DAY 9

1. **Sort Students by Roll.No**

```
import java.util.*;
class Student implements Comparable<student>{</student>
int rollNo; String name; int marks;
Student(int r,String n,int m){
rollNo=r;name=n;marks=m;
}
public int compareTo(Student s){
return rollNo-s.rollNo;
}
public String toString(){return rollNo+" "+name+" "+marks;}
}
public class Main1{
public static void main(String[] args){
List<student> list=new ArrayList&#x3C;>();</student>
list.add(new Student(4,"Harshu",85));
list.add(new Student(2,"Thejas",95));
list.add(new Student(1,"Dhanya",80));
Collections.sort(list);
for(Student s : list) System.out.println(s);
}
}
```

Output

1 Dhanya 80
2 Thejas 95
4 Harshu 85

2. **Product by price**

```
import java.util.*;
class Product implements Comparable<product>{</product>
String name; double price;
Product(String n,double p){name=n;price=p;
}
public int compareTo(Product p)
{
return Double.compare(price,p.price);
}
public String toString(){return name+" "+price;}
}
public class Main2{
public static void main(String[] args){
List<product> list=new ArrayList&#x3C;>();</product>
list.add(new Product("Item1",250));
list.add(new Product("Item2",120));
list.add(new Product("Item3",180));
Collections.sort(list);
for(Product p : list) System.out.println(p);
}
}
```

Output
Item2 120.0
Item3 180.0
Item1 250.0

3. **Employee by name**

```
import java.util.*;
class Employee implements Comparable<employee>{</employee>
String name; Employee(String n){name=n;}
public int compareTo(Employee e){return name.compareTo(e.name);}
public String toString(){return name;}
}
public class Main3{
public static void main(String[] args){
List<employee> list=new ArrayList&#x3C;>();</employee>
list.add(new Employee("Harshu"));
list.add(new Employee("Anu"));
list.add(new Employee("Yakshith"));
Collections.sort(list);
for(Employee e : list) System.out.println("Employee: " + e);
}
}
```

Output
Employee: Anu
Employee: Harshu
Employee: Yakshith

4. **Books by bookId descending**

```
import java.util.*;
class Book implements Comparable<book>{</book>
int bookId; String title;
Book(int id,String t){
bookId=id;title=t;
}
public int compareTo(Book b){
return b.bookId-bookId;
}
public String toString(){return bookId+" "+title;}
}
public class Main4{
public static void main(String[] args){
List<book> list=new ArrayList&#x3C;>();</book>
list.add(new Book(5,"TitleB"));
list.add(new Book(3,"TitleC"));
list.add(new Book(4,"TitleA"));
Collections.sort(list);
System.out.println("Sorted Books: " + list);
}
}
```

Output
Sorted Books: [5 TitleB, 4 TitleA, 3 TitleC]

5. **Sort custom objects & show before/after**

```
import java.util.*;
class Item implements Comparable<item>{</item>
int id; String name;
Item(int i,String n){
id=i;name=n;
}
public int compareTo(Item o){
return id-o.id;
}
public String toString(){return id+" "+name;}
}
public class Main5{
public static void main(String[] args){
List<item> list=new ArrayList&#x3C;>();</item>
list.add(new Item(6,"Notebook"));
list.add(new Item(4,"Eraser"));
list.add(new Item(5,"Marker"));
System.out.println("Before: " + list);
Collections.sort(list);
System.out.println("After: " + list);
}
}
```

Output

Before: [6 Notebook, 4 Eraser, 5 Marker]

After: [4 Eraser, 5 Marker, 6 Notebook]

6. **Students by marks descending**

```
import java.util.*;
class Student2{
String name; int marks;
Student2(String n,int m){name=n;marks=m;}
public String toString(){return name+" "+marks;}
}
public class Main6{
public static void main(String[] args){
List<student2> list=new ArrayList&#x3C;>();</student2>
list.add(new Student2("X",75));
list.add(new Student2("Y",95));
list.add(new Student2("Z",88));
list.sort((a,b)->b.marks-a.marks);
System.out.println("Top Students: " + list);
}
}
```

