

Application Development with Java

For 5th Sem

Computer Science and Engineering

(2016-2017)

Prepared By

**Mr. Santosh
Asst.Professor**

**Computer Science Deptatment
Canara Engineering College
Bejanapadavu-574219**

TABLE OF CONTENT

SL.NO	TOPIC
1.	<p>Develop a program for department management system for the following details:</p> <ol style="list-style-type: none"> 1. create arraylist class which contains: faculty_name, dept, branch, faculty_id. 2. create hashset class which contains: stu_name,stu_usn sub_name, marks, attendance. <p>create a class faculty which contains two constructors one to read faculty details from arraylist collection class and other one for marks details from hashset collection class. display marks allocated to subject.</p>
2.	<p>develop a java program using string class and stringbuffer class methods for the following details:</p> <ol style="list-style-type: none"> 1) read input string as “i like this program very much”. 2) reverse the individual words, we get the below string. 3) reverse the whole string from start to end and you get the desired output. 4) count number of words in it. 5) count number of characters in it.
3.	<p>Develop a login page using servlet to insert record into database and check for authentication, if login is successful display message as “well come to my servelt page” else stay in same page(login html page).</p> <p>note: create table login with user_name, password</p>
4.	<p>Develop a program for student database management system for the following details:</p> <ol style="list-style-type: none"> 1.create linkedlist class which conatins: s.name, usn, branch, sem. 2.create treeset class which contains: sub_name, marks. <p>create a class student which contains two constructors one to read student details from linkedlist collection class and other one for marks details from treeset collection class.display marks allocated to usn.</p>
5.	<p>Develop a program for library management system for the following details:</p> <ol style="list-style-type: none"> 1. create linkedlist class which contains: s.name, usn, branch, sem. 2. create treeset class which contains: book_name, book_author, book_editions. <p>create a class library which contains two constructors one to read student details from linkedlist collection class and other one for book details from treeset collection class. display book details for particular usn</p>
6.	<p>Develop a program for banking management system for the following details:</p> <ol style="list-style-type: none"> 1. create linkedlist class which contains: s.name, acc_no, address, p.no,balance. 2. create treeset class which contains: withdraw_amount <p>create a class bank which contains two constructors one to read person details from linkedlist collection class and other one for transaction details from treeset</p>

	collection class. find balance amount in particular acc_no.												
7.	<p>Develop a program for railway reservation system for the following details:</p> <ol style="list-style-type: none"> 1. create linkedlist class which contains: train_name, train_no, src, dest. 2. create treeset class which contains: passanger_name, p_mobilen. <p>create a class railway which contains two constructors one to read train details from linkedlist collection class and other one for booking details from treeset collection class. display booking details for particular passanger_name</p>												
8.	<p>Given a string of length n reverse the whole string without reversing the individual words in it.</p> <p>input:</p> <p>the first line contains t denoting the number of test cases. then follows description of test cases. each case contains a string containing spaces and characters.</p> <p>output:</p> <p>for each test case, output a single line containing the reversed string.</p> <p>example:</p> <table> <tr> <td>input:</td> <td>output</td> </tr> <tr> <td>2</td> <td>much very program this like i</td> </tr> <tr> <td>i like this program very much</td> <td>mno pqr</td> </tr> </table>	input:	output	2	much very program this like i	i like this program very much	mno pqr						
input:	output												
2	much very program this like i												
i like this program very much	mno pqr												
9.	<p>Given a string s and two integers l and r. print the characters in the range l to r of the string.</p> <p>note: assume zero based indexing.</p> <p>input: first line of input contains a single integer t which denotes the number of test cases. t test cases follows, first line of each test case contains a string s. second line consists of two integers l and r.</p> <p>output: corresponding to each test case, print the required output.</p> <p>example:</p> <table> <tr> <td>input:</td> <td>output:</td> </tr> <tr> <td>2</td> <td></td> </tr> <tr> <td>cdbkdub</td> <td>cdbkdu</td> </tr> <tr> <td>0 5</td> <td>blcsd</td> </tr> <tr> <td>sdiblcdbud</td> <td></td> </tr> <tr> <td>3 7</td> <td></td> </tr> </table>	input:	output:	2		cdbkdub	cdbkdu	0 5	blcsd	sdiblcdbud		3 7	
input:	output:												
2													
cdbkdub	cdbkdu												
0 5	blcsd												
sdiblcdbud													
3 7													
10.	<p>Given a string s containing all the characters and a character x, the task is to find last index of x in string s. print the characters in the form range x index to end of the string</p> <p>input:</p> <p>the first line of input contains an integer t denoting the number of test cases. then t test cases follow. each test case consists of two lines. first line of each</p>												

