Question 1 **(35 marks)**

Figure 1 represents the **`Final.java`** program shown in Figure 2.

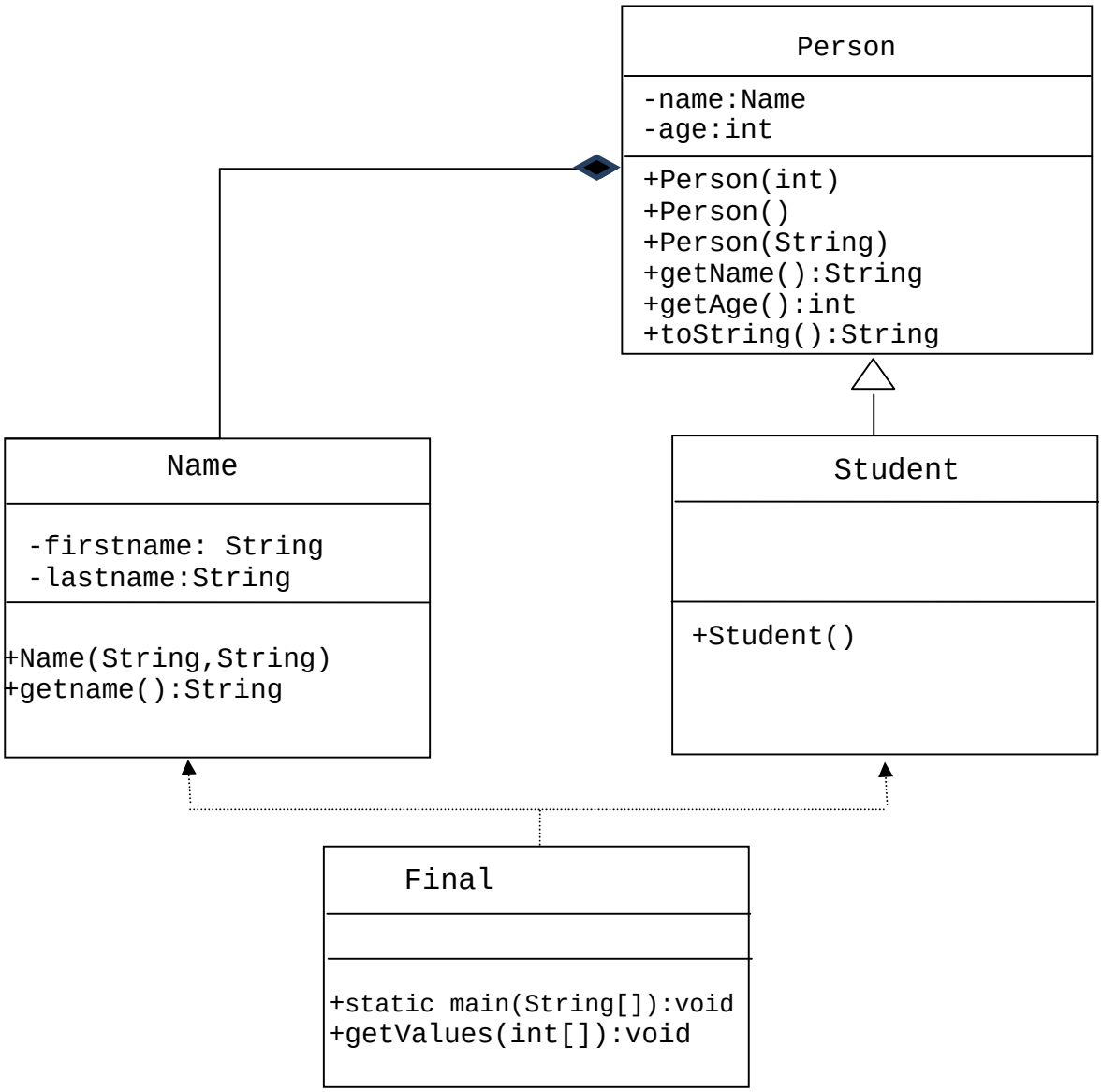

Figure 1: Class Diagram for `Final.java`.

Download java source codes (**`Final.java`**) from e-learning. Figure 2 shows the **`Final.java`** source codes for your reference. The program has errors and you are required to debug the errors. Compile and run the program. The program should produce the output, as shown in Figure 3.

