Ex. No. 7	Exception Handling
Date of Exercise	23-02-2018

Aim:

Create an interface called Library which contains the functions such as ReturnBook, BorrowBook, and Display. Create Student class and Faculty class both of which implements the Library interface. Use necessary class members such as regno, name, department, no_books, etc. Perform menu driven operations like CreateStudent, CreateFaculty, ReturnBook, BorrowBook and Display from a Main class.

Note: You should use array of objects.

Create custom Exceptions for following conditions:

a) A student can borrow maximum of 5 books otherwise raise exception.

b) A faculty can borrow maximum of 7 books otherwise raise exception.

Algorithm:

1) Start.
2) Create an interface called library containing the functions ReturnBook, BorrowBook and display.
3) Create a student class and faculty class both implementing library interface.
4) Initialize data members such as regno, name, department, no_books, etc.
5) Then, in main function perform menu driven operations. Use try catch to handle exception.
6) Also, use array of objects.

Source Code:

```
package exp7;

import java.util. Random;

import java. util. Scanner;
```

```
/**
 * @author JUSTIN PAUL
 */
class MyException extends Throwable{
    MyException(String msg){
        super(msg);}}
interface Library{
    void ReturnBook();
    void  BorrowBook()throws MyException;
    void Display();}
class Student implements Library{
    String regno;
    String name;
    String department;
    int np_books=200;
    private Scanner s;
    public String getRegno() {
        return regno;}
    public void setRegno(String regno) {
        this.regno = regno;}
    public String getName() {
        return name;}
    public void setName(String name) {
        this.name = name;
    }
```

```
public String getDepartment() {
    return department;}
public void setDepartment(String department) {
    this.department = department;}
public int getNp_books() {
    return np_books;}
  public void setNp_books(int np_books) {
    this.np_books = np_books;}
int no1;
@Override
public void BorrowBook()throws MyException{
   s=new Scanner(System.in);
   System.out.print("\nEnter the number of books you want to borrow: ");
   no1=s.nextInt();
   if(no1<6){
   if(no1<np_books){
       System.out.print("Book issued successfully.....");
       np_books-=no1;
       Random r=new Random();
       int mon=r.nextInt(12);
       int day=r.nextInt(30);
       int yr=2018;
       System.out.print("\nDue date: "+day+" "+mon+" "+yr);
    }
```

```
    Else{
        System.out.println("Books not available!!!");}}
    else{
        throw new MyException("You cannot borrow more than 5 books...");}}
  @Override
  public void ReturnBook(){
    if(no1>0){
    s=new Scanner(System.in);
    int no;
    System.out.print("\nEnter the number of books you want to return: ");
    no=s.nextInt();
    np_books+=no;
    System.out.print("\n"+no+" Books returned successfully....");}
  else
        System.out.println("No Books borrowed!!!!");}
  @Override
  public void Display(){
    s=new Scanner(System.in);
    System.out.println("Name: "+name);
    System.out.println("Register Number: "+regno);
    System.out.println("Department: "+department);
    System.out.println("No of books available: "+np_books);
    System.out.println("No of books borrowed: "+no1);
  }}
class Faculty implements Library{
```

```
String empno;
String name;
String department;
int np_books=150;
Scanner s;
public String getEmpno() {
    return empno; }
public void setEmpno(String empno) {
    this.empno = empno;}
public String getName() {
    return name;  }
public void setName(String name) {
    this.name = name; }
public String getDepartment() {
    return department;}
public void setDepartment(String department) {
    this.department = department;  }
public int getNp_books() {
    return np_books;}
public void setNp_books(int np_books) {
    this.np_books = np_books;}
 @Override
public void ReturnBook(){
   int no;
   s=new Scanner(System.in);
```

```
    System.out.print("\nEnter the number of books you want to return: ");
    no=s.nextInt();
    np_books+=no;
    System.out.print("\n"+no+" Books returned successfully....");}
int no1;
@Override
public void BorrowBook() throws MyException{
    s=new Scanner(System.in);
    System.out.print("\nEnter the number of books you want to borrow: ");
    no1=s.nextInt();
    if(no1<8){
    if(no1<np_books){
        System.out.print("\nBook issued..... ");
        np_books-=no1;
        Random r=new Random();
        int mon=r.nextInt(12);
        int day=r.nextInt(30);
        int yr=2018;
        System.out.println("\nDue date: "+day+" "+mon+" "+yr);}
    Else{
        System.out.println("Books not available!!!");}}
Else{
        throw new MyException ("You cannot borrow more than 7 books.....");}}
@Override
public void Display(){
```

```
        System.out.println("Name: "+name);
        System.out.println("Employee Number: "+empno);
        System.out.println("Department: "+department);
        System.out.println("No of books available: "+np_books);
        System.out.println("No of books borrowed: "+no1);}}
public class Exp7 {
    public static void main(String[] args) {
         Scanner s=new Scanner(System.in);
        boolean b,b2;
        b=true;
        b2=true;
        Faculty f[]= new  Faculty[3];
        for(int i=0;i<3;i++){
            f[i]= new Faculty();}
        Student st[]= new  Student[3];
        for(int i=0;i<3;i++){
            st[i]= new Student();}
        OUTER: while(b){
         System.out.println("CHOICES TO DO.....");
         System.out.println("1.Library for Students\n2.Library for faculty\n3.Exit");
         System.out.print("\nEnter the choice: ");
         int ch;
         ch=s.nextInt();
         switch(ch){
             case 1:
```

