CORE JAVA PROJECT

PROJECT TITLE: STUDENT DETAILS MANAGEMENT SYSTEM

AIM:

- To handle the student information without any interruptions.

LANGUAGES:

- JAVA

SOFTWARE REQUIREMENTS:

- Eclipse for java

SYNOPSIS:

In this STUDENT DETAILS MANAGEMENT SYSTEM there 5 main operations are there.

That MAIN OPERATIONS are:

1.INSERT	-->	Admin perform that operation.
2.DISPLAY	-->	Admin perform that operation.
3.SEARCH	-->	Admin perform that operation.
4.DELETE	-->	Admin perform that operation.
5.UPDATE	-->	Admin perform the operation.

The control is given only for admin or system operator.First all the details of the students are inserted.Then the inserted details are shown using the next operation.

Then the particular detail of a student is retrieved using the id of a student. In the same way like using id the unwanted details are deleted.Then if an information is entered by mistake then that will be deleted or updated.

INITIAL SETUP:

(IN JAVA):

Package Name: STUDENT_project.

Classes Name: StudentManagement

STUDENMANAGEMENT:

CODE:

```
*****************************CORE JAVA PROJECT*******************
*                       CSR CAPGEMINI TRAINING PROJECT
*                      EDUBRIDGE INDIA PRIVATE LIMITED
*              PROJECT TITLE: STUDENT DETAILS MANAGEMENT SYSTEM
*               UNDER THE GUIDENCE OF TRAINER MRS.INDRAKA MALLI
*                                              @DONE BY VISHALI P
*
*In STUDENT DETAILS MANAGEMENT SYSTEM:
*MAIN OPERATIONS:
*1.INSERT     –>Admin only perform this operation
*2.DISPLAY    –>Admin only perform this operation
*3.SEARCH     –>Admin only perform this operation
*4.DELETE     –>Admin only perform this operation
*5.UPDATE     –>Admin only perform this operation
```

```
import java.util.Scanner;
import java.util.*;
import java.util.ArrayList;
import java.util.List;
class Student
{
    private int studentid;
    private String studentname;
    private int studentatd;
    private int studentfees;

    Student(int studentid,String studentname,int studentatd,int studentfees)
    {
        this.studentid = studentid;
        this.studentname = studentname;
        this.studentatd = studentatd;
        this.studentfees = studentfees;
    }
    public int getStudentid()
    {
        return studentid;
    }
    public String getStudentname()
    {
        return studentname;
    }
```

```
public int getStudenatd()
    {
        return studentatd;
    }
    public int getStudentfees()
    {
        return studentfees;
    }
    public String toString()
    {
        return studentid+" "+studentname+" "+studentatd+" "+studentfees;
    }
}
public class StudentManagement
{
    public static void main(String[] args)
    {
        List<Student> s=new ArrayList<Student>();
        Scanner sc=new Scanner(System.in);
        Scanner sc1=new Scanner(System.in);
        int ch;
        do
        {
            System.out.println("1.Insert");
            System.out.println("2.Display");
            System.out.println("3.Search");
```

```
System.out.println("4.Delete");
System.out.println("5.Update");
System.out.println("6.Exit");
System.out.println("Enter Your Choice : ");
ch = sc.nextInt();

switch(ch)
{
    case 1:
        System.out.print("Enter Studentid : ");
        int sid = sc.nextInt();
        System.out.print("Enter Studentname : ");
        String sname = sc1.nextLine();
        System.out.print("Enter Studentatd in % : ");
        int satd = sc.nextInt();
        System.out.print("Enter Studentfees : ");
        int sfees = sc.nextInt();

        s.add(new Student(sid,sname,satd,sfees));
    break;
    case 2:
        System.out.println("--------------------------");
        Iterator<Student> i=s.iterator();
        while(i.hasNext())
        {
            Student e=i.next();
```