Marks Distribution:

1) Finding all errors (25 marks).
2) Program able to compile and run successfully (5 marks).
3) Produce the expected output (5 marks)

```
1  import java.util.Scanner;
2  import java.util.Array;
3
4  public class Final
5  {
6     public static void main (String [] args)
7     {
8        int [5] myArray = new int [];
9        getValues(myArray);
10       System.out.print("Series of Input inserted: ");
11       System.out.print(myArray);
12
13       ArrayList<int>myArrList = new ArrayList<int>();
14       myArrList.addItem("Kluang"); //  Inserts an item
15       String str = (String)myArrList.get(0);
16       myArrList.set(1,"Muar"); // replace at index 0
17       myArrList.add("Johor Bahru");
18       System.out.println("\nItems in the arraylist: ");
19       for(int i=0;i<myArrList.size();i++)
20             System.out.println(myArrList.get(i)+" ");
21
22       Student stu = new Student("Ali");
23       Person person = new Person(10);
24       person.toString();
25    } // end main()
26
27    public static void getValues(int[] myArray)
28    {
29       Scanner kb= new Scanner(System.in);
30       System.out.println("Enter a series of " + myArray.length +"
31           numbers.");
32       for (int i=0; i<myArray.length; i++)
33       {
34             System.out.print("Enter number " + (i+1) + ":");
35             MyArray = kb.nextInt();
36       }
37    }//end getValues()
38 } //end Final
39
40 class Person
41 {     private String name;
42       private int age;
43
44       public Person(int i)
45       {
46             name= Name("Mohamad", "Ali");
```

```
47              this.age=i;
48        }
49        public Person()
50        { System.out.println("This is the superclass " +
51                                              "no-arg constructor."); }
52
53        public String Person(String arg)
54        {System.out.println("The following argument was " +
55                  " passed to the superclass constructor: " + arg);}
56
57        public Name getName()
58        { return name; }
59        public int getAge()
60        { return age; }
61        public String toString()
62        { return '\n' + name.getName() + '\n' + getAge() +'\n'; }
63
64  }// end Person
65
66  class Name
67  {
68        String firstname, lastname;
69
70        Name (String n1, String n2)
71        {
72                  firstname=n1;
73                  lastname=n2;
74        }
75
76        public String getName()
77        { return firstname+" "+lastname;}
78  } // end Name
79
80
81  class Student implements Person {
82        Student()
83        {
84            super(10.5);
85            System.out.println("This is the " + "subclass constructor.");
86        }
87  } // end Student
88
```

Figure 2: Source Codes for `Final.java`

```
Enter a series of 5 numbers.
Enter number 1:1
Enter number 2:2
Enter number 3:3
Enter number 4:4
Enter number 5:5
Series of Input inserted: 12345
Items in the arraylist:
Muar
Johor Bahru
The following argument was  passed to the superclass constructor: Super
This is the subclass constructor.
Muhamad Ali
10
```

Figure 3 : Expected Output for `Final.java`

Question 2 **[65 marks]**

Answer the question based on the UML class diagram given in Figure 4.