Output

Top Students: [Y 95, Z 88, X 75]

7. **Product multiple sorting strategies**

```
import java.util.*;
class Product2{
String name; double price;
Product2(String n,double p){name=n;price=p;}
public String toString(){return name+" "+price;}
}
public class Main7{
```

```
public static void main(String[] args){
List<product2> list=new ArrayList&#x3C;>();</product2>
list.add(new Product2("ProdB",160));
list.add(new Product2("ProdA",220));
list.add(new Product2("ProdC",110));
list.sort(Comparator.comparingDouble(p->p.price));
System.out.println("By Price Asc: " + list);
list.sort((a,b)->Double.compare(b.price,a.price));
System.out.println("By Price Desc: " + list);
list.sort(Comparator.comparing(p->p.name));
System.out.println("By Name: " + list);
}
}
```

Output

By Price Asc: [ProdC 110.0, ProdB 160.0, ProdA 220.0]
By Price Desc: [ProdA 220.0, ProdB 160.0, ProdC 110.0]
By Name: [ProdA 220.0, ProdB 160.0, ProdC 110.0]

8. **Employee by joining date**

```
import java.util.*;
import java.time.*;
class Emp{
String name; LocalDate date;
Emp(String n,LocalDate d){
name=n;date=d;
}
public String toString(){
return name+" "+date;
}
}
public class Main8{
public static void main(String[] args){
List<emp> list=new ArrayList&#x3C;>();</emp>
list.add(new Emp("P",LocalDate.of(2023,6,2)));
list.add(new Emp("Q",LocalDate.of(2021,4,12)));
list.add(new Emp("R",LocalDate.of(2022,8,18)));
list.sort(Comparator.comparing(e->e.date));
for(Emp e : list) System.out.println("Joined: " + e);
}
}
```

Output

Joined: Q 2021-04-12
Joined: R 2022-08-18
Joined: P 2023-06-02

9. **Cities by population**

```
import java.util.*;
class City{
String name; int pop;
City(String n,int p){
name=n;pop=p;
}
public String toString(){return name+" "+pop;}
```

```
}
public class Main9{
public static void main(String[] args){
List<city> list=new ArrayList&#x3C;>();</city>
list.add(new City("CityX",6000));
list.add(new City("CityY",3000));
list.add(new City("CityZ",9000));
list.sort((a,b)->b.pop-a.pop);
System.out.println("Populated Cities: " + list);
}
}
```

Output

Populated Cities: [CityZ 9000, CityX 6000, CityY 3000]

10. **Strings by length**

```
import java.util.*;
public class Main10{
public static void main(String[] args){
List<string> list=Arrays.asList("aaaa","bb","ccccc");</string>
list.sort(new Comparator<string>(){</string>
public int compare(String a,String b){return a.length()-b.length();}
});
for(String s : list) System.out.println("String: " + s);
}
}
```

Output

String: bb

String: aaaa

String: ccccc

11. **Student Comparable by name, Comparator by marks**

```
import java.util.*;
class Stu implements Comparable<stu>{</stu>
String name; int marks;
Stu(String n,int m){name=n;marks=m;}
public int compareTo(Stu s){
return name.compareTo(s.name);
}
public String toString(){return name+" "+marks;}
}
public class Main11{
public static void main(String[] args){
List<stu> list=new ArrayList&#x3C;>();</stu>
list.add(new Stu("Dave",88));
list.add(new Stu("Eve",92));
list.add(new Stu("Frank",82));
Collections.sort(list);
System.out.println("By Name: " + list);
list.sort((a,b)->b.marks-a.marks);
System.out.println("By Marks: " + list);
}
}
```