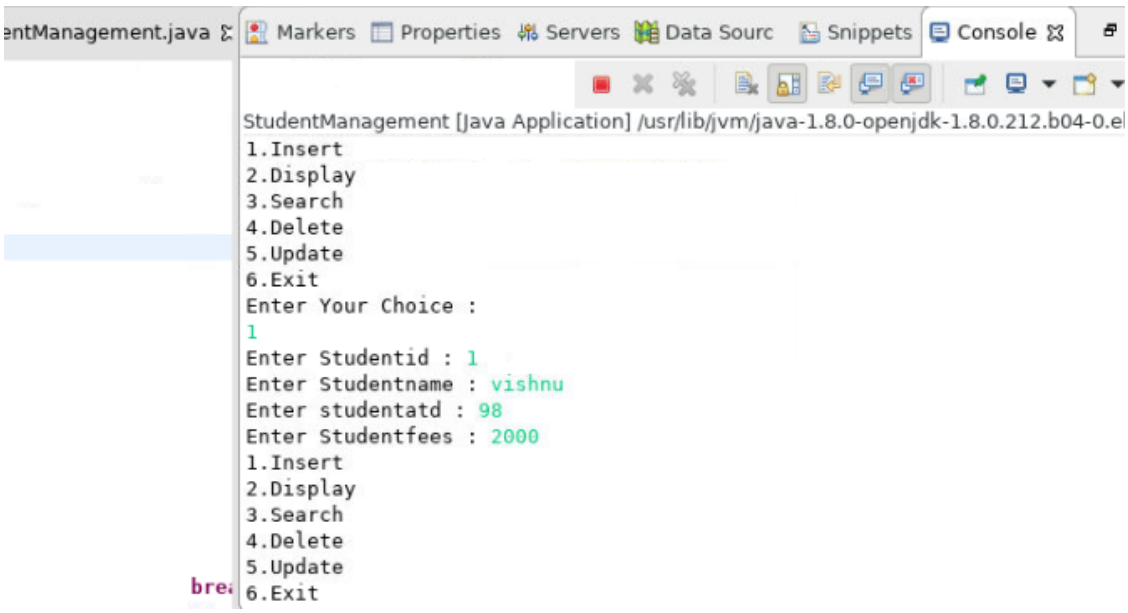
```
            System.out.println(e);
        }
        System.out.println("-------------------------");
    break;
    case 3:
        boolean found=false;
        System.out.println("Enter Studentid to Search : ");
        int studentid =sc.nextInt();
        System.out.println("-------------------------");
        i=s.iterator();
        while(i.hasNext())
        {
            Student e=i.next();
            if(e.getStudentid()==studentid)
            {
                System.out.println(e);
                found=true;
            }
        }
        if(!found)
        {
            System.out.println("Record Not Found");
        }
        System.out.println("-------------------------");
    break;
    case 4:
```

