**package** com.source;

**import** java.util.Arrays;

//EMPLOYEE -- ID,NAME,SAL

//SORT BASED ON ID ... ASC

**class** Emplyee **implements** Comparable<Emplyee>

{

**int** id;

String name;

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

**return** id;

}

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

**this**.id = id;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** Emplyee(**int** id, String name) {

**super**();

**this**.id = id;

**this**.name = name;

}

@Override

**public** **int** compareTo(Emplyee o) {

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

**return** id-o.id;

}

@Override

**public** String toString() {

**return** "Emplyee [id=" + id + ", name=" + name + "]";

}

}

**public** **class** Example

{

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

Emplyee[] emp=**new** Emplyee[5];

emp[0]=**new** Emplyee(1,"P");

emp[1]=**new** Emplyee(6,"A");

emp[2]=**new** Emplyee(12,"C");

emp[3]=**new** Emplyee(4,"Z");

emp[4]=**new** Emplyee(10,"B");

Arrays.*sort*(emp);

System.***out***.println(Arrays.*toString*(emp));

}}

**2 property sorting::::::::::::::**

**package** com.source;

**import** java.util.List;

**class** Person {

String fname;

**int** age;

**public** Person() {

}

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

**return** age;

}

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

**this**.age = age;

}

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** Person(String fname,**int** age) {

**this**.fname = fname;

**this**.age = age;

}

@Override

**public** String toString() {

**return** fname + "," + "," + age;

}

}

**public** **class** Main{

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

List<Person> persons = **new** java.util.ArrayList<Person>();

persons.add(**new** Person("abc3, 10));

persons.add(**new** Person("abc1,32));

persons.add(**new** Person("abc1, 65));

persons.add(**new** Person("abc1", 10));

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

persons.sort((p1, p2) -> {

**if** (p2.getFname().compareTo(p1.getFname()) == 0) {

**return** ((Integer)p2.getAge()).compareTo((Integer)p1.getAge());

} **else** {

**return** p2.getFname().compareTo(p1.getFname());

}

});

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

}

}

**Sorting on two properties using comparator:::**

**package** com.source;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.List;

**class** Person {

String fname;

String lname;

**int** age;

**public** Person() {

}

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

**return** age;

}

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

**this**.age = age;

}

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return** lname;

}

**public** **void** setLname(String lname) {

**this**.lname = lname;

}

**public** Person(String fname, String lname, **int** age) {

**this**.fname = fname;

**this**.lname = lname;

**this**.age = age;

}

@Override

**public** String toString() {

**return** fname + "," + lname + "," + age;

}

}

**public** **class** Main{

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

List<Person> persons = **new** java.util.ArrayList<Person>();

persons.add(**new** Person("abc3", "def3", 10));

persons.add(**new** Person("abc1", "def2", 32));

persons.add(**new** Person("abc1", "def1", 65));

persons.add(**new** Person("abc1", "def4", 10));

Collections.*sort*(persons,**new** Comparator(){

**public** **int** compare(Person p1, Person p2) {

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

**if** (p2.getFname().compareTo(p1.getFname()) == 0) {

**return** ((Integer)p2.getAge()).compareTo((Integer)p1.getAge());

} **else** {

**return** p2.getFname().compareTo(p1.getFname());

}

}

@Override

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

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

**return** 0;

}

});

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

}

}

**Sort Map by Value :::**

**package** com.source;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.LinkedList;

**import** java.util.List;

**import** java.util.Map;

**public** **class** Main{

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

Map persons = **new** HashMap ();

persons.put( "Amma",1020);

persons.put( "a",1150);

persons.put( "m",10);

persons.put("b",120);

System.***out***.println("Unsort Map......");

*printMap*(persons);

System.***out***.println("\nSorted Map......By Value");

Map<String, Integer> personsMap = *sortByValue*(persons);

*printMap*(personsMap);

}

**private** **static** Map<String, Integer> sortByValue(Map<String, Integer> personsMap) {

// 1. Convert Map to List of Map

List<Map.Entry<String, Integer>> list =

**new** LinkedList<Map.Entry<String, Integer>>(personsMap.entrySet());

// 2. Sort list with Collections.sort(), provide a custom Comparator

// Try switch the o1 o2 position for a different order

Collections.*sort*(list, **new** Comparator<Map.Entry<String, Integer>>() {

**public** **int** compare(Map.Entry<String, Integer> o1,

Map.Entry<String, Integer> o2) {

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

}

});

// 3. Loop the sorted list and put it into a new insertion order Map LinkedHashMap

Map<String, Integer> personMap = **new** LinkedHashMap<String, Integer>();

**for** (Map.Entry<String, Integer> entry : list) {

personMap.put(entry.getKey(), entry.getValue());

}

**return** personMap;

}

**public** **static** <K, V> **void** printMap(Map<K, V> map) {

**for** (Map.Entry<K, V> entry : map.entrySet()) {

System.***out***.println("Key : " + entry.getKey()

+ " Value : " + entry.getValue());

}

}

}