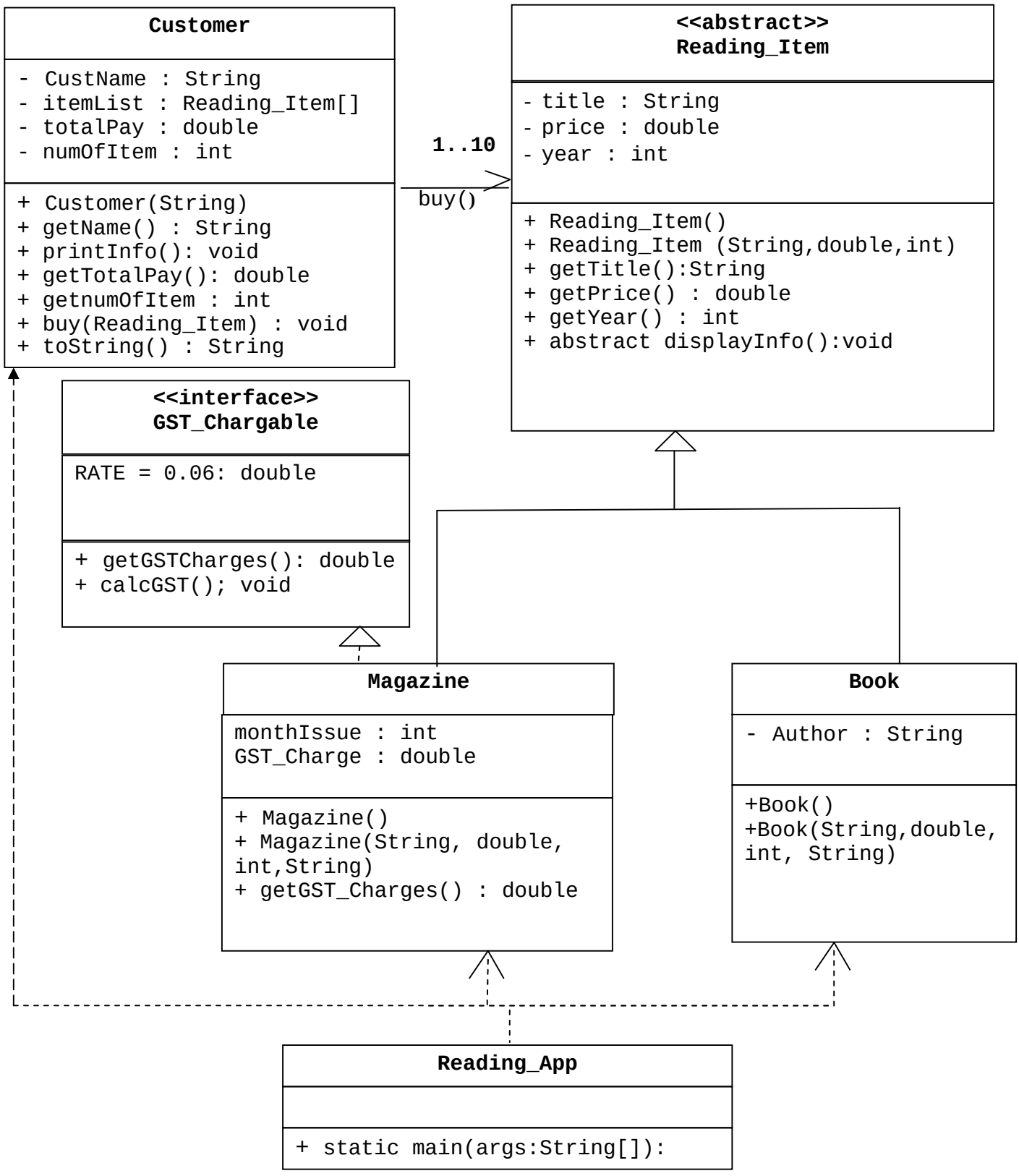

Figure 4 : UML Class Diagram

```
class Customer {
  private String CustName;
  private Reading_Item[] itemList;
  private int numOfItem = 0;
  private double totalPay = 0;
  public Customer(String n){
        CustName=n;
        itemList = new Reading_Item[10];
  }// end Customer()
  public String getName() {
          return CustName;
  } //end getName()
public double getTotalPay() {
          return totalPay;
  }
public int getNumOfItem() {
          return numOfItem;
  }
public void buy(Reading_Item rd)
{   // write codes to insert the new item bought into the array
    // update the number of items bought
    // print the information of the item bought using displayInfo()
    // update the amount of totalPayment
}

public void printInfo() {
// write codes to display number of Reading items bought
// and Total Charge for the items
}
public String toString() {
// write codes to Welcome the customer and
// display message List of Reading Items Bought
}
```

Figure 5 : Customer Class

```
public class ReadingApp {
 public static void main(String[] args) {
  Reading_Item r1 = new Book("Java Programming",50.00,2015, ”Norazah”);
  Reading_Item r2 = new Magazine("Sport Magazine”, 15.00,2015, ”May”);
  Reading_Item r3 = new Magazine("Siswa ”,8.00,2015,”June”);

     Customer c1 = new Customer(“Hilmi Hafizi”);
     System.out.println(c1.toString());
     c1.buy(r1);
     c1.buy (r2);
     c1.buy (r3);
     c1.printInfo();

  }
}
```

Figure 6 : ReadingApp Class

```
WELCOME Hilmi Hafizi
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
```

Figure 7 : Expected Output of the Program

Write a complete java program named, **`ReadingApp.java`** that represents the class diagram in Figure 4. The program has a driver program **`ReadingApp.java`** (Shown in Figure 6) and produce output as in Figure 7. Your program must fulfill the following specifications:

1. Given the class diagram shown in Figure 4, write the class definitions for **`Reading_Item`**, **`Magazine`** and **`Book`** and interface definition for **`GST_Chargable`**.

   [10 marks]

   The specifications for each methods are as follows:

2. Write the the codes for all methods to fulfill the following requirements.

   [35 marks]

   a) 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 parameter).

   b) 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)

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

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

3. Given the Customer class codes as shown in Figure 5, complete the following method specifications:

a) Write the codes for `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 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)

[10 marks]

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

[5 marks]

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

[5 marks]

Answer Question 2:

```
public void buy(Reading_Item rd)
{    itemList[numOfItem] = rd;
          numOfItem++;
     totalPay += itemList[numOfItem].getPrice();
  }
```

```
  public void printInfo() {
    System.out.println("\nWELCOME          :"+ CustName);
    System.out.println("YOU HAVE BOUGHT %d ITEMS :" + numOfItem);
         System.out.println("LIST OF ITEM(s) BOUGHT   :");
    for(int i=0;i<numOfItem;i++) {
        ReadingItem s=(ReadingItem)itemList[i];
                System.out.println("Item " + (i+1))
                s.displayInfo();
    }// end for
  }// end print()
}
```

