Object-Oriented Design Lab Report (Java 2)

Bisakh Mondal 001810501079

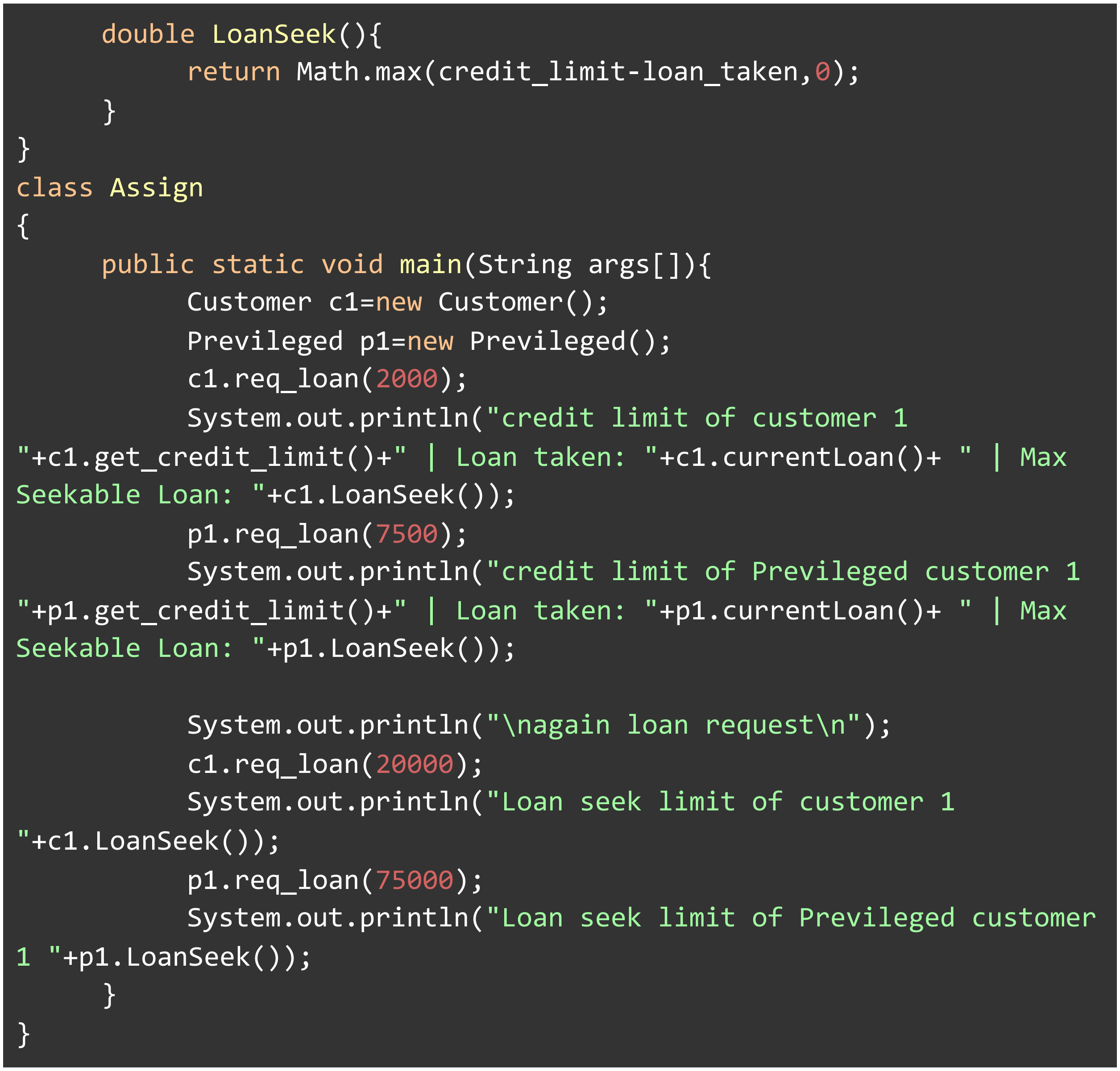
Format: Approach(if notable) | Code | Output

Java Assignment

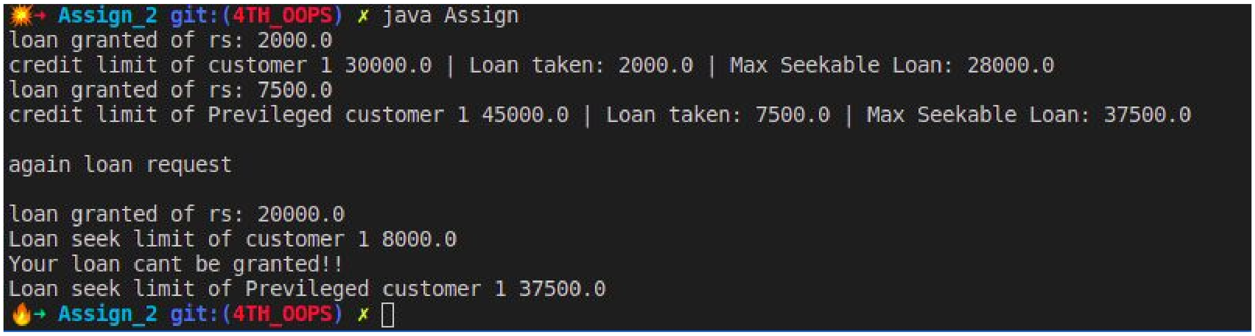
(Q1)

Inheriance in java is nice. Credit limit has been declared **static**, Previledged & customer both has credit limit field but variables in java are not **polymorphic** they doesn’t override each other and is not specified in function i.e. no overloaded function the base class variables is assigned during compile time.

```
import java.util.*;
class Customer{
    public String name,ph;
    public int id;
    public double loan_taken;
    static double credit_limit=30000;
    static int id_gen=123456;
    Customer(){
        name="";
        ph="";
        id=id_gen + (int)(Math.random()*100);
        loan_taken=0.0;
    }
//  public static void change_credit(double ne){
//      credit_limit=ne;
//  }
    Public static double get_credit_limit(){
        return credit_limit;
    }
    double currentLoan(){
        return loan_taken;
    }
    double LoanSeek(){
        return Math.max(credit_limit-loan_taken,0);
    }
    void setName(String name){
        this.name=name;
    }
    void setPhone(String phone){
        this.ph=phone;
    }
    public void req_loan(double ammount){
        if(ammount+loan_taken>credit_limit){
            System.out.println("Your loan cant be granted!!");
        }
        else{
            loan_taken+=ammount;
            System.out.println("loan granted of rs: "+ ammount);
        }

    }

}
class Previleged extends Customer{
    //due to no polymorphic nature in data variable.
    static double credit_limit=45000;
    public double get_credit_limit(){
        return credit_limit;
    }

    public void req_loan(double ammount){
        if(ammount+loan_taken>credit_limit){
            System.out.println("Your loan cant be granted!!");
        }
        else{
            loan_taken+=ammount;
            System.out.println("loan granted of rs: "+ ammount);
        }

    }
```



```
	double LoanSeek(){
		return Math.max(credit_limit-loan_taken,0);
	}
}
class Assign
{
	public static void main(String args[]){
		Customer c1=new Customer();
		Previleged p1=new Previleged();
		c1.req_loan(2000);
		System.out.println("credit limit of customer 1
"+c1.get_credit_limit()+" | Loan taken: "+c1.currentLoan()+ " | Max
Seekable Loan: "+c1.LoanSeek());
		p1.req_loan(7500);
		System.out.println("credit limit of Previleged customer 1
"+p1.get_credit_limit()+" | Loan taken: "+p1.currentLoan()+ " | Max
Seekable Loan: "+p1.LoanSeek());

		System.out.println("\nagain loan request\n");
		c1.req_loan(20000);
		System.out.println("Loan seek limit of customer 1
"+c1.LoanSeek());
		p1.req_loan(75000);
		System.out.println("Loan seek limit of Previleged customer
1 "+p1.LoanSeek());
	}
}
```