CASE STUDY1:

Develop a program for department management system for the following details:

- 1. create arraylist class which contains: faculty_name, dept,faculty_id.*
- 2. create hashset class which contains: stu_name,stu_usn sub_name, marks, attendance.*

create a class faculty which contains two constructors one to read faculty details from arraylist collection class and other one for marks details from hashset collection class. display marks allocated to subject.

```
class faculty
{
string faculty_name;
string dept;
string faculty_id;
string stu_name;
string stu_usn;
string sub_name;
string faculty_id;
int marks, attendance;
faculty (string name,string dept,string f_id)
{
faculty_name=name;
dept=dept;
faculty_id=f_id }
}
faculty (string name,string usn,string sub_name,int marks,int attd)
```

```
{  
stu_name=name;  
stu_usn=usn;  
sub_name=sub_name;  
marks=marks;  
attendance=attd;  
}
```

```
class arraylist1  
{  
public static void adddetails()  
{  
arraylist< faculty > l=new linkedlist< faculty >();  
l.add(new a("amar","cse","123"));  
l.add(new a("annu","cse","456"));  
l.add(new a("raj","cse","789"));  
system.out.println(l);  
}
```

```
class hashset1  
{  
public static void studentdetails()  
{  
faculty f[]=new faculty[100];  
hashset < faculty > l=new hashset < faculty >();
```

```
scanner s1=new scanner(system.in);
system.out.println("enter 5 students details");
for(int i=0;i<5;i++)
{
    string name=s1.next();
    string usn=s1.next();
    string sub_name=s1.next();
    int marks=s1.nextInt();
    int attd=s1.nextInt();
    l.add(f[i]=fnew a(name,usn,sub_name,marks,attd));
}
void getmarks()
{
    system.out.println("enter subject name");
    scanner s=new scanner(system.in);
    string sub=s.next();
    for(int i=0;i<5;i++)
    if(sub.equals(f[i].sub_name))
    system.out.println(marks);
}
class main
{
    public static void main(string ar[])
    {
```

```
arraylist1 a=new arraylist1();
```

```
hashset1 b=new hashset1();
```

```
a.adddetails();
```

```
b.studentdetails();
```

```
b.getmarks();
```

```
}
```

```
}
```


CASE STUDY2: Develop a java program using string class and stringbuffer class methods for the following details:

- 1) read input string as “i like this program very much”.***
- 2) reverse the individual words,***
- 3) reverse the whole string from start to end***
- 4) count number of words in it.***
- 5) count number of characters in it.***

```
public class stringReverse {  
  
    public static String reverseWord(String str){  
  
        String words[]=str.split("\\s");  
  
        String reverseWord="";  
  
        for(String w:words){  
  
            StringBuilder sb=new StringBuilder(w);  
  
            sb.reverse();  
  
            reverseWord+=sb.toString()+" ";  
  
        }  
  
        return reverseWord.trim();  
  
    }  
  
    static String reverseString(String str[]){  
  
        for(int i=s.length - 1; i >= 0; i--) {  
  
            ans+=str[i]+" ";  
  
        }  
  
        System.out.println("Reversed String:");  
  
        System.out.println(ans.substring(0,ans.length()-1));  
  
    }  
  
    static int wordcount(String string)
```

```

    {
        int count=0;

        char ch[]= new char[string.length()];

        for(int i=0;i<string.length();i++)

        {
            ch[i]= string.charAt(i);

            if( ((i>0)&&(ch[i]!=' ')&&(ch[i-1]!=' ')) || ((ch[0]!=' ')&&(i==0)) )

                count++;

        }

        return count;

    }

static int charcount(String string)

    {

        int count=0;

        for(int i = 0; i < string.length(); i++) {

            if(string.charAt(i) != ' ')

                count++;

        }

        return count;

    }

public static void main(String[] args) {

    System.out.println(reverseWord("i like this program very much "));

    System.out.println(reverseString("i like this program very much "));

    System.out.println(wordcount("i like this program very much "));

```

```
System.out.println(charWord("i like this program very much "));  
}  
}
```

CASE STUDY3: Develop a login page using servlet to insert record into database and check for authentication, if login is successful display message as “well come to my servelt page” else stay in same page(login html page).

note: create table login with user_name, password

Index.html

```
<form action="servlet1" method="post">  
Name:<input type="text" name="username"/><br/><br/>  
Password:<input type="password" name="userpass"/><br/><br/>  
<input type="submit" value="login"/>  
</form>
```

FirstServelt.java

```
import java.io.*;  
import javax.servlet.*;  
  
public class FirstServlet extends HttpServlet {  
    public void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
response.setContentType("text/html");  
PrintWriter out = response.getWriter();  
String n=request.getParameter("username");  
String p=request.getParameter("userpass");  
  
if(LoginDao.validate(n, p)){  
    RequestDispatcher rd=request.getRequestDispatcher("servlet2");  
    rd.forward(request,response);  
}  
else{  
    out.print("Sorry username or password error");  
    RequestDispatcher rd=request.getRequestDispatcher("index.html");  
    rd.include(request,response);  
}
```

```
    out.close();  
  }  
}
```

WelCome.java

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

```
import javax.servlet.*;
```

```
public class WelcomeServlet extends HttpServlet {  
  public void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
  
    String n=request.getParameter("username");  
    out.print("Welcome "+n);  
  
    out.close();  
  }  
}
```

CASE STUDY4:Develop a program for student database management system for the following details:

1.create linkedlist class which conatins: s.name, usn, branch, sem.

2.create treeset class which contains: sub_name, marks,usn.

create a class student which contains two constructors one to read student details from Linkedlist collection class and other one for marks details from treeset collection class.display student information allocated to usn.

```
class student
{
string stu_name;
string stu_usn;
string sub_name;
string branch;
int marks,sem;
static student f[]=new student[100];
static student s[]=new student[100];
student(sting usn,string sub_name,int marks)
{
stu_usn=usn;
sub_name=sun_name;
marks=marks;
}
student (string name,string usn,string bran,int sem)
{
stu_name=name;
```

```

stu_usn=usn;
branch=bran;
this.sem=sem;
}

static void getmarks()
{
system.out.println("enter usn");
scanner s=new scanner(system.in);
string usn=s.next();
for(int i=0;i<5;i++)
if(usn.equals(f[i]. stu_usn))
system.out.println(s[i].stu_name +""
+s[i].stu.usn+""+s[i].branch+""+s[i].sem+""+f[i].sub_name+""+f[i].marks);
}

}

class linkedlist1
{
public static void adddetails()
{
linkedlist<student> l=new linkedlist<student>();
system.out.println("enter 5 students details");
for(int i=0;i<5;i++)
{ string name=s.next();