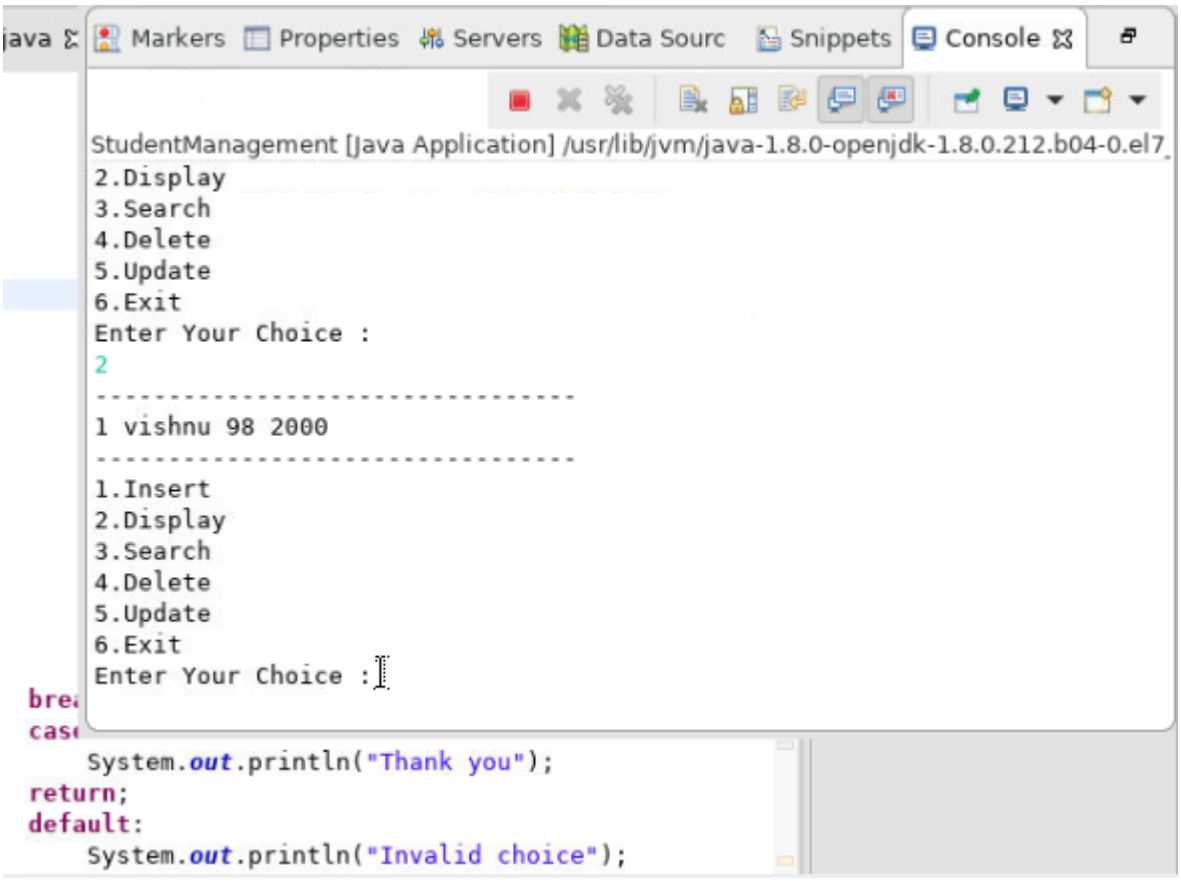
```
                    found=false;
                    System.out.println("Enter Studentid to Delete : ");
                    studentid =sc.nextInt();
                    System.out.println("-------------------------");
                    i=s.iterator();
                    while(i.hasNext())
                    {
                        Student e=i.next();
                        if(e.getStudentid()==studentid)
                        {
                            i.remove();
                            found=true;
                        }
                    }
                    if(!found)
                    {
                        System.out.println("Record Not Found");
                    }
                    else
                    {
                        System.out.println("Record is Deleted
Successfully....!");
                    }
                    System.out.println("-------------------------");
                break;
                case 5:
                    found=false;
```

