Date:___.___.___

JAVA ASSIGNMENT - 4 :-

Name - Abhivrat Pathak

Roll no - 2401730045

u 1)>

```
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 ret () { 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 <> ().
         Member (int id , string ~~n~~ 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 void Show () { System . out . println
      (mid +"|" +name + "|" +email+ "|" + list); }
```

```
class BookErr extends Exception {
        Book Err (String m) {super(m); }
}
ClassLib {
Map<Integer, Book > bmap = new
Hash Map<> () ;
Map < Integer, Member > mmap = new HashMap<>();
int bc = 100, mC = 200 ;
Lib () { load (), acato(); }
void  add Book (String t, String a, String c) {
Book b = new Book (++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 Book
         Err {
         if (!bmap.containsKey (bid) || !mmap.contains
        Key (mid)) return;
          Book b = bmap.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) || !
```

Date:__.__.__

```
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 . compa
   ring (b -> b . title)) . forEach (Book :: show) ;
 }
     void save () {
       try (BufferedWriter w = new Bufferd Writer (
        new File Writer ("books . txt"))) {
            for (Book b : bmap . values ()) w . write
        (b . id + "," + b . title + "," + b . auth + "," + b . cat
     + b . issued) + "\n") ;
  } catch (Exception e) { } ~~try~~
  }
    Void load () {
       try (BufferedReader r = new BufferedReader
   (new FileReader ("books . txt"))) {
             String s ; while ((s = r . readLine ()) !
    = null) {
              String p[] = s . Split (",") ;
           Book b = new Book (integer . parseInt
             (p [0]) ;
      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 (int age.parseInt
          (p[0], p[1], p[2]);
            mmap.put(m.mid, m); mc = Math.max
                (mc, m.mid);
          }
        } catch (Exception e) {}
      }
      void auto () {
        thread t = new
      Thread(() -> {try {while(true){ save () thread.
        Sleep(3000); }} catch (Exception e) {}});
              t.setDaemon(true); t.start();
      }
  }
  public class LibrarySystem {
      public static void main (String []a) {
          Lib l = new Lib();
        Scanner s = new
      Scanner (System.in);
          while (true) {
          System.out.println("1.Add Book 2.Add Mem
          3.Issue 4.Return 5.Search 6.Sort 7.Exit");
        try {
            int c = s.nextInt();
```

```
Case 1 -> {
    s.nextline();
System.out.print("Title : "); String t = s.nextLine();
    System.out.print("Auth :"); String au = s.
        nextline();
    System.out.print ("Cat :"; String c1 = s.
        next Line ();
    l.addBook (t, au, c1);
}
    case 2 -> {
        s.nextLine();
        System.out.print~~("Cat :"); String~~
        ~~Cat~~ ("Name :"); String n = s.next
    Line();
        System.out.print("Email : ");
        String e = s.nextLine();
        l.addMem (n, e);
    }
        Case 3 -> {
    System.out.print("Bid : "); int bid
        = s.nextInt();
        System.out.print("Mid : ");
        int mid = s.next Int();
    ~~System.out~~ l.issue (bid, mid);
    }
        Case 4 -> {
    System.out.print ("Bid :"); int bid =
        s.nextInt();
    System.out.print ("Mid :"); int mid = s
        .next int ();
        l.out (bid, mid);
}
Case 5 -> {
    s.nextline();
  System.out.print("key : "); l.Search(s.nextline
            ());
     }
     Case 6 -> l.Sort();
      Case 7 -> {l.Save(); return;}
     }
   } catch (Book Err e)
{system.out.println(e.getMessage()); }
         Catch (Exception e){System.out.println
     ("Err"); s.nextLine(); }
    }
  }
}
```

oid

;

bid=

mid=s