```

```
string usn=s.next();
string branch=s.next();
int sem=s.nextint();
l.add(s[i]=fnew student(name,usn,branch,sem));
}
}
class treeset1
{
public static void studentdetails()
{
treeset <student> l=new treeset <student>();
scanner s1=new scanner(system.in);
system.out.println(“enter 5 students details”);
for(int i=0;i<5;i++)
{
string usn=s1.next();
sub_name=s1.next();
int marks=s1.nextint();
l.add(f[i]=fnew student(usn,sub_name,marks));
}
}
class main
{
public static void main(string ar[])
{
```



```
arraylist1 a=new arraylist1();
```

```
treeSet1 b=new treeSet1();
```

```
a.addDetails();
```

```
b.studentDetails();
```

```
faculty f=new faculty();
```

```
getmarks();
```

```
}
```

```
}
```

CASE STUDY5:Develop a program for library management system for the following details:

1. create linkedlist class which contains: s.name, usn, branch, sem.

2. create treeset class which contains: book_name, book_author, book_editions.

create a class library which contains two constructors one to read student details from linkedlist collection class and other one for book details from treeset collection class. display book details for particular usn

```
class library
{
static library f[]=new library [100];
static library s[]=new library [100];
string stu_name,book_name,book_author,book_editions;
string stu_usn;
string sub_name;
string branch;
int sem;
library (string b_name,string b_author,string b_editions,string usn)
{
book_name=b_name;
book_author=b_author;
book_editions=b_editions;
stu_usn=usn;
}
library (string name,string usn,string bran,int sem)
{
```

```

stu_name=name;
stu_usn=usn;
branch=bran;
this.sem=sem;
}
static void getmarks()
{
system.out.println("enter usn");
scanner s=new sacnner(system.in);
string usn=s.next();
for(int i=0;i<5;i++)
if(usn.equals(f[i]. stu_usn))
system.out.println(s[i].stu_name
+"""+f[i].stu_usn+"""+f[i].book_name+"""+f[i].book.author+"""+f[i].book.editions);
}
}

class linkedlist1
{
public static void adddetails()
{
linkedlist< libary > l=new linkedlist< libary >();
system.out.println("enter 5 students details");
for(int i=0;i<5;i++)

```

```
{ string name=s.next();
string usn=s.next();
string branch=s.next();
int sem=s.nextint();
l.add(s[i]=new library(name,usn,branch,sem));
}
}
class treeset1
{
public static void studentdetails()
{
treeset <library> l=new treeset <library>();
scanner s1=new scanner(system.in);
system.out.println("enter 5 books details");
for(int i=0;i<5;i++)
{
string uns=s1.next();
string b_name=s1.next();
string b_author=s1.next();
string b_editions=s1.next();
l.add(f[i]=new library(b_name,b_author,b_editions,usn));
}
}
class main
```

```
{  
public static void main(string ar[])  
{  
arraylist1 a=new arraylist1();  
treeset1 b=new treeset1();  
a.adddetails();  
b.studentdetails();  
faculty f=new faculty();  
f.getmarks();  
}  
}
```

CASE STUDY 6 : Develop a program for banking management system for the following details:

1. create linkedlist class which contains: s.name, acc_no, address, p.no,amount.