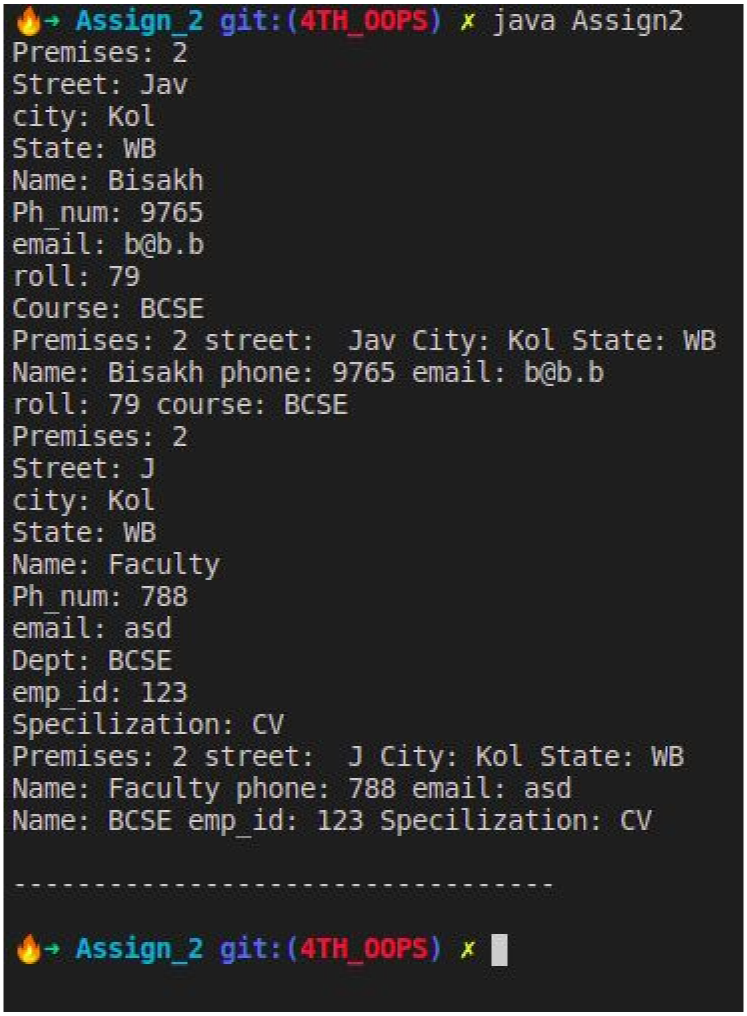


```
💥→ Assign_2 git:(4TH_OOPS) ✗ java Assign
loan granted of rs: 2000.0
credit limit of customer 1 30000.0 | Loan taken: 2000.0 | Max Seekable Loan: 28000.0
loan granted of rs: 7500.0
credit limit of Previleged customer 1 45000.0 | Loan taken: 7500.0 | Max Seekable Loan: 37500.0

again loan request

loan granted of rs: 20000.0
Loan seek limit of customer 1 8000.0
Your loan cant be granted!!
Loan seek limit of Previleged customer 1 37500.0
🔥→ Assign_2 git:(4TH_OOPS) ✗ ▯
```

(Q2)

Here it’s another example of **inheritance** along with the concept of **Abstract classes.**

```
import java.util.*;
class Address{
        private int pNum;
        private String street,city,pin,state;
        Address(){
                pNum=0;
                street="";
                city="";
                pin="";
                state="";
        }
        public void getAddress(){
                Scanner in=new Scanner(System.in);
                System.out.print("Premises: ");
                pNum=in.nextInt();
                System.out.print("Street: ");
                street=in.next();
                System.out.print("city: ");
                city=in.next();
                System.out.print("State: ");
                state=in.next();

        }
        public void print(){
                System.out.println("Premises: "+pNum+" street:  "+street+" City: "+city+" State: "+state);
        }

}


abstract class Individual{
        public String name,ph_num,email;
        Address a;
        Individual(){
            a=new Address();
      }
      public void setInput(){
            a.getAddress();
            Scanner in=new Scanner(System.in);
            System.out.print("Name: ");
            name=in.nextLine();
            System.out.print("Ph_num: ");
            ph_num=in.next();
            System.out.print("email: ");
            email=in.next();
      }
      public void print(){
            a.print();
            System.out.println("Name: "+name+" phone: "+ph_num+" email: "+email);
      }
}

class Student extends Individual{
      String roll,course;
      public void set(){
            super.setInput();
            Scanner in=new Scanner(System.in);
            System.out.print("roll: ");
            roll=in.next();
            System.out.print("Course: ");
            course=in.next();
      }
      public void print(){
            super.print();
            System.out.println("roll: "+roll+" course: "+course);
      }

}

class faculty extends Individual{
      String dept,employee_id,specilization;
      public void set(){
           this.setInput();
           Scanner in=new Scanner(System.in);
           System.out.print("Dept: ");
           dept=in.next();
           System.out.print("emp_id: ");
           employee_id=in.next();
           System.out.print("Specilization: ");
           specilization=in.next();
      }
      public void print(){
           super.print();
           System.out.println("Name: "+dept+" emp_id: "+employee_id+"
Specilization: "+specilization);

System.out.println("\n----------------------------------\n");
      }

}
class Assign2{
      public static void main(String args[]){
           Student s=new Student();
           s.set();
           s.print();
           faculty f=new faculty();
           f.set();
           f.print();
      }
}
```



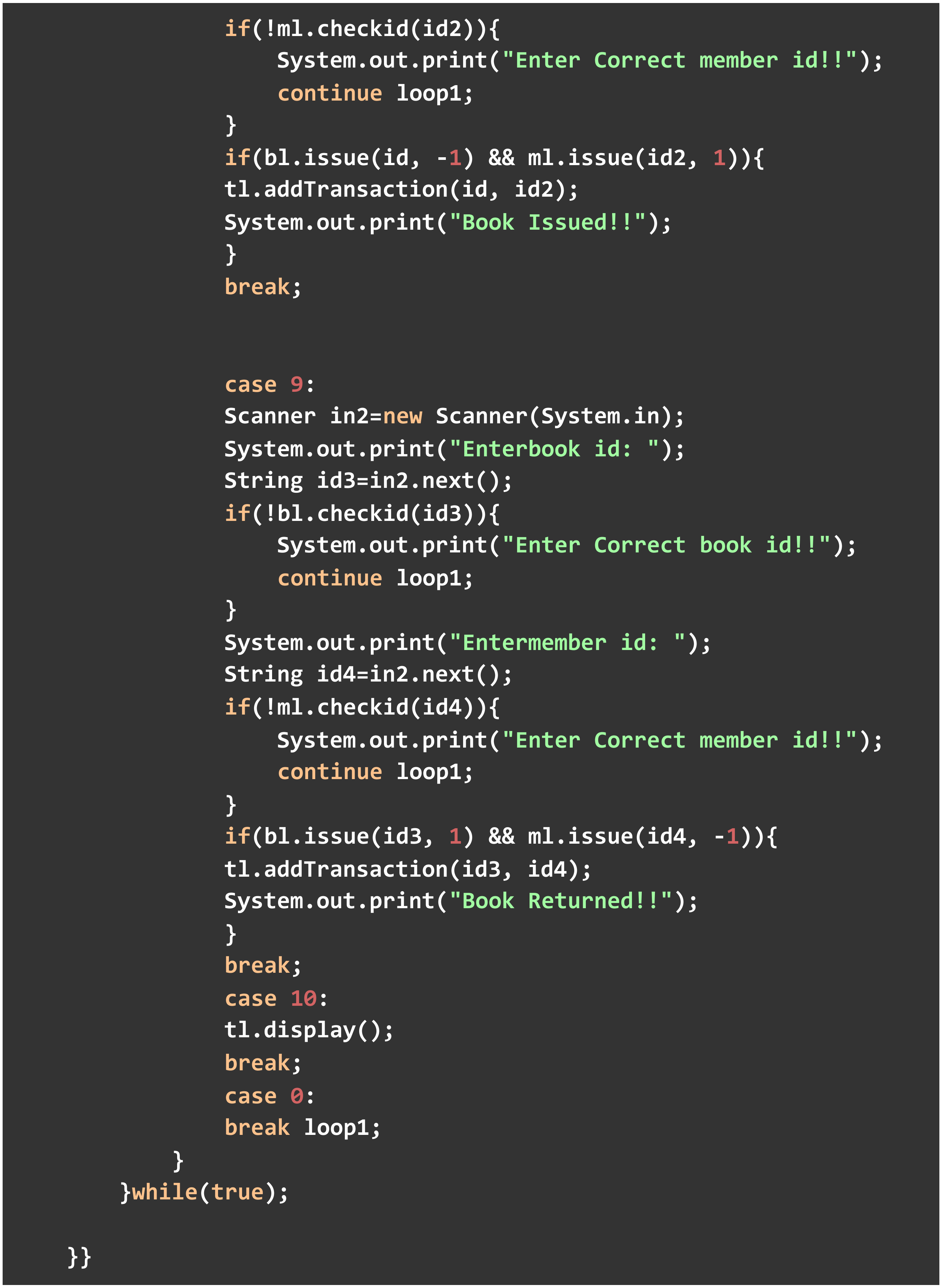
```
🔥→ Assign_2 git:(4TH_OOPS) ✗ java Assign2
Premises: 2
Street: Jav
city: Kol
State: WB
Name: Bisakh
Ph_num: 9765
email: b@b.b
roll: 79
Course: BCSE
Premises: 2 street:  Jav City: Kol State: WB
Name: Bisakh phone: 9765 email: b@b.b
roll: 79 course: BCSE
Premises: 2
Street: J
city: Kol
State: WB
Name: Faculty
Ph_num: 788
email: asd
Dept: BCSE
emp_id: 123
Specilization: CV
Premises: 2 street:  J City: Kol State: WB
Name: Faculty phone: 788 email: asd
Name: BCSE emp_id: 123 Specilization: CV

----------------------------------

🔥→ Assign_2 git:(4TH_OOPS) ✗ 
```

(Q3)

