

```
import java.util.*;

/**
 * A class to store the data of Library Members
 *
 * @author Deepak
 * @version 30.11.2023
 */
public class LibraryMember
{
    // instance variables
    private String memberId;
    private String memberName;
    private String memberContact;
    private ArrayList<PhysicalBook> booksBorrowed;

    /**
     * Constructor for class LibraryMember objects.
     */
    public LibraryMember(String memid, String memname, String memcontact)
    {
        memberId = memid;
        memberName = memname;
        memberContact = memcontact;
        booksBorrowed = new ArrayList<>();
    }

    /**
     * A method to get the Member Id of Library Member.
     */
    public String getMemberId()
    {
        return memberId;
    }

    /**
     * A method to set the Member Id of Library Member.
     */
    public void setMemberId(String memid)
    {
        memberId = memid;
    }

    /**
     * A method to get the Member Name of Library Member.
     */
    public String getMemberName()
    {
        return memberName;
    }
}
```

```
/**
 * A method to set the Member Name of Library Member.
 */
public void setMemberName(String memname)
{
    memberName = memname;
}

/**
 * A method to get the Member Contact of Library Member.
 */
public String getMemberContact()
{
    return memberContact;
}

/**
 * A method to set the Member Contact of Library Member.
 */
public void setMemberContact(String memcontact)
{
    memberContact = memcontact;
}

/**
 * An ArrayList to store the data of the PhysicalBook class.
 */
public ArrayList<PhysicalBook> getBooksborrowed()
{
    return booksBorrowed;
}

/**
 * A method to add the books borrowed by Library Member.
 */
public void addBooksBorrowed(PhysicalBook book)
{
    if (booksBorrowed.contains(book))
    {
        System.out.println("Book already borrowed");
    }
    else
    {
        booksBorrowed.add(book);
    }
}

/**
 * A method to print the details of Library Member.
```

```
    */
    public void printDetails()
    {
        System.out.println("Member Id: " + memberId);
        if (booksBorrowed.isEmpty())
        {
            System.out.println("No books borrowed");
        }
        else
        {
            System.out.println("Books borrowed: ");
            for (PhysicalBook book : booksBorrowed)
            {
                System.out.println("* " + book.getBookTitle());
            }
        }
    }

    /**
     * A method to get the the number of books borrowed by the Library Member.
     */
    public int getNumberOfBooksBorrowed()
    {
        return booksBorrowed.size();
    }
}
```