2. create treeset class which contains: withdraw_amount

create a class bank which contains two constructors one to read person details from linkedlist collection class and other one for transaction details from treeset collection class. find balance amount in particular acc_no.

```
import java.util.*;

public class bank

{

string c_name;

int acc_no;

string address;

int amount,withdraw_amt,ph_no;

static bank f[]=new bank[100];

static bank s1[]=new bank[100];

bank(string name,int acc_no,string add,int pno,int amt)

{

c_name=name;

acc_no=acc_no;

address=add;

ph_no=pno;

amount=amt;

}
```

```
bank (int withdraw,int accno)
```

```
{
```

```
withdraw_amt=withdraw;
```

```
acc_no=accno;
```

```
}
```

```
static void getmarks()
```

```
{
```

```
system.out.println("enter acc_no");
```

```
scanner s=new scanner(system.in);
```

```
int acc_no=s.nextInt();
```

```
for(int i=0;i<2;i++)
```

```
if(acc_no==f[i]. acc_no)
```

```
{
```

```
int amount=s1[i].amount-f[i].withdraw_amt;
```

```
s1[i].amount=amount;
```

```
system.out.println(s1[i].c_name +"" +s1[i].acc_no+""+s1[i].amount);
```

```
}
```

```
}
```

```
public static void main(string ar[])
```

```
{
```

```
linkedlist1 a1=new linkedlist1();
```

```
treaset1 b1=new treaset1();
```

```
a1.adddetails();
```

```
b1.studentdetails();
```

```
getmarks();
```

```
}
```

```
}
```

```
class linkedlist1
```

```
{
```

```
void adddetails()
```

```
{ linkedlist<bank> l=new linkedlist<bank>();
```

```
bank s1[]=new bank[100];
```

```
scanner s=new scanner(system.in);
```

```
system.out.println("enter 5 customer details");
```

```
for(int i=0;i<2;i++)
```

```
{
```

```
string name=s.next();
```

```
string add=s.next();
```

```
int a_no=s.nextint();
```

```
int pno=s.nextint();
```

```
int amt=s.nextint();
```

```
l.add(s1[i]=new bank(name,a_no,add,pno,amt));
```



```
}  
}  
  
}  
class treeset1  
{static bank f[]=new bank[100];  
void studentdetails()  
{  
treeset <bank> l=new treeset <bank>();  
scanner s=new scanner(system.in);  
system.out.println("enter withdraw amount ");  
for(int i=0;i<2;i++)  
{  
int amt=s.nextInt();  
int a_no=s.nextInt();  
l.add(f[i]=new bank(amt,a_no));  
}  
}  
}
```

CASE STUDY 7:Develop a program for railway reservation system for the following details:

1. create linkedlist class which contains: train_name, train_no, src, dest, p_mobilen.

2. create treeset class which contains: passanger_name, p_mobilen.

create a class railway which contains two constructors one to read train details from linkedlist collection class and other one for booking details from treeset collection class. display booking details for particular p_mobilen

```
class railway
{
static railway f[]=new railway[100];
static railway s[]=new railway[100];
string p_name,p_no,src,dest,train_name;
int train_no;
railway(sting tname,int tno,string src,string dest,int pno)
{
train_name=tname;
train_no=tno;
src=src;
dest=dest;
p_no=pno;
}
student (string pname,int pno)
{
```

```

p_name=pname;
p_no=pno;
}
static void getmarks()
{
system.out.println("enter passanger phone number);
scanner s=new sacnner(system.in);
int pno=s.nextint();
for(int i=0;i<5;i++)
if(pno==f[i]. p_no))
system.out.println(f[i].p_name +""
+f[i].p_no+""+s[i].train_name+""+s[i].train_no""+s[i].src+""+s[i].dest);
}
}

```

```

class linkedlist1
{
public static void adddetails()
{ scanner s=new scanner(system.in);
linkedlist<railway> l=new linkedlist<railway>();
system.out.println("enter 5 train details");
for(int i=0;i<5;i++)
{ string tname=s.next();
string src=s.next();
string dest=s.next();

```

```
int tno=s.nextint();
int pno=s.next();
l.add(s[i]=new railway(tname,tno,src,dest,pno));
}
}
class treeset1
{
public static void studentdetails()
{
treeset <bank> l=new treeset <bank>();
scanner s1=new scanner(system.in);
system.out.println("enter 5 passanger details");
for(int i=0;i<5;i++)
{
string pname=s1.next();
int pno=s1.nextint();
l.add(f[i]=fnew bank(pname,pno));
}
}
class main
{
public static void main(string ar[])
{
arraylist1 a=new arraylist1();
treeset1 b=new treeset1();
```

```
a.adddetails();
```

```
b.studentdetails();
```

```
getmarks();
```

```
}
```

```
}
```

