**Experiment1:**

**Aim:**

**To implement serialization with IO handling.**

**Program:**

**import** java.io.\*;

**import** java.util.\*;

**class** Employee **implements** Serializable {

**private** String name;

**private** **int** eno;

**private** **float** salary;

**private** **int** age;

**private** Date doj;

@SuppressWarnings("deprecation")

**public** Employee( ) {

name="";

eno=100;

salary=0.0f;

age = 0;

doj = **new** ~~Date~~(2023,01,01);

}

**public** Employee(**int** eno,String name,**float** salary,**int** age,Date doj) {

**this**.eno = eno;

**this**.name = name;

**this**.salary = salary;

**this**.age = age;

**this**.doj = doj;

}

**public** String getName() {

**return** name;

}

**public** **int** getEno() {

**return** eno;

}

**public** **void** setEno(**int** eno) {

**this**.eno = eno;

}

**public** **void** setName(String name ) {

**this**.name=name;

}

**public** **void** setSalary(**float** salary) {

**this**.salary = salary;

}

**public** **float** getSalary() {

**return** salary;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setDate(Date doj) {

**this**.doj = doj;

}

**public** Date getDate( ) {

**return** doj;

}

@Override

**public** String toString() {

**return** getEno() + " "+getName() + " " + getSalary() + " " + getAge() + " " + getDate() + "\n";

}

}

**public** **class** Temp {

@SuppressWarnings("deprecation")

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Employee e1 = **new** Employee(1,"God",100000.00f,45,**new** ~~Date~~(1970,8,15));

**try** {

FileOutputStream fos = **new** FileOutputStream("d:\\x.txt");

ObjectOutputStream oos = **new** ObjectOutputStream(fos);

oos.writeObject(e1);

fos.close();

oos.close();

}**catch**(Exception e) {

}

**try** {

FileInputStream fis = **new** FileInputStream("d:\\x.txt");

ObjectInputStream ois = **new** ObjectInputStream(fis);

e1=(Employee)ois.readObject();

System.***out***.println(e1);

fis.close();

ois.close();

}**catch**(Exception e) {

}

}

}

Experiment 2:

Problem statement:

Java Comparator example to sort Student object by name, age, and fees. In order to sort Student object on different criteria, we need to create multiple comparators e.g. NameComparator, AgeComparator, and FeesComparator, this is known as custom sorting in Java. This is different from the natural ordering of objects, provided by the compareTo() method of java.lang.Comparable interface. Though both compare() and compareTo() method looks similar they are different in the sense that, former accepts one parameter,

**Program:**

**import** java.io.\*;

**import** java.util.\*;

**class** Student {

**int** rollno;

String name;

**float** fees;

String branch;

**int** year;

**int** sem;

**int** age;

**static** String *clg*;

**public** Student(**int** rollno,String name,**float** fees,String branch,**int** year,**int** sem,**int** age) {

**this**.rollno = rollno;

**this**.name = name;

**this**.fees = fees;

**this**.branch = branch;

**this**.year = year;

**this**.sem = sem;

**this**.age = age;

*clg*="PU";

}

@Override

**public** String toString() {

**return** rollno + " "+ name + " " + fees + " " + branch + " " + year + sem + " " + age + " " + *clg* + "\n";

}

}

**class** AgeComparator **implements** Comparator {

**public** **int** compare(Object o1,Object o2){

Student s1=(Student)o1;

Student s2=(Student)o2;

**if**(s1.age==s2.age)

**return** 0;

**else** **if**(s1.age>s2.age)

**return** 1;

**else**

**return** -1;

}

}

**class** NameComparator **implements** Comparator{

**public** **int** compare(Object o1,Object o2){

Student s1=(Student)o1;

Student s2=(Student)o2;

**return** s1.name.compareTo(s2.name);

}

}

**class** FeesComparator **implements** Comparator {

**public** **int** compare(Object o1,Object o2){

Student s1=(Student)o1;

Student s2=(Student)o2;

**if**(s1.fees==s2.fees)

**return** 0;

**else** **if**(s1.fees>s2.fees)

**return** 1;

**else**

**return** -1;

}

}

**public** **class** Temp1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList sl=**new** ArrayList();

sl.add(**new** Student(1,"Shiva",10000.00f,"cse",1,1,18));

sl.add(**new** Student(2,"Venky",15000.00f,"ise",1,2,20));

sl.add(**new** Student(3,"Jesus",17000.00f,"ece",1,1,19));

sl.add(**new** Student(3,"Alla",12000.00f,"eee",1,1,19));

sl.add(**new** Student(3,"Budha",11000.00f,"mech",1,1,21));

System.***out***.println("Sorting by Name");

System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Collections.*sort*(sl,**new** NameComparator());

Iterator itr=sl.iterator();

**while**(itr.hasNext()){

Student st=(Student)itr.next();

System.***out***.println(st.rollno+" "+st.name+" "+ st.fees+ " " + st.branch+ " " + st.year + " " + st.sem + " " + st.age + " " + Student.*clg*);

}

System.***out***.println("Sorting by age");

System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Collections.*sort*(sl,**new** AgeComparator());

Iterator itr2=sl.iterator();

**while**(itr2.hasNext()){

Student st=(Student)itr2.next();

System.***out***.println(st.rollno+" "+st.name+" "+ st.fees+ " " + st.branch+ " " + st.year + " " + st.sem + " " + st.age + " " + Student.*clg*);

}

System.***out***.println("Sorting by fees");

System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Collections.*sort*(sl,**new** FeesComparator());

Iterator itr1=sl.iterator();

**while**(itr1.hasNext()){

Student st=(Student)itr1.next();

System.***out***.println(st.rollno+" "+st.name+" "+ st.fees+ " " + st.branch+ " " + st.year + " " + st.sem + " " + st.age + " " + Student.*clg*);

}

}

}