Assign2_3.java

```
import library.book.*;
import library.member.*;
import library.transaction.*;
import java.util.*;
public class Assign2_3{
    public static void main(String[] args) {
        Booklist bl = new Booklist();
        Memberlist ml=new Memberlist();
        Transactionlist tl=new Transactionlist();
       loop1: do{
```

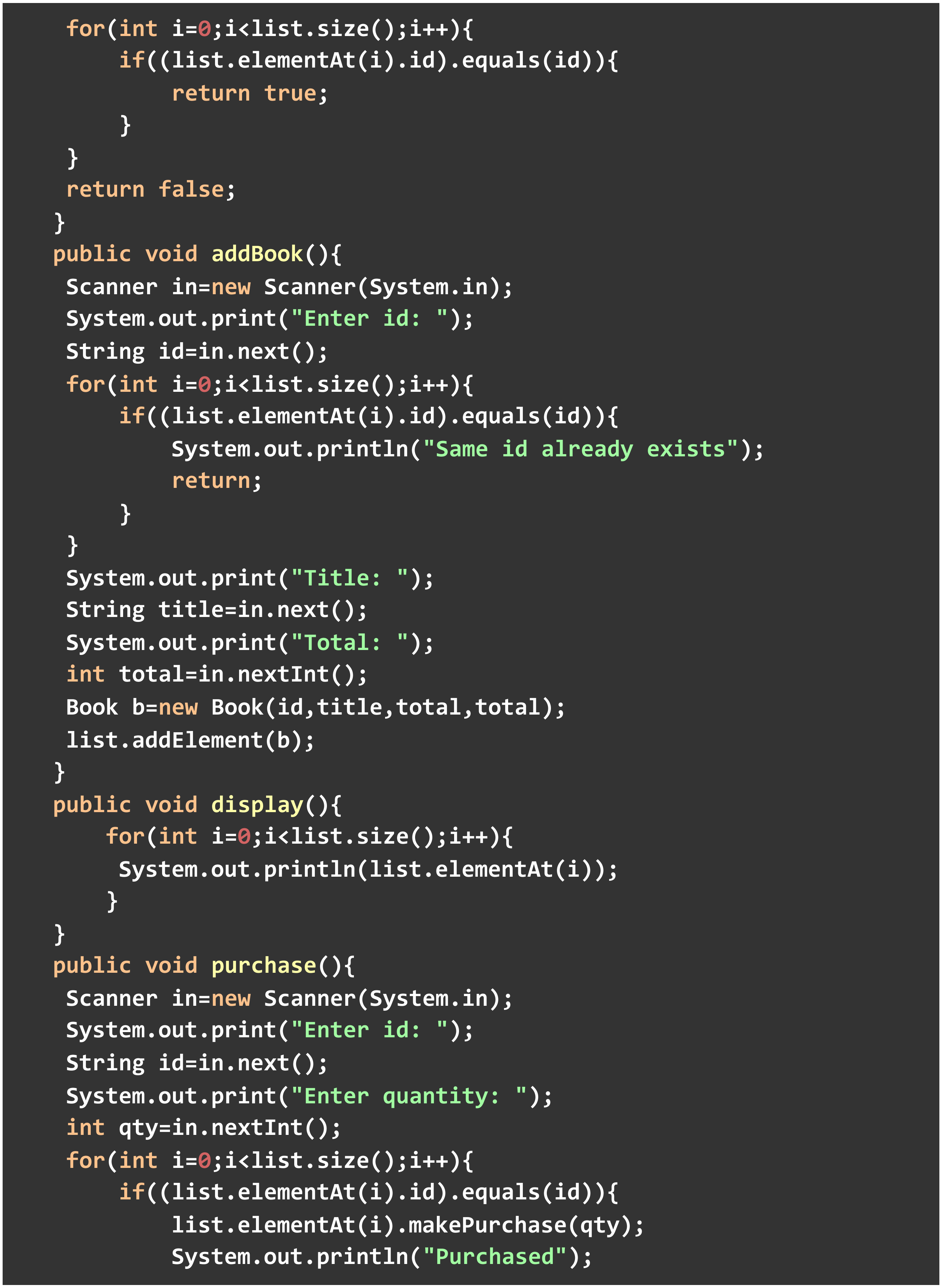
```
System.out.println("\n1.Add_book\n2.Add_quantity\n3.Display all\n4.Display partricular\n5:Add Member\n6:Display all member\n7:Display Particular member\n8:Issue a book\n9:Return a book\n10:Transaction History\n0.Exit\n");
            Scanner in=new Scanner(System.in);
            System.out.print("Enter choice: ");
            int choice=in.nextInt();
            switch (choice) {
                case 1:
                    bl.addBook();
                    break;
                case 2:
                    bl.purchase();
                    break;
                case 3:
                bl.display();
                break;
                case 4:
                bl.show();
                break;
                case 5:
                ml.addMember();
                break;
                case 6:
                ml.display();
                break;
                case 7:
                ml.show();
                break;
                case 8:
                Scanner in1=new Scanner(System.in);
                System.out.print("Enterbook id: ");
                String id=in1.next();
                if(!bl.checkid(id)){
                    System.out.print("Enter Correct book id!!");
                    continue loop1;
                }
                System.out.print("Entermember id: ");
                String id2=in1.next();
```



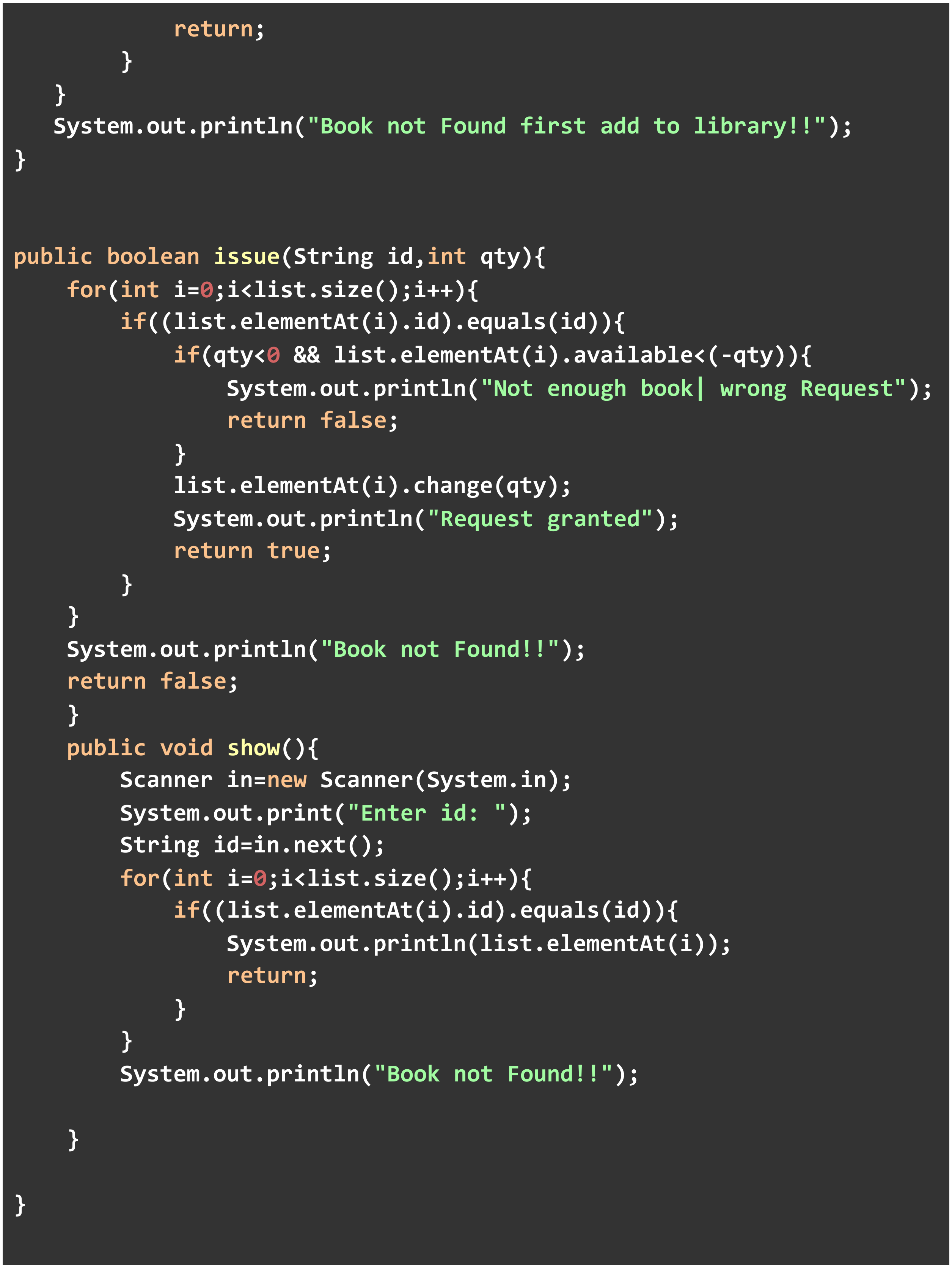
```
                if(!ml.checkid(id2)){
                    System.out.print("Enter Correct member id!!");
                    continue loop1;
                }
                if(bl.issue(id, -1) && ml.issue(id2, 1)){
                tl.addTransaction(id, id2);
                System.out.print("Book Issued!!");
                }
                break;


                case 9:
                Scanner in2=new Scanner(System.in);
                System.out.print("Enterbook id: ");
                String id3=in2.next();
                if(!bl.checkid(id3)){
                    System.out.print("Enter Correct book id!!");
                    continue loop1;
                }
                System.out.print("Entermember id: ");
                String id4=in2.next();
                if(!ml.checkid(id4)){
                    System.out.print("Enter Correct member id!!");
                    continue loop1;
                }
                if(bl.issue(id3, 1) && ml.issue(id4, -1)){
                tl.addTransaction(id3, id4);
                System.out.print("Book Returned!!");
                }
                break;
                case 10:
                tl.display();
                break;
                case 0:
                break loop1;
            }
        }while(true);

    }}
```