CASE STUDY 8: Given a string of length n reverse the whole string without reversing the individual words in it.

input:

the first line contains t denoting the number of test cases. then follows description of test cases. each case contains a string containing spaces and characters.

output:

for each test case, output a single line containing the reversed string.

example:

input:

output

1 i like this program very much

much very program this like i

```
import java.util.*;
import java.io.*;
public class Main {

void fun() throws exception
{
system.out.println("enter string");
inputstreamreader r=new inputstreamreader(system.in);
bufferedReader s1=new bufferedreader(r);
string s2=s1.readline();
string s[]=s2.split(" ");
string ans="";
for(int i=s.length - 1; i >= 0; i--) {
ans+=s[i]+" ";
}
system.out.println("reversed string:");
system.out.println(ans.substring(0,ans.length()-1));
}
public static void main(string[] args)throws exception {
Main m=new Main();
for(int i=0;i<2;i++)
m.fun();
}
}
```

CASE STUDY 9: Given a string *s* and two integers *l* and *r*. print the characters in the range *l* to *r* of the string. note: assume zero based indexing.

input: first line of input contains a single integer *t* which denotes the number of test cases. *t* test cases follows, first line of each test case contains a string *s*. second line consists of two integers *l* and *r*.

output: corresponding to each test case, print the required output.

example:

input:

2

Cdbkdub 0 5

Sdiblcdbud 3 7

output:

cdbkdu

blcsd

```
import java.util.*;
import java.io.*;
public class main {

void fun() throws exception{
scanner s3=new scanner(system.in);
system.out.println("enter string");
inputstreamreader r=new inputstreamreader(system.in);
bufferedReader s1=new bufferedreader(r);
string s2=s1.readline();
system.out.println("enter range start and end index");
int s=s3.nextint();
int e=s3.nextint();
system.out.println(s2.substring(s,e+1));

}
public static void main(string[] args)throws exception {
main m=new main();
for(int i=0;i<2;i++)
m.fun();
}
}
```

CASE STUDY 10: Given a string *s* containing all the characters and a character *x*, the task is to find last index of *x* in string *s*. print the characters in the form range *x* index to end of the string

input: The first line of input contains an integer *t* denoting the number of test cases. then *t* test cases follow. each test case consists of two lines. first line of each test case contains a string *s* and the second line contains a character *x*.

output: For each test case, print the last index in new line, if *x* is not present in the string print "-1".

constraints:

$1 \leq t \leq 100$

$1 \leq \text{string length} \leq 105$

example:

input:

2

wel come to cec college

hello , iam strong in java

output:

10 llege

12 strong in java

```
import java.util.*;
import java.io.*;
public class main {

void fun() throws exception{
scanner s3=new scanner(system.in);
system.out.println("enter string");
inputstreamreader r=new inputstreamreader(system.in);
bufferedReader s1=new bufferedreader(r);
string s2=s1.readline();
system.out.println("enter range start index");
string s=s3.next();
int s1=s2.lastIndexOf(s);
system.out.println("last index of string is"+s1+"sub string is"+s2.substring(s));
}
```



```
public static void main(string[] args) throws exception {  
    main m=new main();  
    for(int i=0;i<2;i++)  
        m.fun();  
    }  
}
```

CASE STUDY 11: Given a string containing multiple words, your task is to count the characters in each word and display them.

input: The first line of input contains an integer t denoting the number of test cases. then t test cases follow. each testcase contains a string as input.

output: For each testcase, print the number of characters in each word of the string.

constraints:

$1 \leq t \leq 100$

example:

input:

2

wel come to cec

college benjanapadavu

output:

3 4 2 3

7 13 .

```
import java.util.*;
import java.io.*;
public class main {

void fun() throws exception{
scanner s3=new scanner(system.in);
system.out.println("enter string");
inputstreamreader r=new inputstreamreader(system.in);
bufferedReader s1=new bufferedreader(r);
string s2=s1.readline();
char ch[]=s2.toStringarray();
int i=0;
while(ch[i]!="\0")
{
if(ch[i]!=" ")
count++;
else
{
system.out.println(count);
count=0;
}
i++;
}
```

```
}
```

```
public static void main(String[] args) throws Exception {  
    Main m = new Main();  
    for (int i = 0; i < 2; i++)  
        m.fun();  
}
```

CASE STUDY 12:Develop a student database management system using servlet to insert record into database and display the student information based on usn.