Part C - Long Programming **30 marks**

```
import java.util.*;


interface Chargable {
   double MEMBER = 1.00;
   double NONMEMBER = 1.50;
   public void displayCharges();
}

abstract class Customer {
      private String name;
      protected double parkingCharged;
      private ParkedCar pCar;
            private int parkHour;

      public Customer() {  }

      public Customer(String name, ParkedCar p) {
            this.name = name;
            pCar = p;
      }

      public String getName() {
            return name;
      }

      public double getHour() {
            return parkHour;
      }
           public void calcParkingHour() {
            parkHour = pCar.getDuration()/60;
            if ((pCar.getDuration() % 60) > 0)
                parkHour++;
      }

      public String getCarInfo() {
            return pCar.toString()+ "\nYour parking duration " + getHour();
      }

      public abstract void calcParking();
        public abstract void displayInfo();
}


class Member extends Customer implements Chargable{
      private String memberNum;
        private double memberFee;

      public Member() {}
      public Member(String m, double f, String n, ParkedCar p) {
            super(n, p);
            memberNum = m;
                memberFee = f;
      }
      public void displayCharges(){
          System.out.println ("You will get free parking for the first two
hours");
        }
```

2 m

8 m

7 m

```
        public void calcParking() {
           calcParkingHour();
           parkingCharged = (getHour() - 2 * MEMBER);
      }

      public void displayInfo() {
             System.out.println ("Welcome: "+getName() + " You are a
member.");
             System.out.println ("Your member fee : RM"+memberFee);
             displayCharges();
             calcParking();
             System.out.println (super.getCarInfo());
            System.out.println ("Your Parking Payment: RM " + parkingCharged
+ "\n");
      }
}
class RegularCustomer extends Customer implements Chargable{

  public RegularCustomer(String n, ParkedCar p) {
     super(n, p);
  }

  public void calcParking() {
     calcParkingHour();
     if (getHour()  <= 24)
         parkingCharged = getHour()  * NONMEMBER;
     else
         parkingCharged = 50;
  }
   public void displayCharges(){
        System.out.println ("You will be charged RM 50.00 for parking more
than 20 hours.");
  }

  public void displayInfo() {
     System.out.println ("Welcome: "+getName() + " You are not a member.");
     displayCharges();
     calcParking();
     System.out.println (super.getCarInfo());
     System.out.printf ("Your Parking Payment: RM " + parkingCharged +
"\n");
      }
}



public class CustomerApp {
 public static void main(String[] args) {
     ParkedCar p1 = new ParkedCar("WWW999", 85);
     ParkedCar p2 = new ParkedCar("JQA101", 85);
     Customer c1 = new Member("MM111", 12.00, "Mr. Lim", p1);
     Customer c2 = new RegularCustomer("Sarah Ali", p2);
     c1.displayInfo();
     c2.displayInfo();
  }
}
```

8 m

Answer Question 1:

```
import java.util.Scanner;
```

```
import java.util.ArrayList; //1

public class ErrorDebugbetul
{
public static void main (String [] args) throws MyException //2
{
           int [] myArray = new int [5];  //3
           getValues(myArray);
           for (int i=0; i<myArray.length; i++)
              System.out.print(myArray[i]);//4

           ArrayList myArrList = new ArrayList();
           myArrList.add("Kluang"); // 5 Inserts an item
           String str = (String)myArrList.get(0);
           myArrList.set(0, "Muar"); //6
           for(int i=0;i<myArrList.size();i++)
                System.out.println(myArrList.get(i)+" ");


           Student stu = new Student(); //7
}

public static void getValues(int[] myArray)
{
     Scanner kb= new Scanner(System.in);
     System.out.println("Enter a series of " + myArray.length +"
numbers.");
     for (int i=0; i<myArray.length; i++)
{
           System.out.print("Enter number " + (i+1) + ":");
           myArray[i] = kb.nextInt();  //8..9
     }
}
}

class Person
{
     private Name name;
     private int age;

public Person(int i)
     {
           name= new Name("Mohamad", "Ali"); //10
           this.age=i;
     }
     public Person()
{ System.out.println("This is the superclass " +"no-arg
constructor."); }

public Person(String arg) //11
{System.out.println("The following argument was" +"passed to the
superclass constructor: " + arg);}


public Name getName()
     { return name; }
```

```
      public int getAge()
      { return age; }
      public String toString()
      { return '\n' + name.getName() + '\n' + getAge() +'\n'; } //12

}

class Name
{
      String firstname, lastname;

      Name (String n1, String n2)
{
          firstname=n1;
          lastname=n2;
}

      public String getName()
{ return firstname+" "+lastname;}
}


class Student extends Person { //13
   Student() {
super("Super"); //14
System.out.println("This is the " + "subclass constructor.");
}
}

class MyException extends Exception {
//System.out.println("This is my exception.");
} //15
```