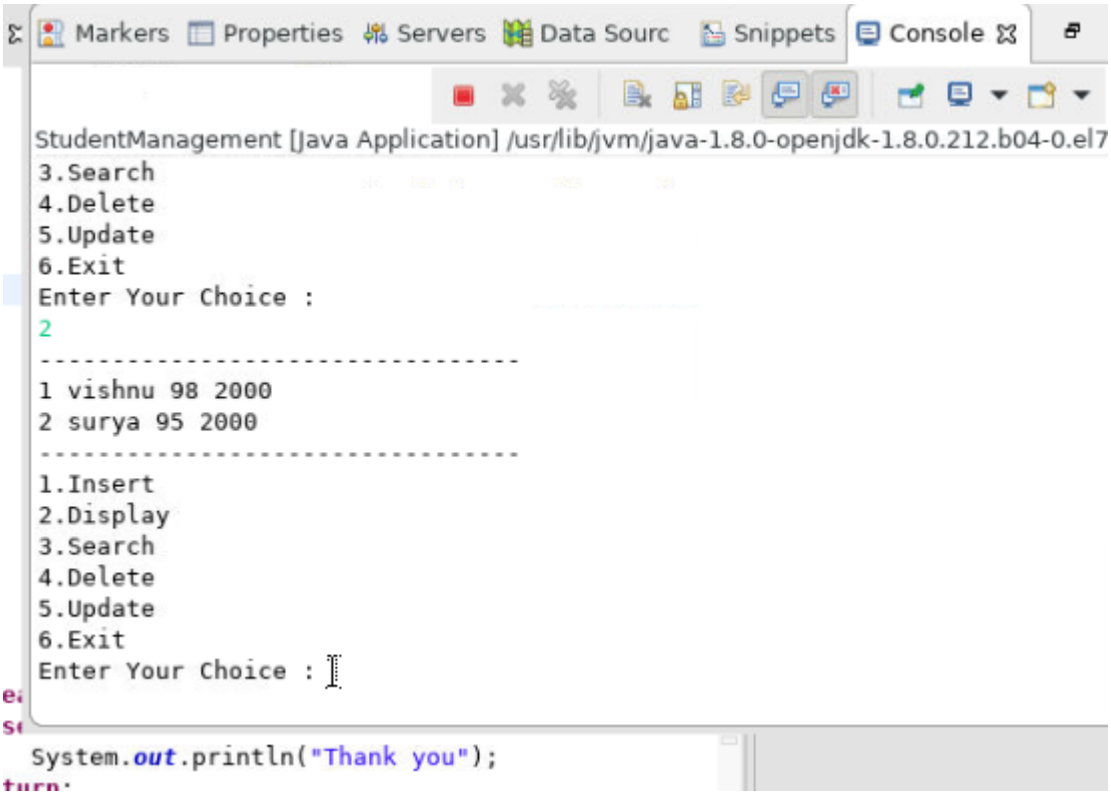
```
                    System.out.println("Enter Studentid to Update : ");
                    studentid =sc.nextInt();
                    System.out.println("-------------------------");
                    ListIterator<Student> li = s.listIterator();
                    while(li.hasNext())
                    {
                        Student e=li.next();
                        if(e.getStudentid()==studentid)
                        {
                            System.out.println("Enter new Name : ");
                            String studentname = sc1.nextLine();

                            System.out.println("Enter new Atd : ");
                            int studentatd = sc.nextInt();

                            System.out.println("Enter new Fees : ");
                            int studentfees = sc.nextInt();
                            li.set(new
Student(studentid,studentname,studentatd,studentfees));
                            found=true;
                        }
                    }
                    if(!found)
                    {
                        System.out.println("Record Not Found");
                    }
                    else
```

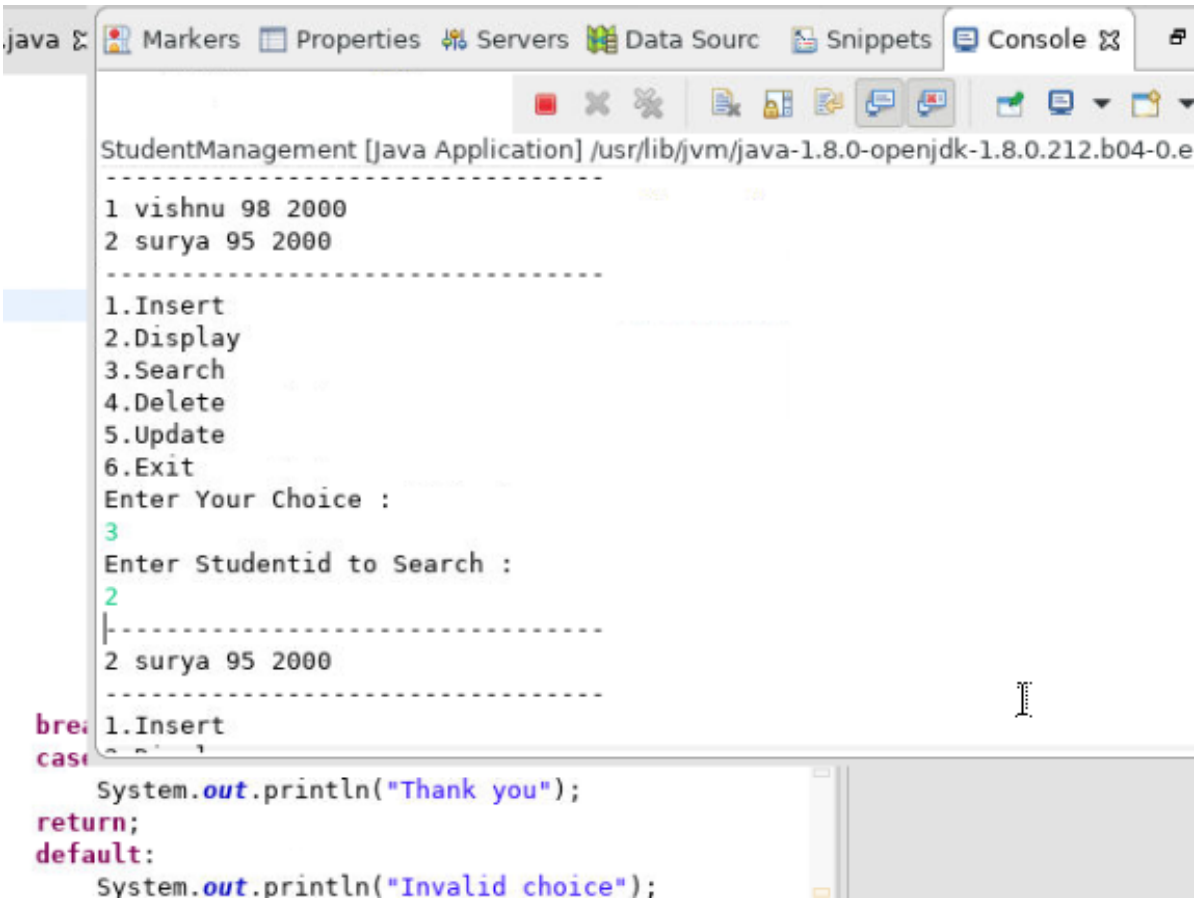
```
                    {
                        System.out.println("Record is Updated
Successfully....!");
                    }
                    System.out.println("-------------------------");
                break;
                case 6:
                    System.out.println("Thank you");
                return;
                default:
                    System.out.println("Invalid choice");
                break;
            }
        }while(ch!=0);
    }
}
```

