

EXPERIMENT 1 – DEPERSIST

```
package exp1;
import java.io.*;
public class Depersist
{    public static void main(String[] args)
    {        // TODO Auto-generated method stub
        try {
            FileInputStream fin = new FileInputStream("d:\\f.txt");
            ObjectInputStream in=new ObjectInputStream(fin);
            Student s=(Student)in.readObject();
            System.out.println(s);
            in.close(); }
        catch(Exception e)
        {    System.out.println(e); }
    } }
```

PERSIST

```
package exp1;
import java.io.*;
public class Persist
{    public static void main(String[] args)
    {        // TODO Auto-generated method stub
        try {
            Student s1 =new Student(1,"ram",10000.00f);
            FileOutputStream fout=new FileOutputStream("d:\\f.txt");
            ObjectOutputStream out=new ObjectOutputStream(fout);
            out.writeObject(s1);
            out.flush();
            out.close();
            System.out.println("success");
        }
        catch(Exception e)
        {
            System.out.println(e);
        } } }
```

STUDENT

```
package exp1;
import java.io.Serializable;
public class Student implements Serializable
{
    int rno;
    String name;
    float fees;
    public Student(int rno, String name, float fees)
    {
        this.rno = rno;
        this.name = name;
        this.fees = fees;
    }
    public String toString()
    { return rno + " " + name + " " + fees + "\n";
    } }
```

EXPERIMENT 2

EMPLOYEE

```
package exp2;
import java.io.*;
import java.util.*;
public class Employee {
    int eno;
    String ename;
    float salary;
    String desig;
    int age;
    static String cname;
    public Employee(int eno, String ename, float salary, String desig, int age) {
        this.eno = eno;
        this.ename = ename;
        this.salary = salary;
        this.desig = desig;
        this.age = age;
        cname = "PU"; }
    @Override
    public String toString() {
        return eno + " " + ename + " " + salary + " " + desig + " " + age + " " + cname + "\n"; } }
class AgeComparator implements Comparator {
    public int compare(Object o1, Object o2) {
        Employee e1 = (Employee) o1;
        Employee e2 = (Employee) o2;
        if (e1.age == e2.age)
            return 0;
        else if (e1.age > e2.age)
            return 1;
        else
            return -1; } }
class NameComparator implements Comparator {
    public int compare(Object o1, Object o2) {
        Employee e1 = (Employee) o1;
        Employee e2 = (Employee) o2;
        return e1.ename.compareTo(e2.ename); } }
class SalaryComparator implements Comparator {
    public int compare(Object o1, Object o2) {
        Employee e1 = (Employee) o1;
        Employee e2 = (Employee) o2;
        if (e1.salary == e2.salary)
            return 0;
        else if (e1.salary > e2.salary)
            return 1;
        else
            return -1; } }
```

PART - 2

```
package exp2;
import java.util.*;
public class Name {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ArrayList el=new ArrayList();
        el.add(new Employee(1,"Shiva",10000.00f,"JE",18));
        el.add(new Employee(2,"Venky",15000.00f,"SE",20));
        el.add(new Employee(3,"Jesus",17000.00f,"JE",19));
        el.add(new Employee(4,"Alla",12000.00f,"SE",19));
        el.add(new Employee(5,"Budha",11000.00f,"JE",21));
        System.out.println("Sorting by Name");
        System.out.println("_____ ");
        Collections.sort(el,new NameComparator());
        Iterator itr=el.iterator();
        while(itr.hasNext()){
            Employee e=(Employee)itr.next();
            System.out.println(e);
        }
        System.out.println("Sorting by age");
        System.out.println("_____ ");
        Collections.sort(el,new AgeComparator());
        Iterator itr2=el.iterator();
        while(itr2.hasNext()) {
            Employee e=(Employee)itr2.next();
            System.out.println(e);
        }
        System.out.println("Sorting by Salary");
        System.out.println("_____ ");
        Collections.sort(el,new SalaryComparator());
        Iterator itr1=el.iterator();
        while(itr1.hasNext()){
            Employee e=(Employee)itr1.next();
            System.out.println(e);
        }
    }
}
```

EXPERIMENT 3

```
package exp3;
import java.sql.*;
import java.util.*;
public class Emp1 {
    public static void main(String[] args) {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/vignesh?characterEncoding=latin1","root","root");
            Statement stmt=con.createStatement();
        int ans=1;
        do {
            System.out.println("1. Insert a record ");
            System.out.println("2. Delete a record ");
            System.out.println("3. Modify/Edit a record ");
            System.out.println("4. Display list of records ");
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter your choice:");
            int ch = sc.nextInt();
            String ename;
            int eno,age;
            String query="";
            switch(ch){
                case 1:
                    System.out.println("Enter employee number:");
                    eno = sc.nextInt();
                    System.out.println("Enter employee name:");
                    ename = sc.next();
                    System.out.println("Enter employee age:");
                    age = sc.nextInt();
                    query = "INSERT INTO emp " + "VALUES (" + eno+ "" + ename+"""+ age+"")";
                    stmt.executeUpdate(query);
                    break;
                case 2:
                    System.out.println("Enter employee number:");
                    eno = sc.nextInt();
                    query = "delete from emp where rno='"+eno+"'";
                    stmt.executeUpdate(query);
                    System.out.println("Record is deleted from the table successfully.....");
                    break;
                case 3:
                    PreparedStatement ps = null;
                    query = "update emp set name=? where rno=? ";
                    ps = con.prepareStatement(query);
                    System.out.println("Enter employee number:");
                    eno = sc.nextInt();
                    System.out.println("Enter employee name:");
                    ename = sc.next();
                    ps.setString(1, ename);
                    ps.setInt(2, eno);
                    ps.executeUpdate();
                    System.out.println("Record is updated successfully.....");
                    break;
                case 4:
                    ResultSet rs=stmt.executeQuery("select * from emp");
                    while(rs.next())
                        System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getInt(3));
}
        System.out.println("Enter another(1/0)");
        ans = sc.nextInt();
    }while(ans==1);

        con.close();
    }catch(Exception e){ System.out.println(e);}
}
```

EXPERIMENT 4

```
package exp4;
import java.util.*;

class Library{
    int bno;
    String bname;
    int duration;
    float amount;

    public Library(int bno, String bname, int duration, float amount) {
        super();
        this.bno = bno;
        this.bname = bname;
        this.duration = duration;
        this.amount = amount;
    }

    @Override
    public String toString() {
        return bno + " " + bname + " " + duration + " " + amount;
    }
}

public class Section {
    public static void main(String[] args) {
        List <Library> s = new ArrayList<Library>();
        s.add(new Library(1,"abc",20,20000.00f));
        s.add(new Library(2,"xyz",15,15000.00f));
        s.add(new Library(3,"def",10,10000.00f));

        System.out.println("Sorting on the basis of book name...");

        // implementing lambda expression
        Collections.sort(s,(s1,s2)->{return s1.bname.compareTo(s2.bname);});
        for(Library i:s){
            System.out.println(i);
        }
        System.out.println("Sorting by duration");

        Collections.sort(s,(s1,s2)-> s1.duration - s2.duration);
        s.forEach((l)->System.out.println(l));

        System.out.println("Sorting by Amount");

        Collections.sort(s,(s1,s2)-> (int)s1.amount - (int)s2.amount);
        s.forEach((l)->System.out.println(l));
    }
}
```