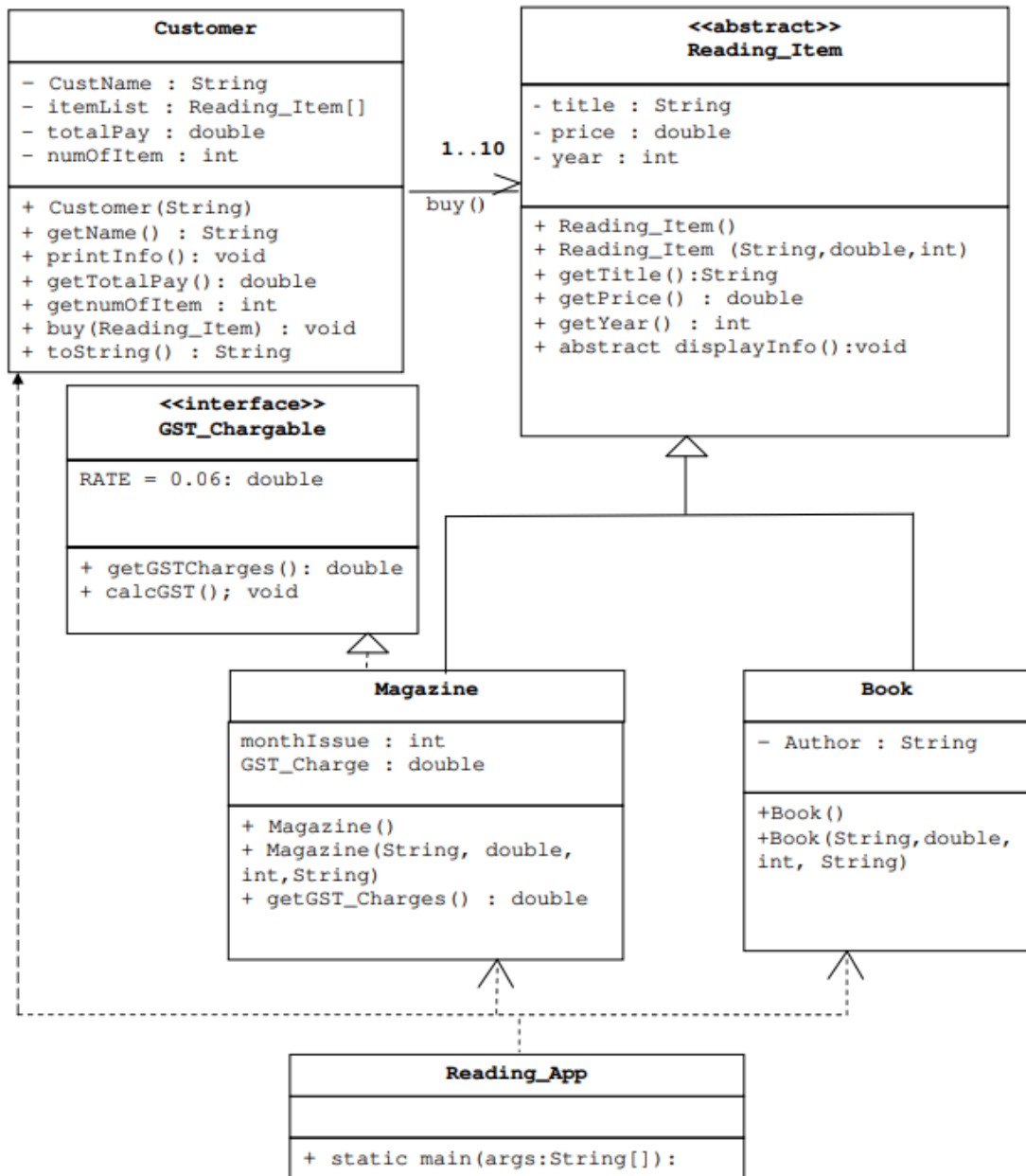# Class diagrams & OOP practice

Q1

The class diagram is given to you, you have to bring the class diagram into a C# program. Identify relevant relationships among classes and implementation should be accordingly.

```
                          ┌──────────────────────────────┐
                          │            Person            │
                          ├──────────────────────────────┤
                          │ -name:Name                   │
                          │ -age:int                     │
          ┌──────────────◆├──────────────────────────────┤
          │               │ +Person(int)                 │
          │               │ +Person()                    │
          │               │ +Person(String)              │
          │               │ +getName():String            │
          │               │ +getAge():int                │
          │               │ +toString():String           │
          │               └──────────────────────────────┘
          │                             △
          │                             │
┌─────────────────────┐      ┌──────────────────────────────┐
│        Name         │      │           Student            │
├─────────────────────┤      ├──────────────────────────────┤
│ -firstname: String  │      │                              │
│ -lastname:String    │      │                              │
├─────────────────────┤      │ +Student()                   │
│ +Name(String,String)│      │                              │
│ +getname():String   │      │                              │
└─────────────────────┘      └──────────────────────────────┘
          ▲                             ▲
          ┊                             ┊
          ┊      ┌──────────────────────────┐
          └┄┄┄┄┄┄│          Final           │┄┄┄┄┄┄┘
                 ├──────────────────────────┤
                 │                          │
                 ├──────────────────────────┤
                 │ +static main(String[]):void │
                 │ +getValues(int[]):void      │
                 └──────────────────────────┘
```

Q2

Customer

- CustName : String
- itemList : Reading_Item[]
- totalPay : double
- numOfItem : int

+ Customer(String)
+ getName() : String
+ printInfo(): void
+ getTotalPay(): double
+ getnumOfItem : int
+ buy(Reading_Item) : void
+ toString() : String

1..10
buy()

**<>
Reading_Item**

- title : String
- price : double
- year : int

+ Reading_Item()
+ Reading_Item (String,double,int)
+ getTitle():String
+ getPrice() : double
+ getYear() : int
+ abstract displayInfo():void

**<<interface>>
GST_Chargable**

RATE = 0.06: double

+ getGSTCharges(): double
+ calcGST(); void

Magazine

monthIssue : int
GST_Charge : double

+ Magazine()