Booklist.java

```
package library.book;
import java.util.Vector;
import java.util.*;

public class Booklist{
   class Book{
       String id,title;
       int available,total;
       Book(String iid,String titlee,int totall,int availablee ){
        id=iid;
        title=titlee;
        available=availablee;
        total=totall;
    }
       public void add(String iid,String titlee,int totall,int
availablee ){
           id=iid;
           title=titlee;
           available=availablee;
           total=totall;
       }
       public void makePurchase(int val){
           total+=val;
           available+=val;
       }
       public void change(int val){
           available+=val;
       }
       public String toString(){
           return "Book: "+title+" |id: "+id+" |total_copy_purchased:
"+total+" |available: "+available;
       }
   }
   Vector<Book> list;
   public Booklist(){
       list=new Vector<Book>();
   }
   public boolean checkid(String id){
```



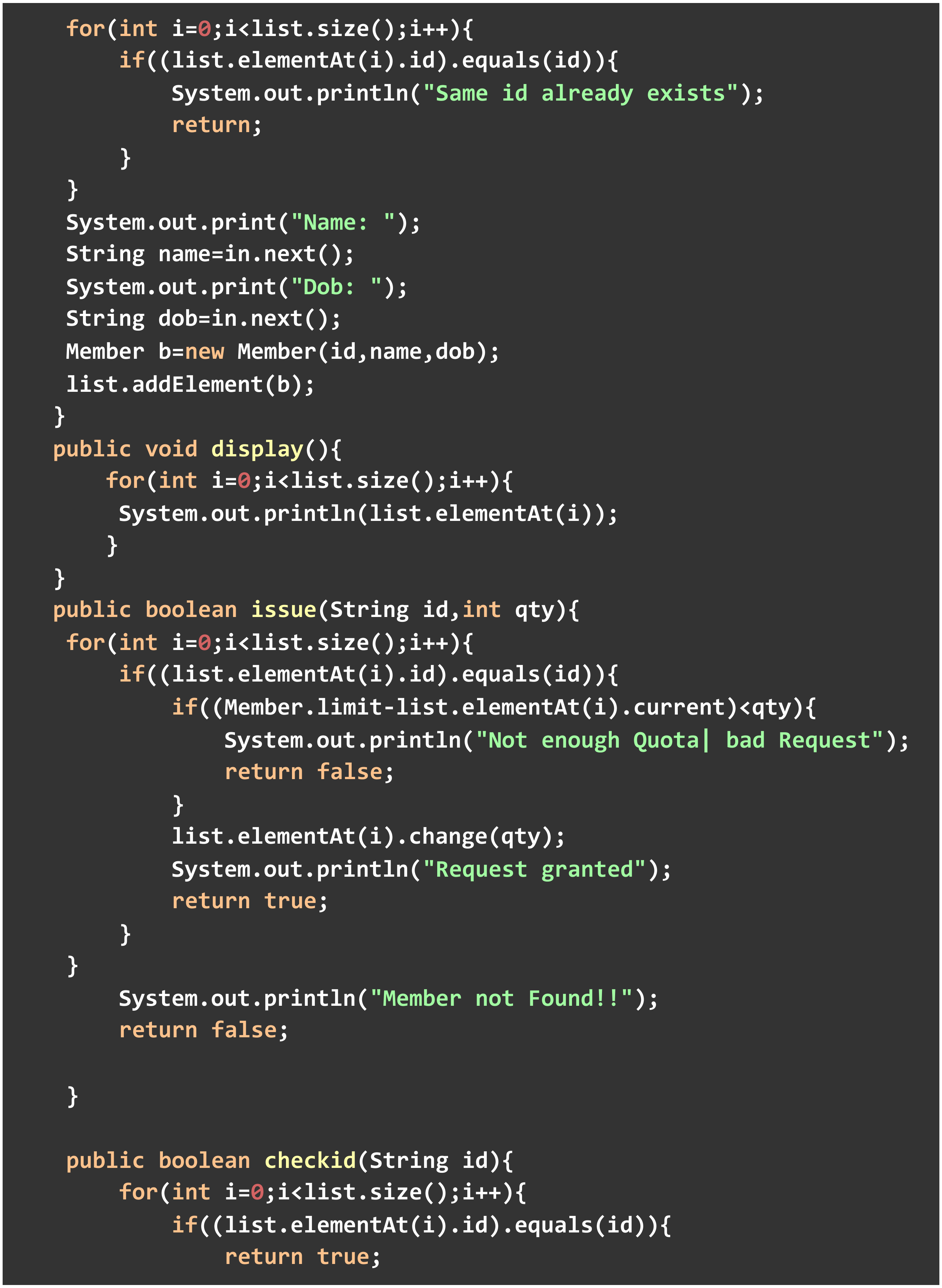
```
 for(int i=0;i<list.size();i++){
     if((list.elementAt(i).id).equals(id)){
         return true;
     }
 }
 return false;
}
public void addBook(){
 Scanner in=new Scanner(System.in);
 System.out.print("Enter id: ");
 String id=in.next();
 for(int i=0;i<list.size();i++){
     if((list.elementAt(i).id).equals(id)){
         System.out.println("Same id already exists");
         return;
     }
 }
 System.out.print("Title: ");
 String title=in.next();
 System.out.print("Total: ");
 int total=in.nextInt();
 Book b=new Book(id,title,total,total);
 list.addElement(b);
}
public void display(){
    for(int i=0;i<list.size();i++){
     System.out.println(list.elementAt(i));
    }
}
public void purchase(){
 Scanner in=new Scanner(System.in);
 System.out.print("Enter id: ");
 String id=in.next();
 System.out.print("Enter quantity: ");
 int qty=in.nextInt();
 for(int i=0;i<list.size();i++){
     if((list.elementAt(i).id).equals(id)){
         list.elementAt(i).makePurchase(qty);
         System.out.println("Purchased");
```



```
            return;
        }
   }
   System.out.println("Book not Found first add to library!!");
}


public boolean issue(String id,int qty){
    for(int i=0;i<list.size();i++){
        if((list.elementAt(i).id).equals(id)){
            if(qty<0 && list.elementAt(i).available<(-qty)){
                System.out.println("Not enough book| wrong Request");
                return false;
            }
            list.elementAt(i).change(qty);
            System.out.println("Request granted");
            return true;
        }
    }
    System.out.println("Book not Found!!");
    return false;
    }
    public void show(){
        Scanner in=new Scanner(System.in);
        System.out.print("Enter id: ");
        String id=in.next();
        for(int i=0;i<list.size();i++){
            if((list.elementAt(i).id).equals(id)){
                System.out.println(list.elementAt(i));
                return;
            }
        }
        System.out.println("Book not Found!!");

    }

}
```

Memberlist.java

```
package library.member;
import java.util.Vector;
import java.util.*;

public class Memberlist{
   class Member{
       String id,name,dob;
        static final int limit = 6;
       int current;
       Member(String iid,String namee,String dobb  ){
        id=iid;
        name=namee;
        dob=dobb;
        current=0;
    }
       public void add(String iid,String namee,String dobb  ){
        id=iid;
        name=namee;
        dob=dobb;
        current=0;
    }
       public void change(int val){
           current+=val;
       }
       public String toString(){
           return "Member: "+name+" |id: "+id+" |total-issued: 
"+current;
       }
   }
   Vector<Member> list;
   public Memberlist(){
       list=new Vector<Member>();
   }
   public void addMember(){
    Scanner in=new Scanner(System.in);
    System.out.print("Enter id: ");
    String id=in.next();
```