Output
By Name: [Dave 88, Eve 92, Frank 82]
By Marks: [Eve 92, Dave 88, Frank 82]

12. **Book (ID,Title,Author)**

```
import java.util.*;
class Book2 implements Comparable<book2>{</book2>
int id; String title,author;
Book2(int i,String t,String a){id=i;title=t;author=a;
}
public int compareTo(Book2 b){
return id-b.id;
}
public String toString(){return id+" "+title+" "+author;}
}
public class Main12{
public static void main(String[] args){
List<book2> list=new ArrayList&#x3C;>();</book2>
list.add(new Book2(5,"Go","AuthorX"));
list.add(new Book2(4,"Rust","AuthorY"));
list.add(new Book2(6,"Swift","AuthorZ"));
Collections.sort(list);
System.out.println("By ID: " + list);
list.sort(Comparator.comparing((Book2 b)->b.title).thenComparing(b->b.author));
System.out.println("By Title then Author: " + list);
}
}
```

Output
By ID: [4 Rust AuthorY, 5 Go AuthorX, 6 Swift AuthorZ]
By Title then Author: [5 Go AuthorX, 4 Rust AuthorY, 6 Swift AuthorZ]

13. **Menu-driven Employee sort**

```
import java.util.*;
class Emp2{
String name; double salary; String dept;
Emp2(String n,double s,String d){name=n;salary=s;dept=d;}
public String toString(){return name+" "+salary+" "+dept;}
}
public class Main13{
public static void main(String[] args){
List<emp2> list=new ArrayList&#x3C;>();</emp2>
list.add(new Emp2("EmpA",32000,"Finance"));
list.add(new Emp2("EmpB",42000,"Sales"));
list.add(new Emp2("EmpC",37000,"Marketing"));
Scanner sc=new Scanner(System.in);
int ch=sc.nextInt();
if(ch==1)list.sort(Comparator.comparing(e->e.name));
else if(ch==2)list.sort(Comparator.comparingDouble(e->e.salary));
else if(ch==3)list.sort(Comparator.comparing(e->e.dept));
for(Emp2 e : list) System.out.println("Employee Info: " + e);
}
}
```

Output
Employee Info: EmpA 32000.0 Finance
Employee Info: EmpC 37000.0 Marketing
Employee Info: EmpB 42000.0 Sales

14. **Comparator.comparing() method references**

```
import java.util.*;
class Person{
String name; int age;
Person(String n,int a){name=n;age=a;}
public String getName() { return name; }
public String toString(){return name+" "+age;}
}
public class Main14{
public static void main(String[] args){
List<person> list=new ArrayList&#x3C;>();</person>
list.add(new Person("PersonA",28));
list.add(new Person("PersonB",22));
list.sort(Comparator.comparing(Person::getName));
System.out.println("Sorted Persons: " + list);
}
}
```

Output
Sorted Persons: [PersonA 28, PersonB 22]

15. **TreeSet custom comparator by age**

```
import java.util.*;
class Person2{
String name; int age;
Person2(String n,int a){name=n;age=a;}
public String toString(){return name+" "+age;}
}
public class Main15{
public static void main(String[] args){
Set<person2> set=new TreeSet&#x3C;>((a,b)->a.age-b.age);</person2>
set.add(new Person2("IndA",28));
set.add(new Person2("IndB",22));
set.add(new Person2("IndC",33));
for(Person2 p : set) System.out.println("Person: " + p);
}
}
```

Output
Person: IndB 22
Person: IndA 28
Person: IndC 33

File Handling & Serialization

1. **Create and write to student.txt**

```
import java.io.*;
public class FH1{
public static void main(String[] args)throws Exception{
FileWriter fw=new FileWriter("student.txt");
```

```
fw.write("ABC\nEFG\nHIJ\nLMN\nOPQ\n");
fw.close();
}
}
```

Output (content of student.txt)

