Assignment – 6

1.

**package** lamdaexpression.Assignments;

**import** java.util.TreeMap;

**import** java.util.Collections;

**import** java.util.Iterator;

**class** Contact

{

String name;

String email;

**long** phonenumber;

String gender;

**public** Contact(**long** phonenumber,String gender,String name,String email)

{

**this**.name=name;

**this**.email=email;

**this**.gender=gender;

**this**.phonenumber=phonenumber;

}

}

**public** **class** Phone {

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

TreeMap<Long,Contact> tree = **new** TreeMap<Long,Contact>(Collections.*reverseOrder*());

Contact obj1 = **new** Contact(9638527410L,"male","dhoni","dhoni@gmail.com");

Contact obj2 = **new** Contact(9638527411L,"male","kohli","kohli@gmail.com");

Contact obj3 = **new** Contact(9638527415L,"female","mandhana","mandhana@gmail.com");

tree.put(741852L,obj1);

tree.put(7418523L,obj2);

tree.put(741856L,obj3);

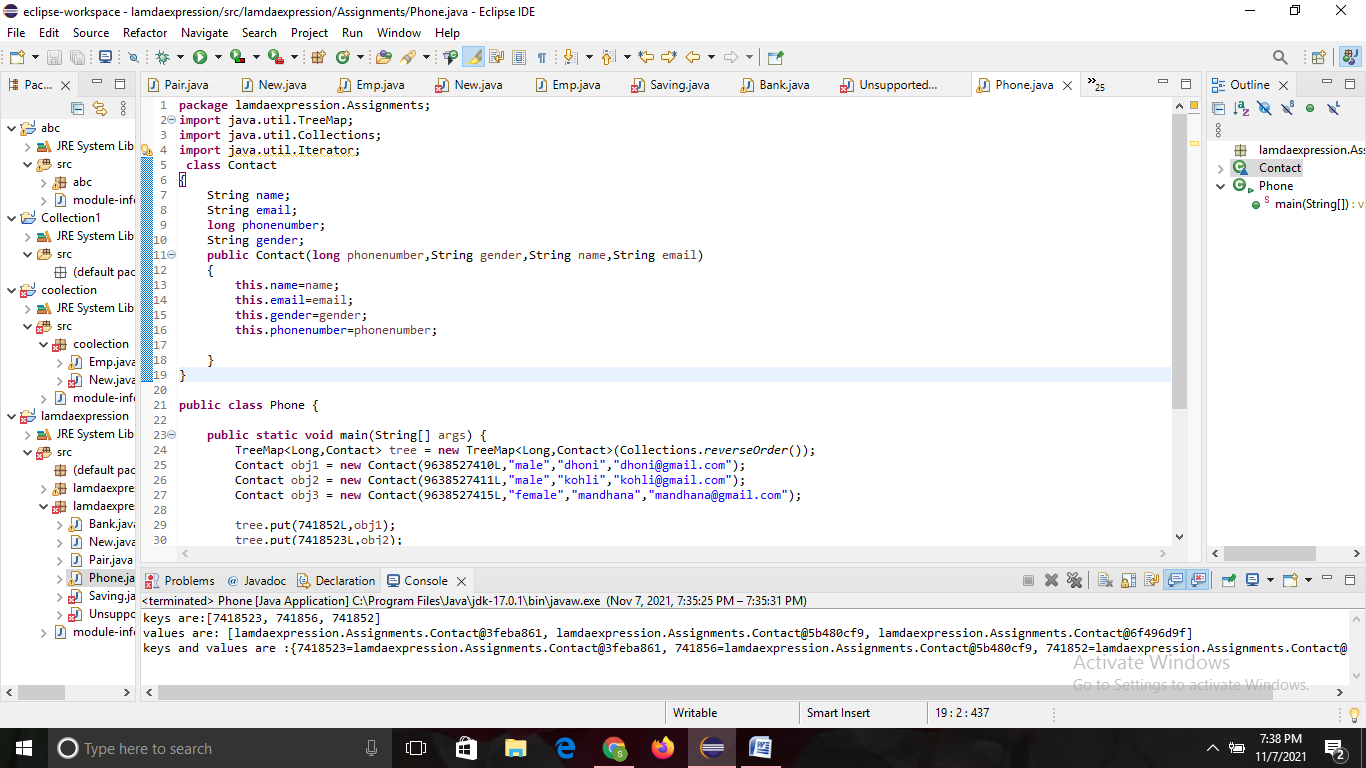
System.***out***.println("keys are:"+tree.keySet());

System.***out***.println("values are: "+tree.values());

