ASSIGNMENT 4

Date ....

Name: Divayam Sidhant Yadav
Roll no: 24017302220
Course: Java Programming

```
import java.io.*;
import java.util.*;

interface Show{ void show (); }

abstract class Item implements Show{
      int id; String title;
      Item(int id, String title){ this.id=id; this.title=
      title;}
}

class Book extends Item{
      String auth, cat; boolean issued;
      Book (int id, String t, String a, String c) {super
      (id, t); auth=a; cat=c; }
      void issue () {issued=true;}}
      void issue () {issued=false;}}
      public void show () {System.out.println(id+"|"+
      title + "|" + auth +"|"+ cat+"|"+ issued);}
;

class Member implements Show{
      int mid; String name, email; List<Integer> list=new
   ArrayList<>();
```

Spiral

Date ....

```
    Member(int id, String n, String e){mid=id;
    name=n; email=e;}
    void add(int id){list.add(id);}
    void rem(int id){list.remove(Integer.valueOf
    (id));}
    public ~~static~~ void show(){System.out.println
    (mid+"|"+name+"|"+email+"|"+list);}
}

class BookErr extends Exception{
    BookErr(String m){super(m);}
}

class Lib{
    Map<Integer, Book> bmap=new HashMap<>();
    Map<Integer, Member> mmap=new HashMap<>();
    int bc=100, mc=200;
    Lib(){load(); auto();}
    void addbook(String t, String a, String c){
        Book b=new Book(id++bc, t, a, c);
        bmap.put(b.id, b);
        System.out.println("Book ID:"+b.id);
    }
    void addMem(String n, String e){
        Member m=new Member(++mc, n, e);
        mmap.put(m.mid, m);
        System.out.println("Member ID:"+m.mid);
    }

    void issue(int bid, int mid) throws BookErr{
     if(!bmap.containsKey(bid)||!mmap.containsKey
                                (mid))
                                 return;
```

Spiral

Date ....

```
Book b= map.get(bid);
if (b.issued) throw new BookErr("Issued");
b.issue();
mmap.get(mid).add(bid);
System.out.println("Done");
}
  void ret(int bid, int mid) {
  if (!bmap.containsKey(bid) || !mmap.containsKey
  (mid)) return;
  bmap.get(bid).ret();
  mmap.get(mid).rem(bid);
  System.out.println("Returned");
}
void search (String k) {
    bmap.values().stream().filter(b->b.title.
    contains(k) || b.auth.contains(k) || b.cat.contains
    (k)).forEach(Book::show);
}
void sort () {
    bmap.values().stream().sorted( Comparator.
    comparing (b->b.title)).forEach(Book::
    show);
}
void save() {
    try(BufferWriter w=new BufferedWriter(new
    FileWriter ("book.txt"))) {
        for (Book b: bmap.values()) w.write
        (b.id +","+ b.title +","+ b.auth +","+
         b.cat +"," +b.issued +"\n");
    } catch (Exception e) {}
    try (BufferedWriter w=new BufferedWriter
    (new FileWriter ("member.txt")))
```

Spiral

Date............

```
        for(Member m: mmap.values()) w.write(m.id +
        "," + m.name + "," + m.email + "," + m.list + "\n");
    } catch (Exception e){}
}
void load(){
        try(BufferedReader r= new BufferedReader(
    new FileReader("books.txt"))){
          String s; while ((s=r.readLine())!=null){
            String b[]=s.split(",");
            Book b= new Book(Integer.parseInt
              (b[0]) (b[1]), b[2], b[3]);
            b.issued = Boolean.parseBoolean(b[4]);
            bmap.put(b.id, b); bc = Math.max(bc, b.id);
        }
    } catch (Exception e){}
    try(BufferedReader r=new BufferedReader(new
      FileReader("member.txt"))){
        String s; while((s=r.readLine())!=null){
          String p[]=s.split(",");
          Member m= new Member(Integer.parseInt
        (p[0]), p[1], p[2]);
        }
    } catch (Exception e){}
}
  Void auto(){
        Thread t= new Thread(()->{try{while(true)
      { save(); Thread.sleep(3000);}}
        catch (Exception e){}});
        t.setDaemon(true);
        t.start();
    }
}
```

Spiral

Date ....

```
Class Library System {
    public static void main (String []a) {
     lib l = new lib();
     Scanner s = new Scanner (System.in);
     While (true) {
        System.out.println ("1 AddBook\n2 Add Mem\n
        3 Issue\n 4 Return\n 5 Search \n6Sort \n
        7 Exit");
     try {
        int c=s.nextInt();
        Switch (c) {
              case 1-> {
                    s.nextLine ();
                    System.out.print ("Title:");
                    String t=s.nextLine();
                    Sout ("Auth:");
                    String au = s.nextLine();
                    Sout ("Cat:");
                    String cl=s.nextLine();
                    l.add book (t, au, cl);
        }
         case 2 -> {
              s.nextLine();
              Sout ("Name:");
              String n = s.nextLine();
              Sout ("Email:");
              String e= s.nextLine();
              l.addMem (n, e);
        }
```

Spiral

Date ....

```
case 3 -> {
    System.out.print("Bid: "); int bid=s.nextInt();
    Sout ("Mid: "); int mid=s.nextInt();
    L.issue (bid, mid);
}
case 4 -> {
    Sout ("Bid: ");
    int bid=s.nextInt();
    Sout ("Mid: ");
    int mid=s.nextInt();
    L.ret(bid, mid);
  }
case 5 -> {
    S.nextLine();
    System.out.print("Key: ");
    L.search(s.nextLine());
}
case 6 -> L.sort();
case 7 -> {L.save(); return; }
}
}
catch (BookErr e){
    System.out.println(e.getMessage());}
  Catch (Exception e) {Sout
      Sout ("Err!");
      S.nextLine();
  }
```

Spiral