+ Magazine(String, double,
int,String)
+ getGST_Charges() : double

Book

- Author : String

+Book()
+Book(String,double,
int, String)

Reading_App

+ static main(args:String[]):

Write a complete C# program named, ReadingApp.cs representing the class diagram. The program has a driver program Reading_App.cs and produces output as below.

```
List of Reading Items Bought

Item 1
Book Title: Java Programming
Book Price : RM50.0
Year Published: 2015
Book Author : Norazah
No GST Charge.

Item 2
Magazine Name: Sport Magazine
Price : RM 15.00
Year Published: 2015
Issue : May
GST Charge : RM 0.90
Total price with GST : RM 15.90

Item 3
Magazine Name: Siswa
Price : RM 8.00
Year Published: 2015
Issue : Jun
GST Charge : RM 0.48
Total price with GST : RM 8.48

YOU HAVE BOUGHT 3 ITEMS.
TOTAL Charge: RM74.38
```

Given the class diagram shown above, write the class definitions for `Reading_Item`, `Magazine` and `Book` and the interface definition for `GST_Chargable`.

The specifications for each method are as follows:

- Write the codes for all constructors with parameters (for all classes) that will initialize all attributes for the class, including the superclass, (if applicable). (Note: There will be no coding for the constructor without parameters).

- Write the codes for the abstract method `displayInfo()`that will print the title, price and year of the `Reading_Item` instances and also print the value of attributes for the subclass. The method will also print the GST charges, if any, by calling `get_GSTCharges()` and print the total price including GST (if applicable)

- Write the codes for the `calcGST()`method that will calculate the GST charges which are 6 percent of the price of the reading item and update the new price that includes GST, (by updating the `price`).

- Write the codes for `get_GSTCharges()`that will call `calcGST()`and return the GST charge.

- Write the codes for the buy(`Reading_Item rd`) method that will receive one argument of instance, update the array named itemList to include the new instance of `Reading_Item` in the array, update the number of books in the array `itemList` and print the information of the item bought (using `displayInfo()` method) and update the total payment for all books bought (including GST)

- Write the codes for toString()method to print the message to Welcome the customer and to display message: List of Reading Items Bought.

- Write the codes for printInfo() method that will display the message: YOU HAVE BOUGHT X ITEMS. and message : TOTAL Charge: RMxx.xx

How to bring Class diagrams into a code:

References:

https://java-programming.mooc.fi/part-11/1-class-diagrams
https://centria.github.io/design-and-documentation/part1/
https://www.upgrad.com/blog/what-is-composition-in-java-with-examples/