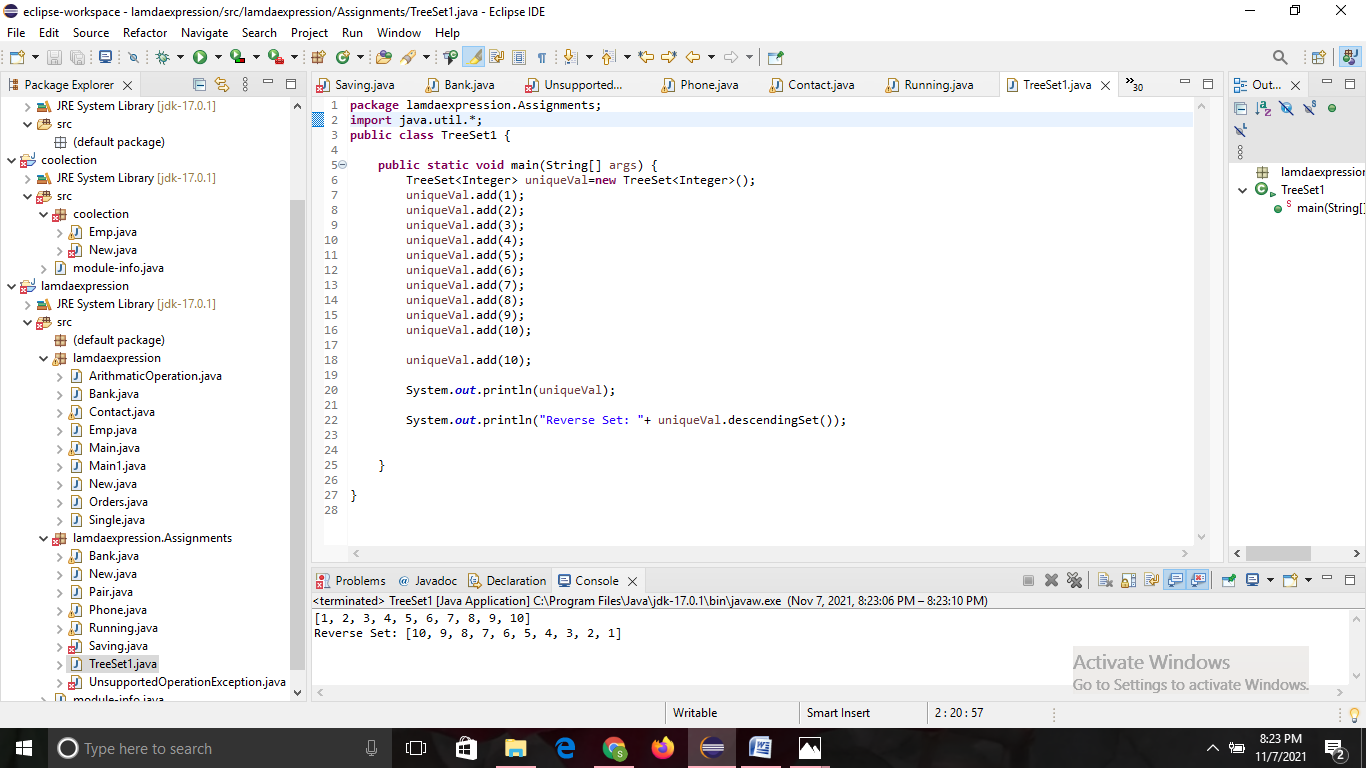
System.***out***.println("keys and values are :"+tree);

}

}



2.



3.

**package** lamdaexpression.Assignments;

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

**class** Employee{

**int** id;

String name ;

String department;

**double** salary;

**public** Employee(**int** id, String name, String department, **double** salary)

{

**this**.id=id;

**this**.name=name;

**this**.department= department;

**this**.salary= salary;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** **double** getSalary() {

**return** salary;

}

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

**this**.salary = salary;

}

}

**class** MynameComp **implements** Comparator<Employee>

{

@Override

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

**return** o1.getName().compareTo(o2.getName());

}

}

**public** **class** Hii {

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

TreeSet<Employee> e = **new** TreeSet<Employee>(**new** MynameComp());

Employee e1 = **new** Employee(1,"smriti","ECE",20000.26);

Employee e2 = **new** Employee(2,"annanya","cse",20100.58);

Employee e3 = **new** Employee(3,"pooja","ME",22000.96);

e.add(e1);

e.add(e2);

e.add(e3);