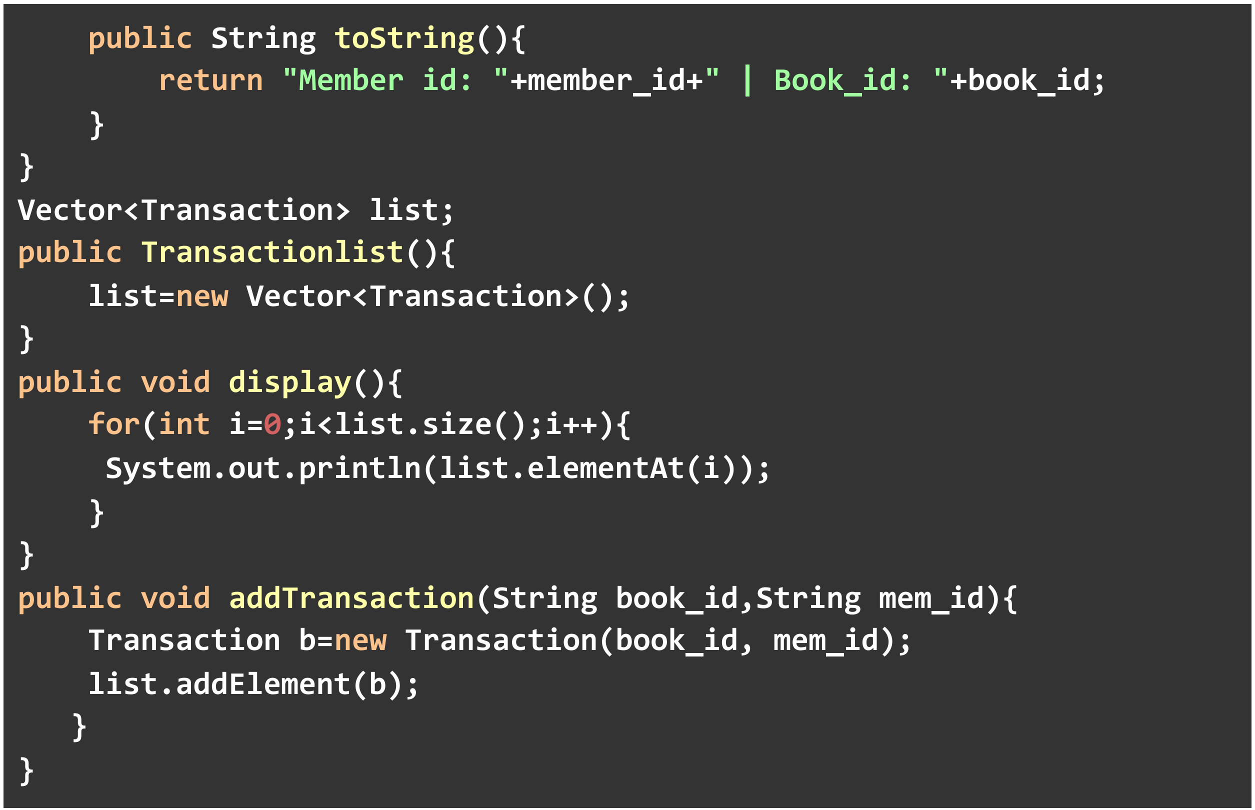


```
 for(int i=0;i<list.size();i++){
     if((list.elementAt(i).id).equals(id)){
         System.out.println("Same id already exists");
         return;
     }
 }
 System.out.print("Name: ");
 String name=in.next();
 System.out.print("Dob: ");
 String dob=in.next();
 Member b=new Member(id,name,dob);
 list.addElement(b);
}
public void display(){
    for(int i=0;i<list.size();i++){
     System.out.println(list.elementAt(i));
    }
}
public boolean issue(String id,int qty){
 for(int i=0;i<list.size();i++){
     if((list.elementAt(i).id).equals(id)){
         if((Member.limit-list.elementAt(i).current)<qty){
             System.out.println("Not enough Quota| bad Request");
             return false;
         }
         list.elementAt(i).change(qty);
         System.out.println("Request granted");
         return true;
     }
 }
     System.out.println("Member not Found!!");
     return false;

 }

 public boolean checkid(String id){
     for(int i=0;i<list.size();i++){
         if((list.elementAt(i).id).equals(id)){
             return true;
```

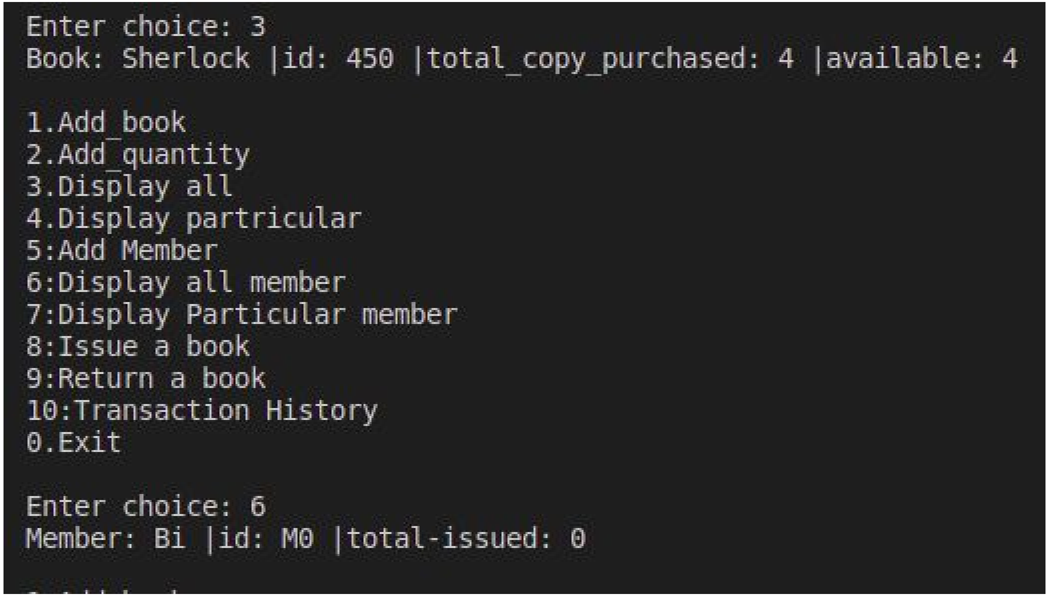
```
            }
        }
        return false;
       }

    public void show(){
        Scanner in=new Scanner(System.in);
        System.out.print("Enter id: ");
        String id=in.next();
        for(int i=0;i<list.size();i++){
            if((list.elementAt(i).id).equals(id)){
                System.out.println(list.elementAt(i));
                return;
            }
        }
        System.out.println("Member not Found!!");

    }

}
```

Transaction.java

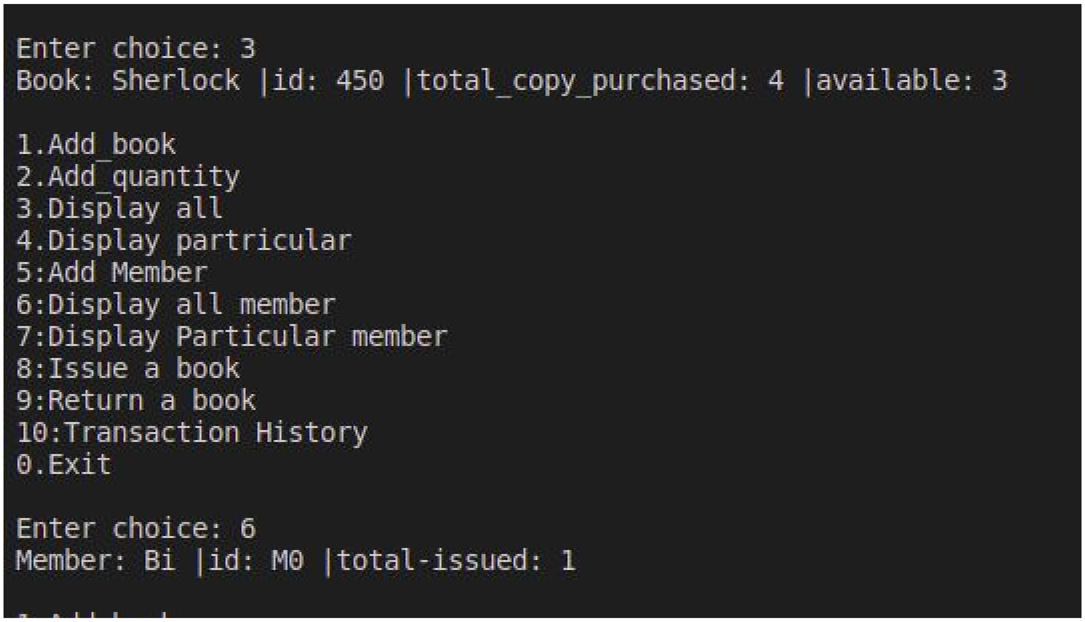
```
package library.transaction;
import java.util.Vector;
import java.util.*;

public class Transactionlist{
   class Transaction{
       String book_id,member_id;
       Transaction(String book_id,String member_id){
           this.book_id=book_id;
           this.member_id=member_id;
       }
       public void add(String book_id,String member_id){
        this.book_id=book_id;
        this.member_id=member_id;
    }
```