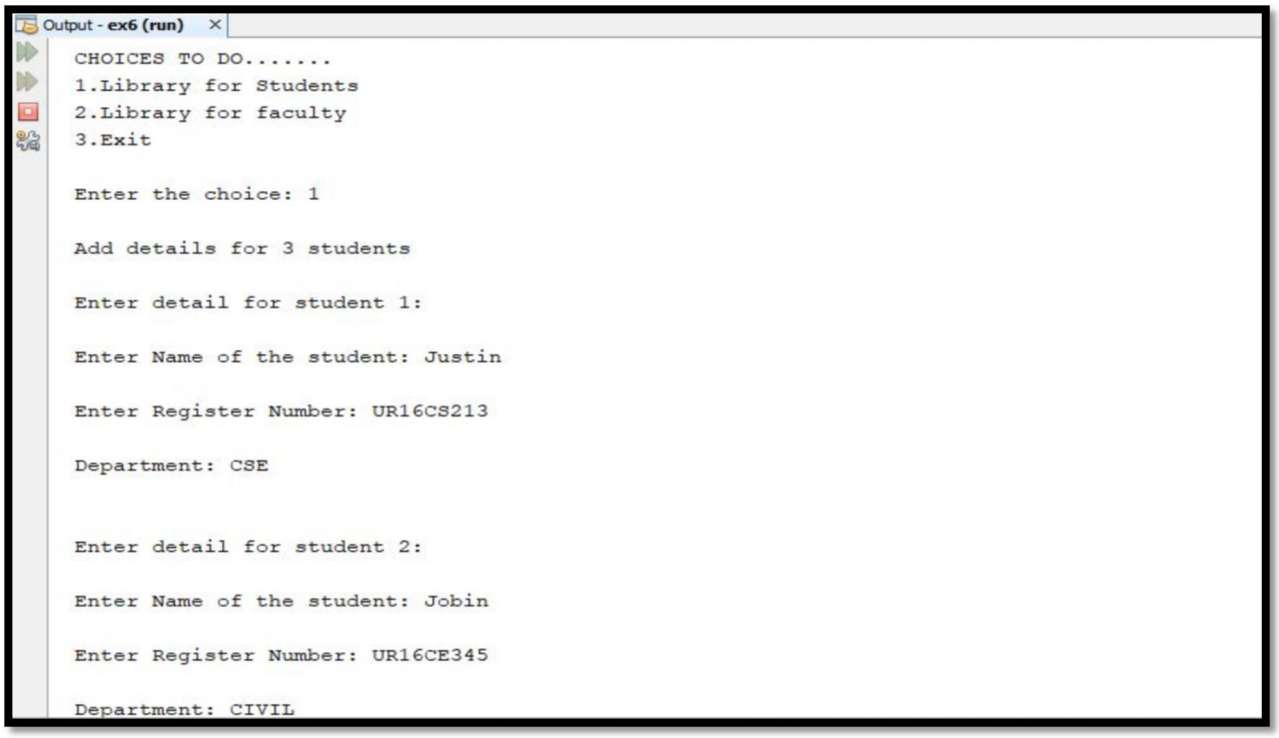
```
String name, department;
String regno;
System.out.print("\nAdd details for 3 students");
for(int i=0;i<3;i++){
System.out.println("\n\nEnter detail for "+(i+1)+" students:");
System.out.print("\nEnter Name of the student: ");
name=s.next();
System.out.print("\nEnter Register Number: ");
regno=s.next();
System.out.print("\nDepartment: ");
department=s.next();
st[i].setDepartment(department);
st[i].setName(name);
st[i].setRegno(regno);}
int cnt=0;
for(int i=0;i<3;i++){
while(b2){
String id;
System.out.print("\nEnter the student id to perform operation: ");
id=s.next();
for(int j=0;j<3;j++){
if(id.equals(st[j].regno)){
   cnt++;
 System.out.println("\nPress...\n1.Borrow Book\n2.Return Book\n3.Display\n4.Exit");
System.out.print("\nEnter the choice: ");
```

