**Exception Handling**

**Exercise 1:***Create a class called Employee which asks the user to input the name and the age of a*

*employee. Raise a custom defined exception when the user enters an employee name*

*that has been already entered and raise another exception if the age is negative or less*

*than 18 or greater than 60.*

*Ans:*

***//StoringName***

***package com.customexception;***

***public class StoringName {***

***String[] arr=new String[]{"Suhel","Aman","Nikhil","Pruthvi","Sumanth"};***

***}***

***//AgeException***

***package com.customexception;***

***public class AgeException extends StoringName{***

***private String name;***

***private int age;***

***public String getName() {***

***return name;***

***}***

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

***return age;***

***}***

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

***for(int i=0;i<arr.length-1;i++) {***

***if(arr[i].equals(this.name))***

***throw new AgeCustExcep("Name already Exists");***

***}***

***this.name = name;***

***}***

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

***if(age<18 || age>60)***

***throw new AgeCustExcep("Age is invalid");***

***this.age = age;***

***}***

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

***try {***

***if(age<18 || age>60)***

***throw new AgeCustExcep("Age is invalid");***

***this.age=age;***

***}***

***catch(AgeCustExcep ex) {***

***System.out.println("please enter Valid age");***

***}***

***}***

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

***try {***

***for(int i=0;i<arr.length-1;i++) {***

***if(arr[i].equals(this.name)) {***

***throw new AgeCustExcep("Name already Exsist");***

***}***

***this.name=name;***

***}***

***}***

***catch(AgeCustExcep ex) {***

***System.out.println(" Name already exists Please enter new name");***

***}***

***}***

***}***

***//AgeCustExcep***

***package com.customexception;***

***public class AgeCustExcep extends Exception{***

***//Checked Exception***

***public AgeCustExcep(String message) {***

***super(message);***

***}***

***}***

***//MainAge***

***package com.customexception;***

***import java.util.Scanner;***

***public class MainAge {***

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

***Scanner scanner=new Scanner(System.in);***

***AgeException aOb=new AgeException();***

***System.out.println("Enter the name");***

***String name=scanner.next();***

***try {***

***aOb.setName(name);***

***} catch (AgeCustExcep e) {***

***System.out.println("Already existing name");***

***}***

***aOb.oneMoreSetName(name);***

***System.out.println("Enter Age : ");***

***int age=scanner.nextInt();***

***try {***

***aOb.setAge(age);***

***System.out.println("Person is eligible for voting");***

***} catch (AgeCustExcep e) {***

***System.out.println("As person age is "+age+" so not eligible for voting");***

***}***

***aOb.oneMoreSetAge(age);***

***}***

***}***

**Collection**

**Exercise 2:***Create a collection that will contain the names of the days in a week. Print the collection.*

*Display the length of the collection and convert the collection into an array and print it.*

*Ans:*

***package Collection;***

***import java.util.ArrayList;***

***import java.util.List;***

***public class NamesofDaysInWeek {***

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

***List<String> list=new ArrayList<String>();***

***list.add("Sun Day");***

***list.add("Mon Day");***

***list.add("TuesDay");***

***list.add("Wednesday");***

***list.add("Thursday");***

***list.add("Friday");***

***list.add("Saturday");***

***for(String s:list)System.out.println(s);***

***System.out.println("=====================");***

***System.out.println("Length of Collection : "+list.size());***

***String arr[]=new String[list.size()];***

***for(int i=0;i<list.size();i++) arr[i]=list.get(i);***

***System.out.println("=====================");***

***for(String s:arr) System.out.println(s);***

***}***

***}***

**Exercise 3:***Write a program to implement a telephone directory. Display the details.*

**Solution Guidance:**  *Name Phone no*

*ABC 1234*

*DEF 5678*

***package Collection;***

***import java.util.Iterator;***

***import java.util.Map;***

***import java.util.Set;***

***import java.util.TreeMap;***

***public class TelephoneDirectory {***

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

***Map<String,Integer> map=new TreeMap<String,Integer>();***

***map.put("ABC",785785788);***

***map.put("DEF",784645465);***

***map.put("MNO",984657457);***

***map.put("XYZ",879565875);***

***Set set=map.keySet();***

***Iterator iterator=set.iterator();***

***while(iterator.hasNext())***

***{***

***Object key=iterator.next();***

***Integer value=map.get(key);***

***System.out.println(key+" "+value);***

***}***

***}***

***}***

**Exercise 4:***Create a program to depict the usage of the dictionary where words along with the meanings are stored. When the user gives a word, its meaning should be displayed.*

*Ans:*

***package Collection;***

***import java.util.Iterator;***

***import java.util.Map;***

***import java.util.Scanner;***

***import java.util.Set;***

***import java.util.TreeMap;***

***public class Dictionary {***

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

***Map<String,String> map=new TreeMap<String,String>();***

***map.put("Concrete","A method wich has an implimentation");***

***map.put("Factory","A method which has Object creation logic");***

***map.put("Recursion","Process in which method calss itself continously");***

***map.put("Array","An Object which contains elements of similar data types");***

***Scanner scn=new Scanner(System.in);***

***System.out.println("Enter a word");***

***Set set=map.keySet();***

***Iterator iterator=set.iterator();***

***while(iterator.hasNext())***

***{***

***String word=scn.next();***

***String value=map.get(word);***

***System.out.println("Meaning of given word is : "+value);***

***}***

***}***

*}*

**Exercise 5:***Create a class called CD whose attributes are Title and singer.*

1. *Arrange the CDs in ascending order based on the singer name*
2. *Arrange the CDs in decending order Based on Title*

Ans:

**public class CD {**

**String title;**

**String singer;**

**public CD(String title, String singer) {**

**this.title = title;**

**this.singer = singer;**

**}**

**@Override**

**public String toString() {**

**return " title=" + title + ", singer=" + singer +"" ;**

**}**

**}**

**import java.util.Comparator;**

**public class SortCdBySinger implements Comparator<CD> {**

**@Override**

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

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

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

**}**

**}**

**import java.util.Comparator;**

**public class SortCdByTitle implements Comparator<CD>{**

**@Override**

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

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

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

**}**

**}**

**import java.util.TreeSet;**

**public class SortCdMain {**

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

**CD cd1=new CD("KGF","Vijayprakash");**

**CD cd2=new CD("Tagaru","Ravi Basur");**

**CD cd3=new CD("Seetha rama kalyana","Arman malik");**

**CD cd4=new CD("Puspa","Sid sri ram");**

**CD cd5=new CD("The villan","Shankar");**

**System.out.println("Sorting in Descending order ");**

**TreeSet<CD> t=new TreeSet<CD>(new SortCdByTitle());**

**t.add(cd1);**

**t.add(cd2);**

**t.add(cd3);**

**t.add(cd4);**

**t.add(cd5);**

**for(CD ce:t) {**

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

**}**

**System.out.println("------------------------------");**

**System.out.println("Sorting in Ascending order");**

**TreeSet<CD> t1=new TreeSet<CD>(new SortCdBySinger());**

**t1.add(cd1);**

**t1.add(cd2);**

**t1.add(cd3);**

**t1.add(cd4);**

**t1.add(cd5);**

**for(CD ce1:t1) {**

**System.out.println(ce1);**

**}**

**}**

**}**