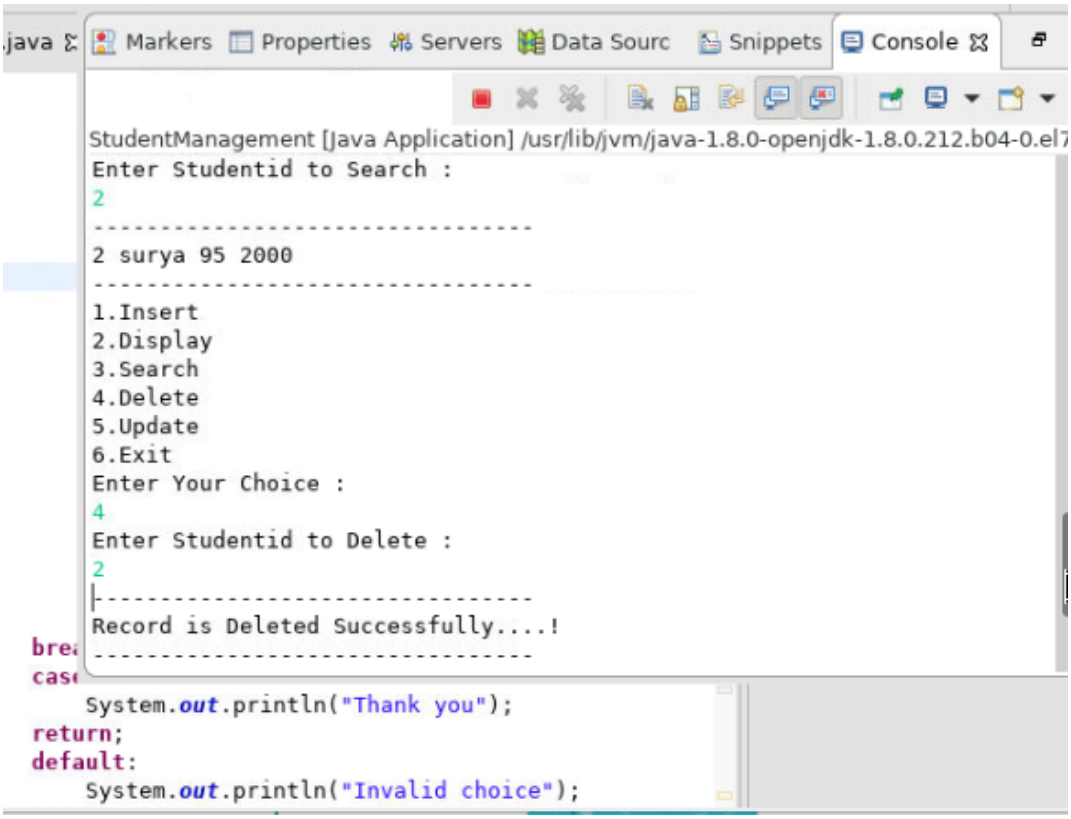
code output:

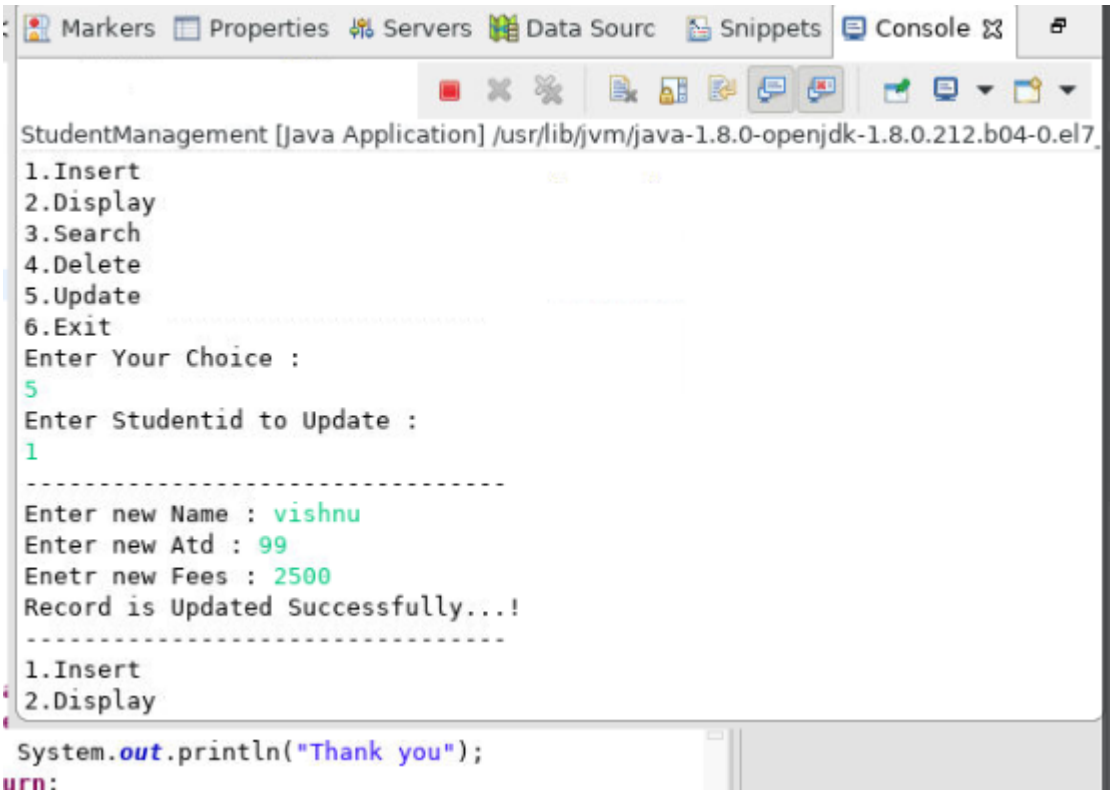


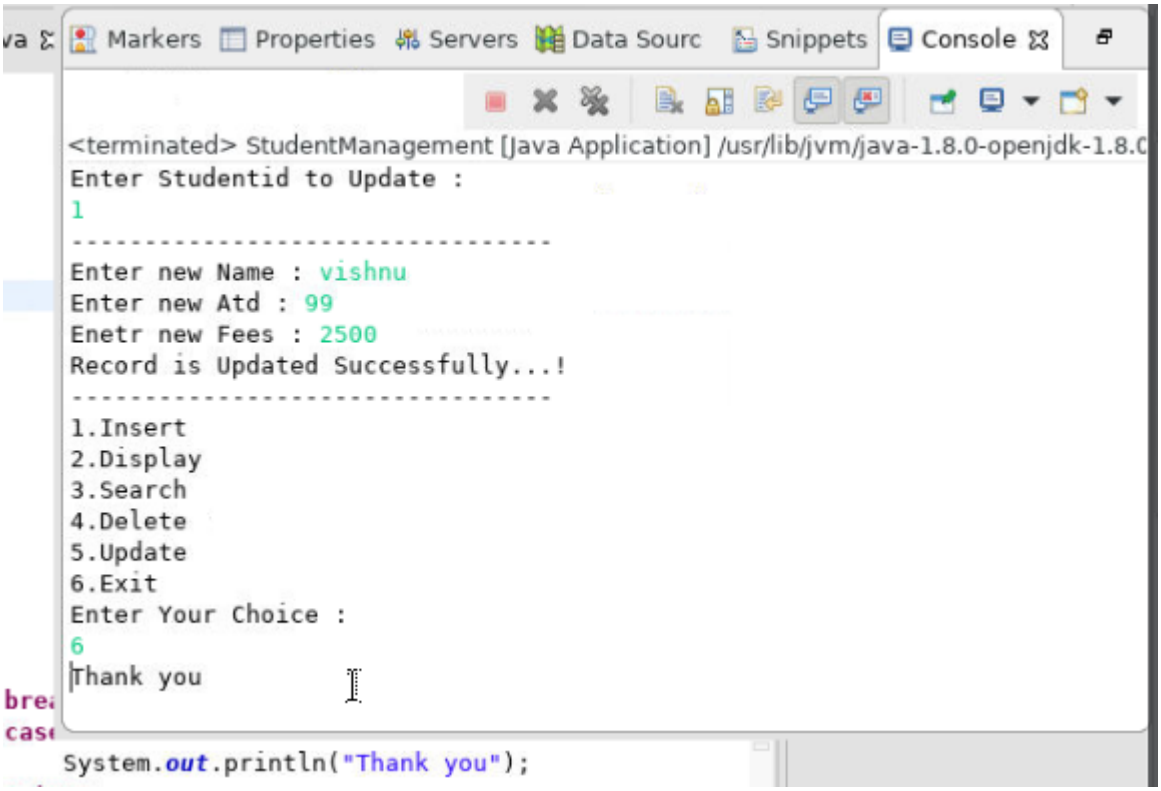

Markers
Properties
Servers
Data Sourc
Snippets
Console
StudentManagement [Java Application] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
2
1 vishnu 98 2000
1.Insert
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
System.out.println("Thank you");
return;
default:
System.out.println("Invalid choice");


Markers
Properties
Servers
Data Sourc
Snippets
Console
StudentManagement [Java Application] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
2
1 vishnu 98 2000
2 surya 95 2000
1.Insert
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
System.out.println("Thank you");


Markers
Properties
Servers
Data Sourc
Snippets
Console
StudentManagement [Java Application] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.e
1 vishnu 98 2000
2 surya 95 2000
1.Insert
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
3
Enter Studentid to Search :
2
2 surya 95 2000
1.Insert
System.out.println("Thank you");
return;
default:
System.out.println("Invalid choice");


Markers
Properties
Servers
Data Sourc
Snippets
Console
StudentManagement [Java Application] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7
Enter Studentid to Search :
2
2 surya 95 2000
1.Insert
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
4
Enter Studentid to Delete :
2
Record is Deleted Successfully....!
System.out.println("Thank you");
return;
default:
System.out.println("Invalid choice");


Markers
Properties
Servers
Data Sourc
Snippets
Console
StudentManagement [Java Application] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7
1.Insert
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
5
Enter Studentid to Update :
1
Enter new Name : vishnu
Enter new Atd : 99
Enetr new Fees : 2500
Record is Updated Successfully...!
1.Insert
2.Display
System.out.println("Thank you");


Markers
Properties
Servers
Data Sourc
Snippets
Console
<terminated> StudentManagement [Java Application] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0
Enter Studentid to Update :
1
Enter new Name : vishnu
Enter new Atd : 99
Enetr new Fees : 2500
Record is Updated Successfully...!
1.Insert
2.Display
3.Search
4.Delete
5.Update
6.Exit
Enter Your Choice :
6
Thank you
System.out.println("Thank you");