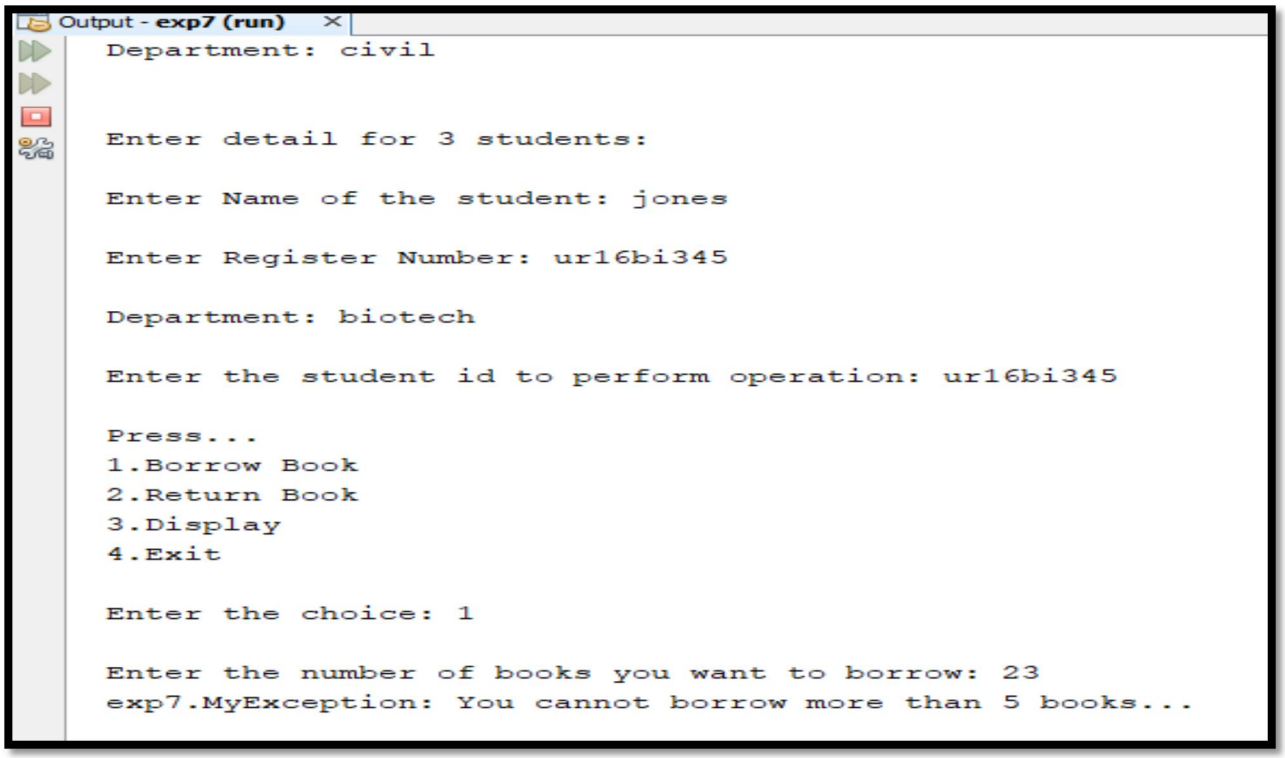
```
int ch1;
ch1=s.nextInt();
switch(ch1){
   case 1:
      try{
         st[j].BorrowBook();}
      catch(MyException e1){
         System.out.println(e1);}
      break;
   case 2:
      st[j].ReturnBook();
      break ;
   case 3:
      st[j].Display();
      break;
   case 4:
      b2=false;
      continue OUTER;}
}}}}
if(cnt==0)
      System.out.println("Register number not found....Please try again...");
case 2:
   System.out.println("Add details for 3 faculties");
   String name1,department1;
   String empno;
```