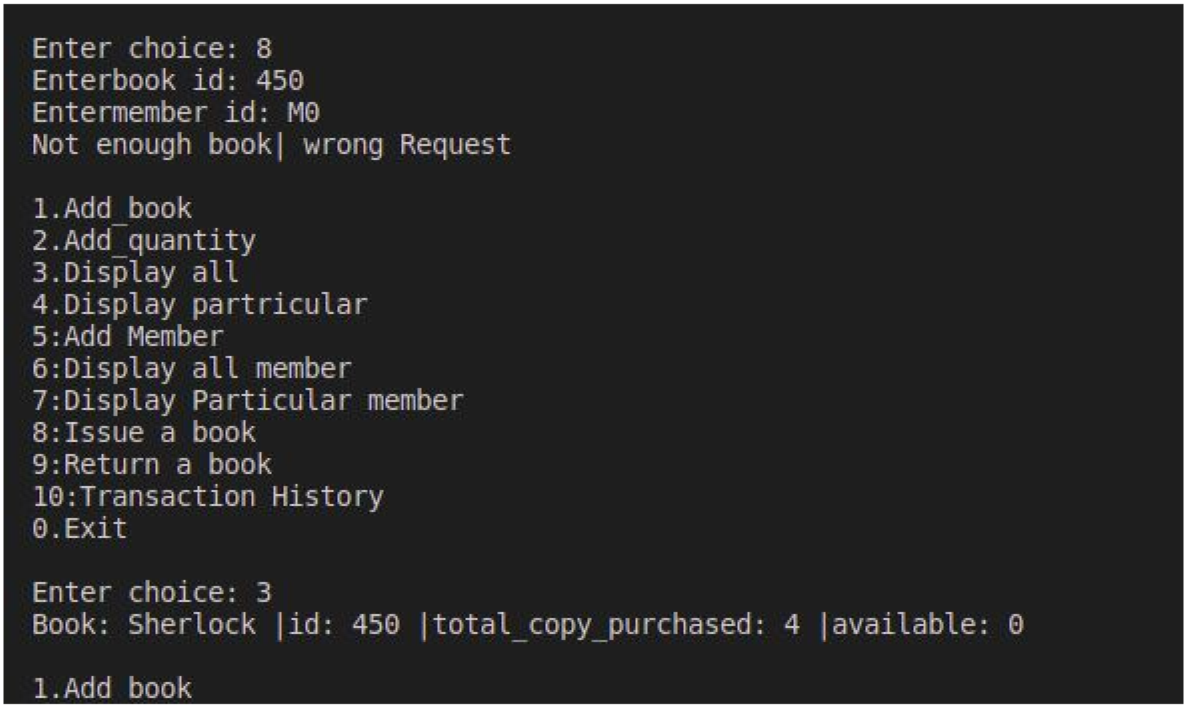
```
    public String toString(){
        return "Member id: "+member_id+" | Book_id: "+book_id;
    }
}
Vector<Transaction> list;
public Transactionlist(){
    list=new Vector<Transaction>();
}
public void display(){
    for(int i=0;i<list.size();i++){
     System.out.println(list.elementAt(i));
    }
}
public void addTransaction(String book_id,String mem_id){
    Transaction b=new Transaction(book_id, mem_id);
    list.addElement(b);
   }
}
```



```
Enter choice: 3
Book: Sherlock |id: 450 |total_copy_purchased: 4 |available: 4

1.Add_book
2.Add_quantity
3.Display all
4.Display partricular
5:Add Member
6:Display all member
7:Display Particular member
8:Issue a book
9:Return a book
10:Transaction History
0.Exit

Enter choice: 6
Member: Bi |id: M0 |total-issued: 0
```



```
Enter choice: 3
Book: Sherlock |id: 450 |total_copy_purchased: 4 |available: 3

1.Add_book
2.Add_quantity
3.Display all
4.Display partricular
5:Add Member
6:Display all member
7:Display Particular member
8:Issue a book
9:Return a book
10:Transaction History
0.Exit

Enter choice: 6
Member: Bi |id: M0 |total-issued: 1

1.Add_book
```

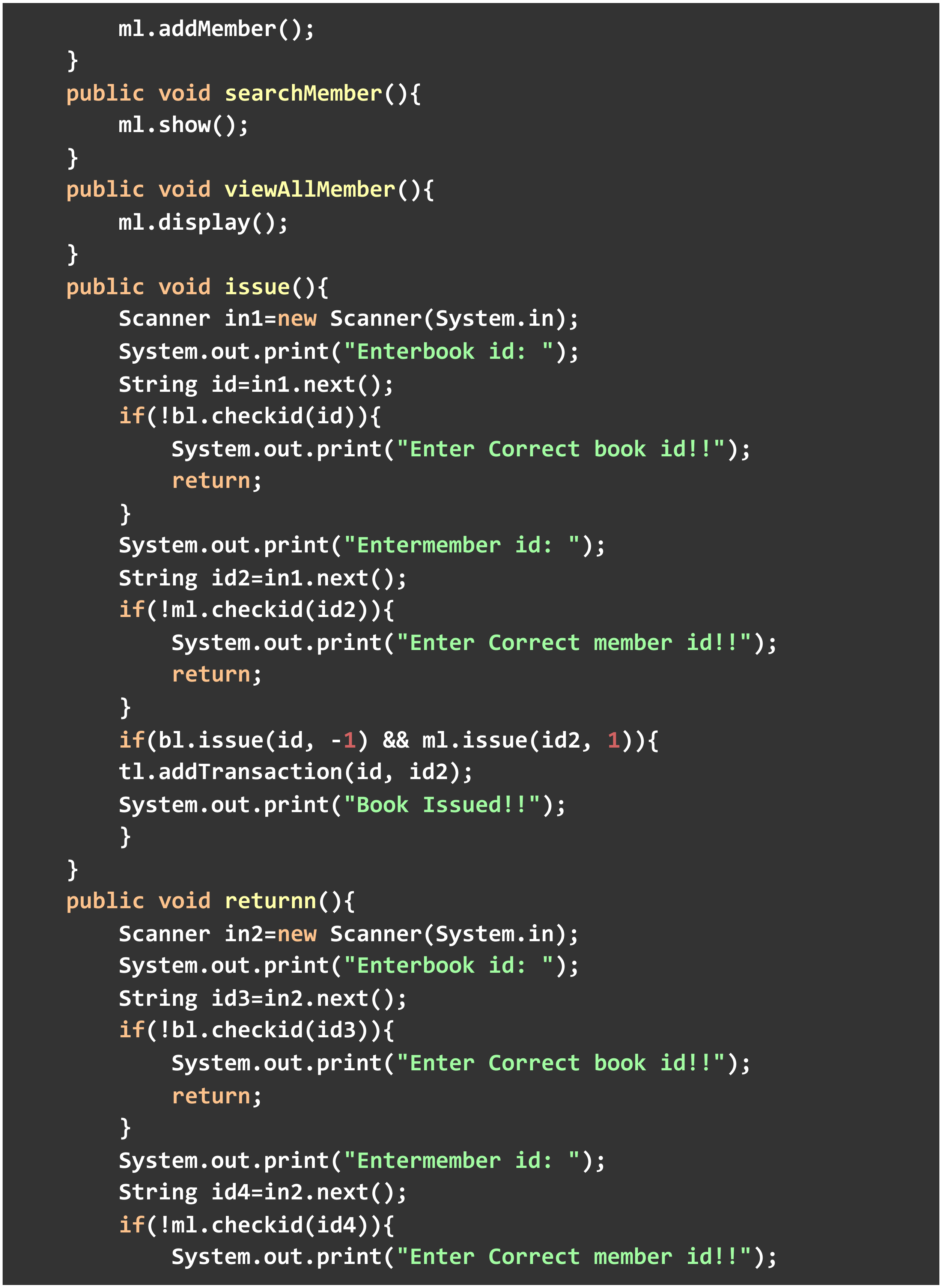


```
Enter choice: 8
Enterbook id: 450
Entermember id: M0
Not enough book| wrong Request

1.Add_book
2.Add_quantity
3.Display all
4.Display partricular
5:Add Member
6:Display all member
7:Display Particular member
8:Issue a book
9:Return a book
10:Transaction History
0.Exit

Enter choice: 3
Book: Sherlock |id: 450 |total_copy_purchased: 4 |available: 0

1.Add_book
```

(Q4)

Just a **Refactoring** of Q3 with an interface of LibraryFeatures and the same output.

```java
/**
 * Assign2.4
 */
import library.book.*;
import library.member.*;
import library.transaction.*;
import java.util.*;

interface LibraryFeatures  {
    static final Booklist bl = new Booklist();
    static final Memberlist ml=new Memberlist();
    static final Transactionlist tl=new Transactionlist();
    void addBook();
    void searchBook();
    void viewAllBook();
    void addMember();
    void searchMember();
    void viewAllMember();
    void issue();
    void returnn();
}


class Library implements LibraryFeatures{
    public void addBook() {
        bl.addBook();
    }
    public void searchBook(){
        bl.show();
    }
    public void viewAllBook(){
        bl.display();
    }
    public void addMember(){
```