Index.html

```
<html>

<body>

<form action="servlet/Search">

Enter your Rollno:<input type="text" name="usn"/><br/>

  <input type="submit" value="search"/>

</form>

</body>

</html>
```

FetchData.java

```
import java.io.*;

import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;

public class Search extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String rollno=request.getParameter("usn");
        int usn=Integer.valueOf(rollno);
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
            PreparedStatement ps=con.prepareStatement("select * from result where rollno=?");
            ps.setInt(1,roll);
            out.print("<table width=50% border=1>");
            out.print("<caption>Result:</caption>");
            ResultSet rs=ps.executeQuery();
```

```
/* Printing column names */
ResultSetMetaData rsmd=rs.getMetaData();
int total=rsmd.getColumnCount();
out.print("<tr>");
for(int i=1;i<=total;i++)
{
out.print("<th>"+rsmd.getColumnName(i)+"</th>");
}
out.print("</tr>");

while(rs.next())
{
out.print("<tr><td>"+rs.getInt(1)+"</td><td>"+rs.getString(2)+"
</td><td>"+rs.getString(3)+"</td><td>"+rs.getString(4)+"</td></tr>");
}
out.print("</table>");

}catch (Exception e2) {e2.printStackTrace();}

finally{out.close();}

}
}
```

CASE STUDY 13:Develop a student database management system using jsp to insert record into database and display the student information based on usn.

Index.html

```
<html>
<body>

<form method="post" action=" Insertrecord.jsp">
First name:<br>
<input type="text" name="name">
<br>Branch:<br>
<input type="text" name="Branch">
<br>USN:<br>
<input type="text" name="USN">
<br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

Insertrecord.jsp

```
<%
String name=request.getParameter("name");
String branch=request.getParameter("Branch");
String usn=request.getParameter("USN");

try
{
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "");
Statement st=conn.createStatement();
int i=st.executeUpdate("insert into
users(first_name,last_name,city_name,email)values('"+name+"','"+branch+"','"+usn+"')");
out.println("Data is successfully inserted!");
}
catch(Exception e)
{
System.out.print(e);
e.printStackTrace();}
%>
```

CASE STUDY 14: Develop a student database management system to insert record into database and display the student information based on ia avg marks >20.

Index.html

```
<html>
<body>
<form action="servlet/Search">
  <input type="submit" value="search"/>
</form>
</body>
</html>
```

FetchData.java

```
import java.io.*;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class Search extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String rollno=request.getParameter("usn");
int usn=Integer.valueOf(rollno);
try{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
PreparedStatement ps=con.prepareStatement("select * from result where iaavg>=?");
ps.setInt(1,20);
out.print("<table width=50% border=1>");
out.print("<caption>Result:</caption>");
ResultSet rs=ps.executeQuery();
```

```
/* Printing column names */
ResultSetMetaData rsmd=rs.getMetaData();
int total=rsmd.getColumnCount();
out.print("<tr>");
for(int i=1;i<=total;i++)
{
out.print("<th>"+rsmd.getColumnName(i)+"</th>");
}
out.print("</tr>");

while(rs.next())
{
out.print("<tr><td>"+rs.getInt(1)+"</td><td>"+rs.getString(2)+"
</td><td>"+rs.getString(3)+"</td><td>"+rs.getString(4)+"</td></tr>");
}
out.print("</table>");

}catch (Exception e2) {e2.printStackTrace();}

finally{out.close();}

}
}
```