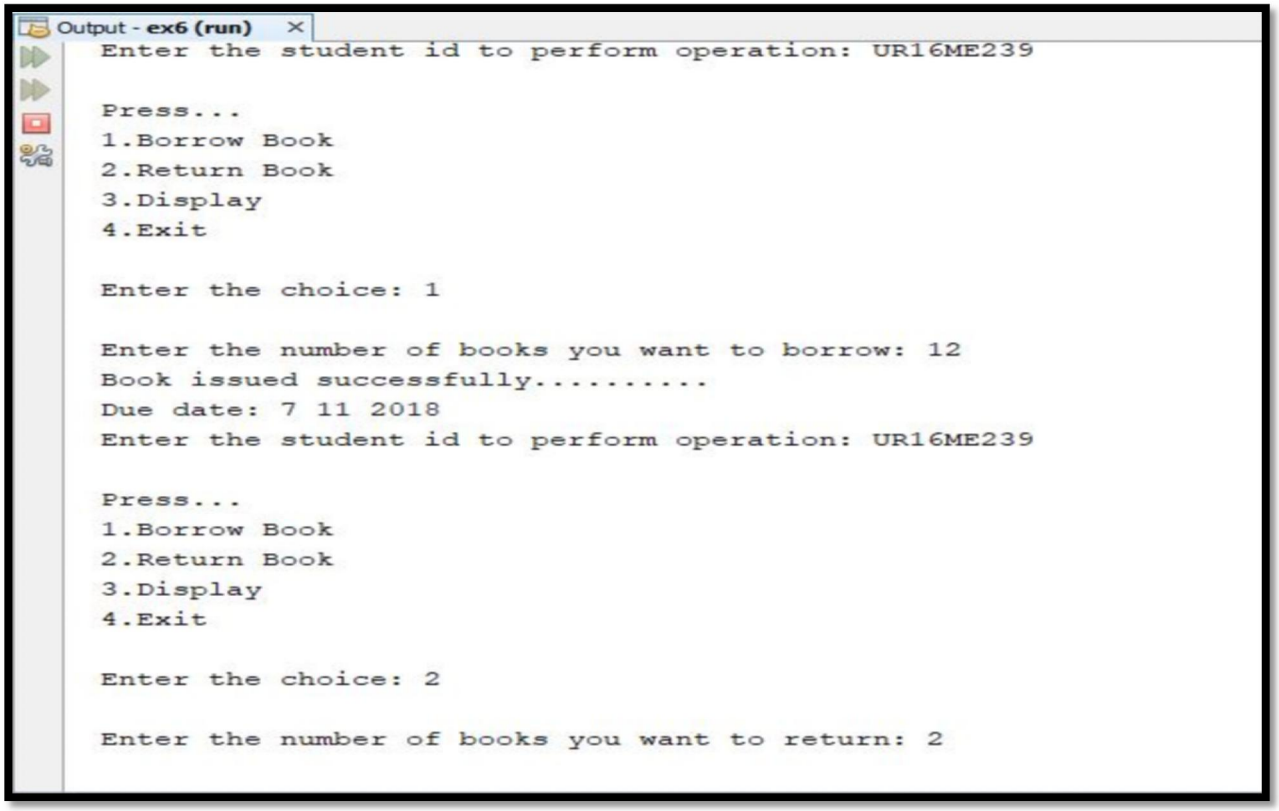
```
for(int i=0;i<3;i++){
    System.out.println("Enter details for "+(i+1)+" faculties");
    System.out.print("\nEnter Name of the Faculty: ");
    name1=s.next();
    System.out.print("\nEnter Employee Number: ");
    empno=s.next();
    System.out.print("\nDepartment: ");
    department1=s.next();
    f[i].setDepartment(department1);
    f[i].setName(name1);
    f[i].setEmpno(empno);
    int cnt1=0;
    while(b2){
        String id;
        System.out.print("\nEnter the faculty id to perform operation: ");
        id=s.next();
        for(int j=0;j<3;j++){
        if(id.equals(f[j].empno)){
            cnt1++;
System.out.println("\nPress...\n1.Borrow Book\n2.Return Book\n3.Display\n4.Exit");
        System.out.print("\nEnter the choice: ");
        int ch1;
        ch1=s.nextInt();
        switch(ch1){
            case 1:
```