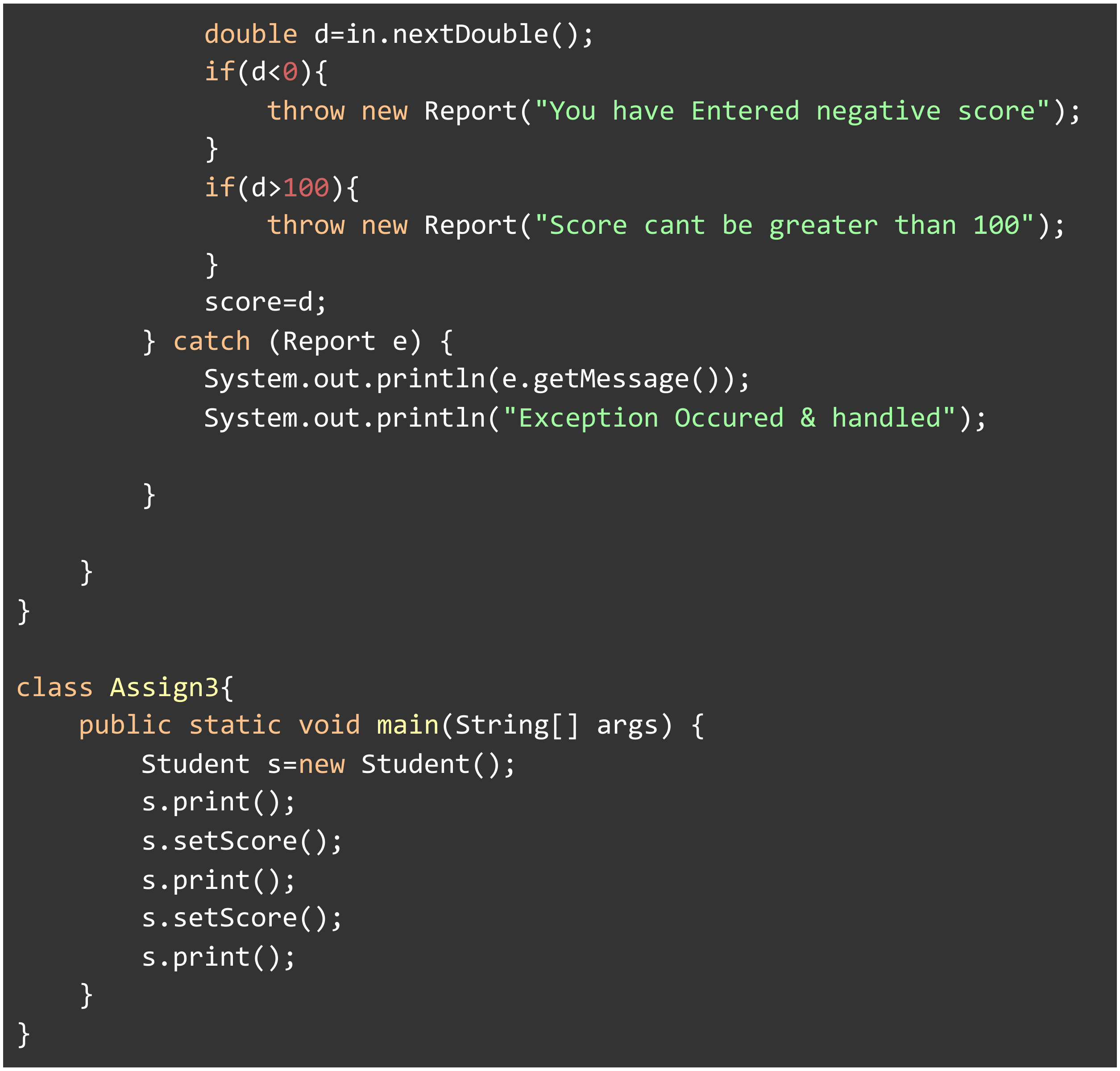
```
        ml.addMember();
    }
    public void searchMember(){
        ml.show();
    }
    public void viewAllMember(){
        ml.display();
    }
    public void issue(){
        Scanner in1=new Scanner(System.in);
        System.out.print("Enterbook id: ");
        String id=in1.next();
        if(!bl.checkid(id)){
            System.out.print("Enter Correct book id!!");
            return;
        }
        System.out.print("Entermember id: ");
        String id2=in1.next();
        if(!ml.checkid(id2)){
            System.out.print("Enter Correct member id!!");
            return;
        }
        if(bl.issue(id, -1) && ml.issue(id2, 1)){
        tl.addTransaction(id, id2);
        System.out.print("Book Issued!!");
        }
    }
    public void returnn(){
        Scanner in2=new Scanner(System.in);
        System.out.print("Enterbook id: ");
        String id3=in2.next();
        if(!bl.checkid(id3)){
            System.out.print("Enter Correct book id!!");
            return;
        }
        System.out.print("Entermember id: ");
        String id4=in2.next();
        if(!ml.checkid(id4)){
            System.out.print("Enter Correct member id!!");
            return;
        }
        if(bl.issue(id3, 1) && ml.issue(id4, -1)){
        tl.addTransaction(id3, id4);
        System.out.print("Book Returned!!");
        }
    }
    public void start(){
        loop1: do{
            System.out.println("\n1.Add_book\n2.Display all\n3.Display partricular\n4:Add Member\n5:Display all member\n6:Display Particular member\n7:Issue a book\n8:Return a book\n0.Exit\n");
            Scanner in=new Scanner(System.in);
            System.out.print("Enter choice: ");
            int choice=in.nextInt();
            switch (choice) {
                case 1:
                    addBook();
                    break;
                case 2:
                    viewAllBook();
                    break;
                case 3:
                searchBook();
                break;
                case 4:
               addMember();
                break;
                case 5:
                viewAllMember();
                break;
                case 6:
                searchMember();
                break;
                case 7:
                issue();
                break;
                case 8:
```

```
                returnn();
                break;
                case 0:
                break loop1;
            }
        }while(true);
    }
}

class Assign4{
    public static void main(String[] args) {
        Library l=new Library();
        l.start();
    }
}
```

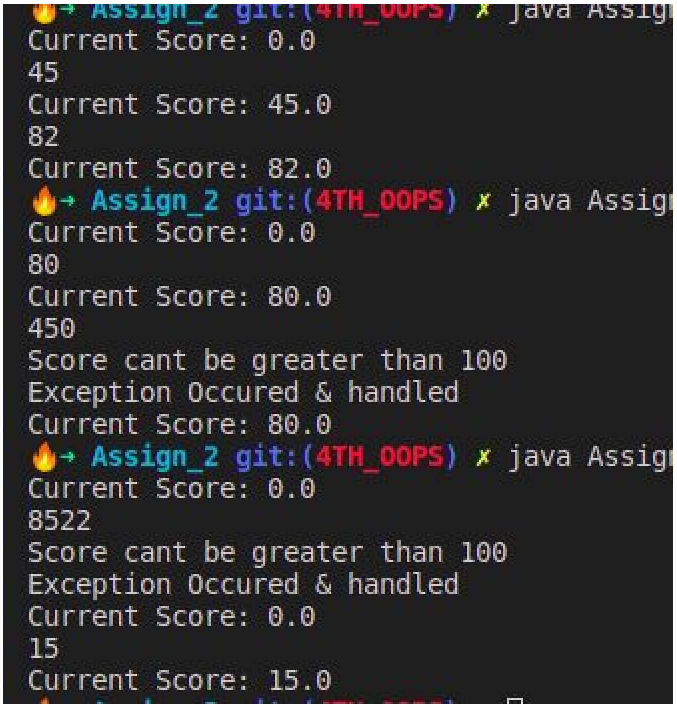
(Q5)

Exception hadling with custom excetion subclass.

```
import java.util.*;
class Report extends Exception{
    Report(String message){
        super(message);
    }
}
class Student{
    private String roll,name;
    private double score;
    Student(){
        score=0.0;
        name="";
    }
    void print(){
        System.out.println("Current Score: "+score);
    }
    public void setScore(){
        Scanner in= new Scanner(System.in);
        try {
```



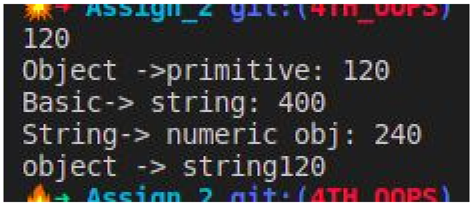
```
            double d=in.nextDouble();
            if(d<0){
                throw new Report("You have Entered negative score");
            }
            if(d>100){
                throw new Report("Score cant be greater than 100");
            }
            score=d;
        } catch (Report e) {
            System.out.println(e.getMessage());
            System.out.println("Exception Occured & handled");

        }

    }
}

class Assign3{
    public static void main(String[] args) {
        Student s=new Student();
        s.print();
        s.setScore();
        s.print();
        s.setScore();
        s.print();
    }
}
```



```
🔥→ Assign_2 git:(4TH_OOPS) ✗ java Assig
Current Score: 0.0
45
Current Score: 45.0
82
Current Score: 82.0
🔥→ Assign_2 git:(4TH_OOPS) ✗ java Assig
Current Score: 0.0
80
Current Score: 80.0
450
Score cant be greater than 100
Exception Occured & handled
Current Score: 80.0
🔥→ Assign_2 git:(4TH_OOPS) ✗ java Assig
Current Score: 0.0
8522
Score cant be greater than 100
Exception Occured & handled
Current Score: 0.0
15
Current Score: 15.0
```