**for**(Employee s : e)

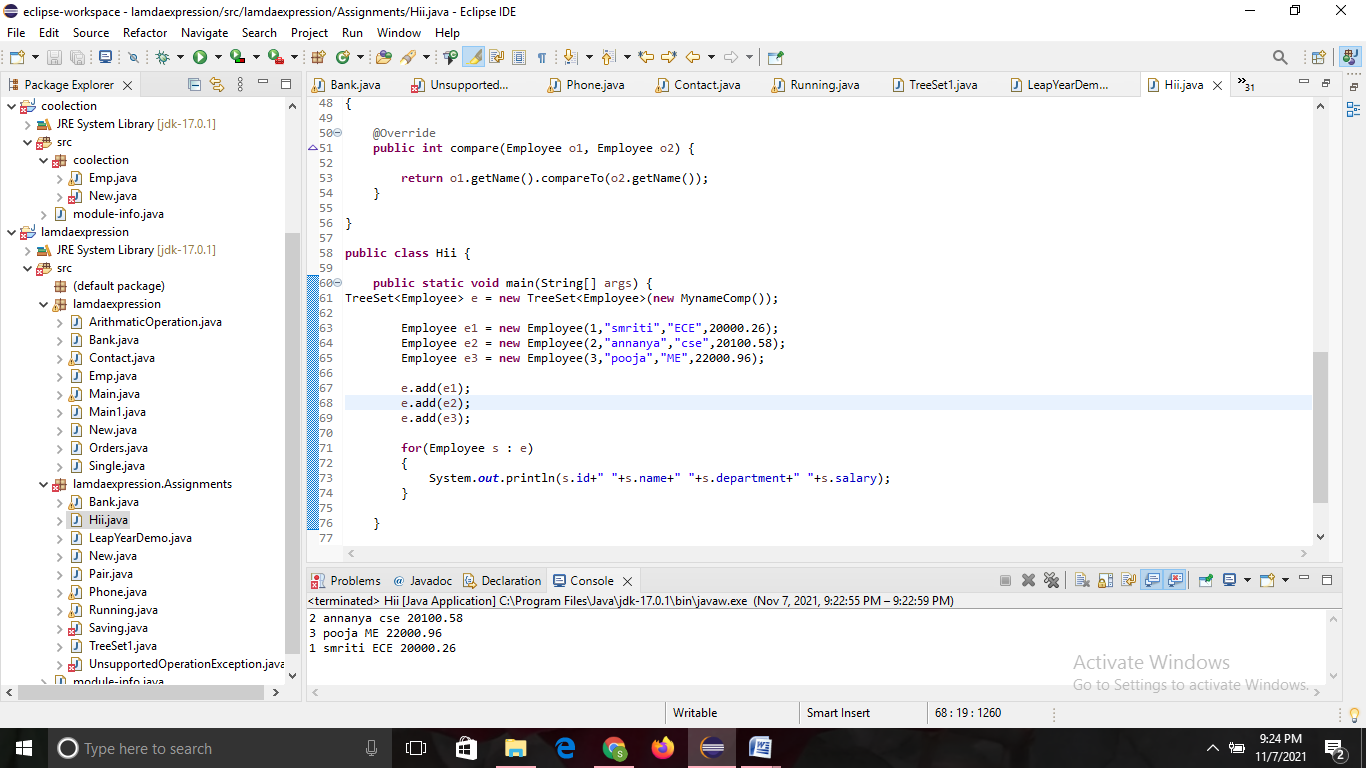
{

System.***out***.println(s.id+" "+s.name+" "+s.department+" "+s.salary);

}

}

}



4.

**package** lamdaexpression.Assignments;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeFormatter;

**import** java.util.LinkedList;

**import** java.util.List;

**public** **class** LeapYearDemo {

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

Dateex date = **new** Dateex("01/09/1999");

Dateex date1 = **new** Dateex("10/12/2000");

Dateex date2 = **new** Dateex("18/10/2003");

List<Dateex> dob = **new** LinkedList<>();

dob.add(date);

dob.add(date1);

dob.add(date2);

DateTimeFormatter df = DateTimeFormatter.*ofPattern*("dd/MM/yyyy");

**for** (**int** i = 0; i < dob.size(); i++) {

LocalDate up = LocalDate.*parse*(dob.get(i).date,df);

String s=(up).format(df);

**if**(up.getYear() % 4 == 0) {

System.***out***.println(s + " is an leap year");

}**else** {

System.***out***.println(s + " is not an leap year");

}

}

}

}

**class** Dateex {

String date;

**public** Dateex(String date) {

**super**();

**this**.date = date;

}

@Override

**public** String toString() {

**return** "[date=" + date + "]";

}

}