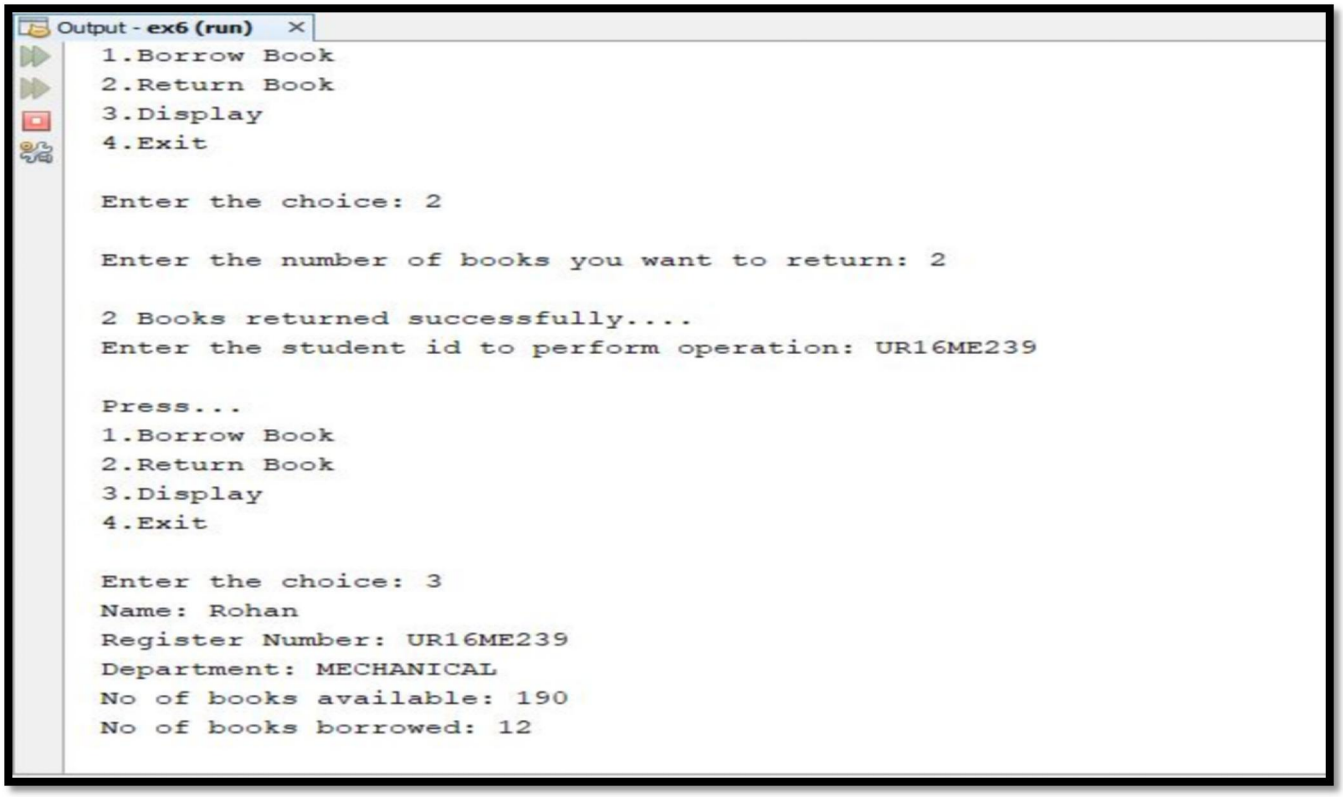
```
            try{
               f[j].BorrowBook();}
            catch(MyException e2){
               System.out.println(e2);}
            break;
         case 2:
            f[j].ReturnBook();
            break ;
         case 3:
            f[j].Display();
            break;
         case 4:
            b2=false;
         }}}}}
      case 3:
         b=false;}}}}
```

Input & Output:






Output - ex6 (run)
Enter the student id to perform operation: UR16ME239
Press...
1.Borrow Book
2.Return Book
3.Display
4.Exit
Enter the choice: 1
Enter the number of books you want to borrow: 12
Book issued successfully.....
Due date: 7 11 2018
Enter the student id to perform operation: UR16ME239
Press...
1.Borrow Book
2.Return Book
3.Display
4.Exit
Enter the choice: 2
Enter the number of books you want to return: 2


Output - ex6 (run)
1.Borrow Book
2.Return Book
3.Display
4.Exit
Enter the choice: 2
Enter the number of books you want to return: 2
2 Books returned successfully....
Enter the student id to perform operation: UR16ME239
Press...
1.Borrow Book
2.Return Book
3.Display
4.Exit
Enter the choice: 3
Name: Rohan
Register Number: UR16ME239
Department: MECHANICAL
No of books available: 190
No of books borrowed: 12

Video URL:

https://youtu.be/zHjiOfGlkxY

Result:

The program to a create a menu driven java application to create an interface called Library and create two classes student and faculty both implementing the interface library, and handle the exception is implemented successfully, and the output is verified.