(Q6)

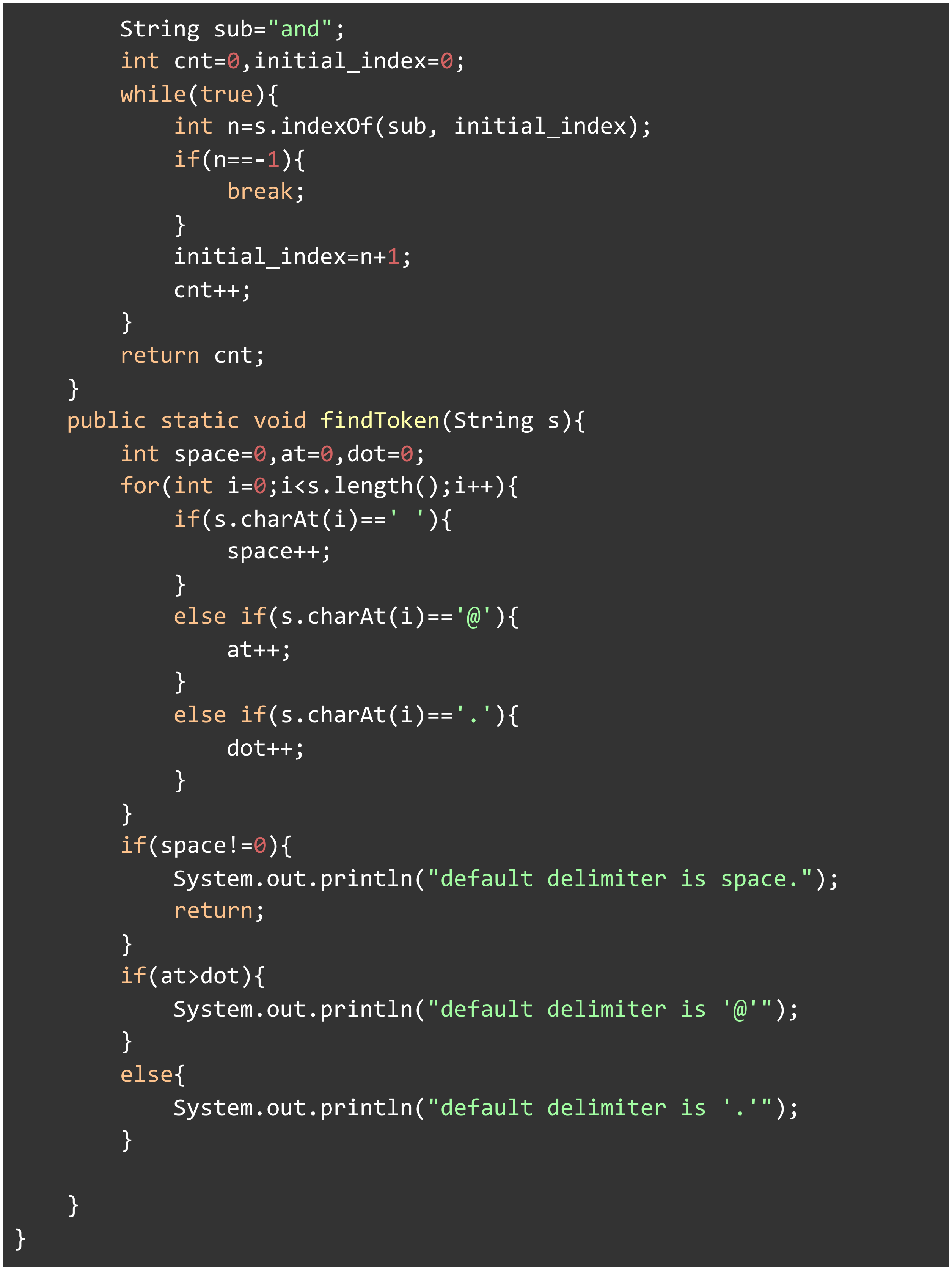
```
import java.util.*;
class Wrapper{
    public static void main(String[] args) {
        int ival=120;
        // Integer i=new Integer(ival);
        //primitive -> object
        Integer iobj=Integer.valueOf(ival);
        System.out.println(iobj);
        //obj-> primitive
        int op=iobj.intValue();
        System.out.println("Object ->primitive: "+ op);
        String ss=Integer.toString(400);
        System.out.println("Basic-> string: "+ss);
        Integer iobj1= Integer.valueOf("240");
        System.out.println("String-> numeric obj: "+iobj1);
        String s=iobj.toString();
        System.out.println("object -> string"+s);

    } }
```



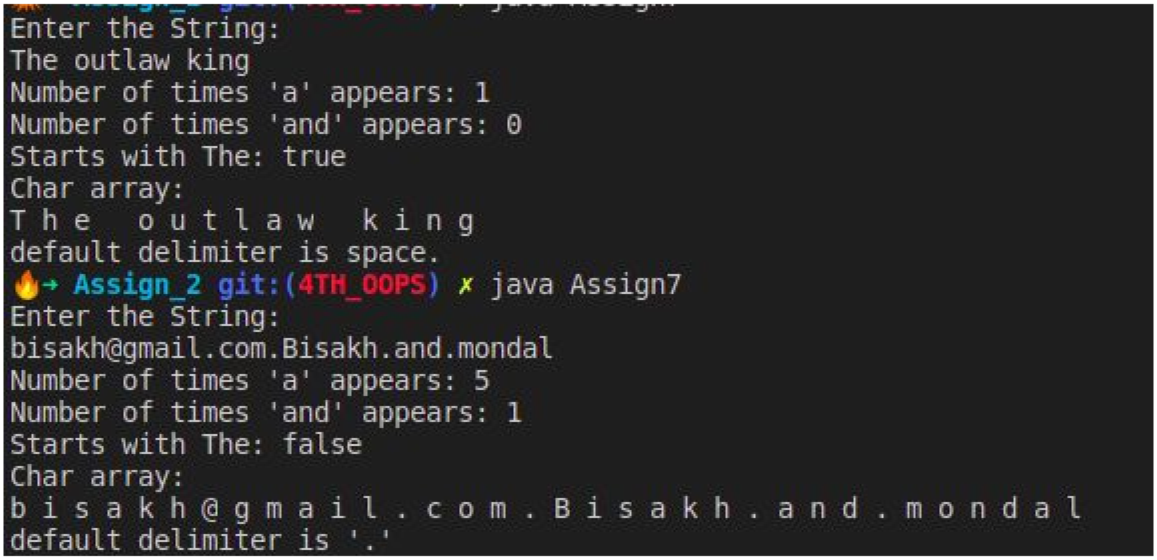
```
Assign_2 git:(4TH_OOPS)
120
Object ->primitive: 120
Basic-> string: 400
String-> numeric obj: 240
object -> string120
Assign_2 git:(4TH_OOPS)
```

(Q7)

```java
import java.util.*;
class Assign7{
    public static void main(String[] args) {
        String s;
        Scanner in=new Scanner(System.in);
        System.out.println("Enter the String: ");
        s=in.nextLine();
        s=s.trim();
        System.out.println("Number of times 'a' appears:
"+countA(s));
        System.out.println("Number of times 'and' appears:
"+countAnd(s));
        System.out.println("Starts with The: "+ s.startsWith("The"));
        char arr[]=s.toCharArray();
        System.out.println("Char array: ");
        for(char c: arr){
            System.out.print(c+" ");
        }
        System.out.println();
        findToken(s);
    }
    public static int countA(String s){
        int cnt=0;
        for(int i=0;i<s.length();i++){
            if(s.charAt(i)=='a'){
                cnt++;
            }
        }
        return cnt;
    }
    public static int countAnd(String s){
```



```
        String sub="and";
        int cnt=0,initial_index=0;
        while(true){
            int n=s.indexOf(sub, initial_index);
            if(n==-1){
                break;
            }
            initial_index=n+1;
            cnt++;
        }
        return cnt;
    }
    public static void findToken(String s){
        int space=0,at=0,dot=0;
        for(int i=0;i<s.length();i++){
            if(s.charAt(i)==' '){
                space++;
            }
            else if(s.charAt(i)=='@'){
                at++;
            }
            else if(s.charAt(i)=='.'){
                dot++;
            }
        }
        if(space!=0){
            System.out.println("default delimiter is space.");
            return;
        }
        if(at>dot){
            System.out.println("default delimiter is '@'");
        }
        else{
            System.out.println("default delimiter is '.'");
        }

    }
}
```



```
Enter the String:
The outlaw king
Number of times 'a' appears: 1
Number of times 'and' appears: 0
Starts with The: true
Char array:
T h e   o u t l a w   k i n g
default delimiter is space.
🔥→ Assign_2 git:(4TH_OOPS) ✗ java Assign7
Enter the String:
bisakh@gmail.com.Bisakh.and.mondal
Number of times 'a' appears: 5
Number of times 'and' appears: 1
Starts with The: false
Char array:
b i s a k h @ g m a i l . c o m . B i s a k h . a n d . m o n d a l
default delimiter is '.'
```