ABC
EFG
HIJ
LMN
OPQ

2. **Read student.txt**

```
import java.io.*;
public class FH2{
public static void main(String[] args)throws Exception{
BufferedReader br=new BufferedReader(new FileReader("student.txt"));
String line;
while((line=br.readLine())!=null) System.out.println("Name: " + line);
br.close();
}
}
```

Output

Name: ABC
Name: EFG
Name: HIJ
Name: LMN
Name: OPQ

3 Append to student.txt

```
import java.io.*;
public class FH3{
public static void main(String[] args)throws Exception{
FileWriter fw=new FileWriter("student.txt",true);
fw.write("XYZ\n");
fw.close();
}
}
```

Output

ABC
EFG
HIJ
LMN
OPQ
XYZ

4. **Count words & lines in notes.txt**

```
import java.io.*;
public class FH4{
public static void main(String[] args)throws Exception{
BufferedReader br=new BufferedReader(new FileReader("notes.txt"));
String line; int lines=0,words=0;
while((line=br.readLine())!=null){
lines++; words+=line.split("\s+").length;
```

```
}
br.close();
System.out.println("Total Lines: "+lines);
System.out.println("Total Words: "+words);
}
}
```

Output

Total Lines: 3

Total Words: 15

5. **Copy contents from source.txt to destination.txt**

```
import java.io.*;
public class FH5{
public static void main(String[] args)throws Exception{
BufferedReader br=new BufferedReader(new FileReader("source.txt"));
FileWriter fw=new FileWriter("destination.txt");
String line;
while((line=br.readLine())!=null){
fw.write(line+"\n");
}
br.close(); fw.close();
}
}
```

6. **Check if report.txt exists and show properties**

```
import java.io.*;
public class FH6{
public static void main(String[] args){
File f=new File("report.txt");
if(f.exists()){
System.out.println("Path: " + f.getAbsolutePath());
System.out.println("Name: " + f.getName());
System.out.println("Writable: " + f.canWrite());
System.out.println("Readable: " + f.canRead());
System.out.println("Size: " + f.length());
}else System.out.println("File does not exist");
}
}
```

Output

Path: /full/path/report.txt

Name: report.txt

Writable: true

Readable: true

Size: 120

7. **Create file and accept user input**

```
import java.io.; import java.util.;
public class FH7{
public static void main(String[] args)throws Exception{
Scanner sc=new Scanner(System.in);
FileWriter fw=new FileWriter("userinput.txt");
fw.write(sc.nextLine() + "\n");
fw.close();
}
}
```

Output Hello World File

8. **Reverse file content**

```
import java.io.; import java.util.;
public class FH8{
public static void main(String[] args)throws Exception{
List<string> lines=new ArrayList&#x3C;>();</string>
BufferedReader br=new BufferedReader(new FileReader("data.txt"));
String line;
while((line=br.readLine())!=null)lines.add(line);
br.close();
FileWriter fw=new FileWriter("reversed.txt");
for(int i=lines.size()-1;i>=0;i--)fw.write(lines.get(i)+"\n");
fw.close();
}
}
```

Output
lines in reverse order.

9. **Serialize Student object**

```
import java.io.*;
class StuS implements Serializable{
int id; String name; int marks;
StuS(int i,String n,int m){id=i;name=n;marks=m;}
}
public class FH9{
public static void main(String[] args)throws Exception{
ObjectOutputStream oos=new ObjectOutputStream(new FileOutputStream("student.ser"));
oos.writeObject(new StuS(2,"EFG",95));
oos.close();
}
}
```

Output
student.ser created with object data.

10. **Deserialize Student object**

```
import java.io.*;
public class FH10{
public static void main(String[] args)throws Exception{
ObjectInputStream ois=new ObjectInputStream(new FileInputStream("student.ser"));
StuS s=(StuS)ois.readObject();
ois.close();
System.out.println("Deserialized: " + s.id+" "+s.name+" "+s.marks);
}
}
```

Output
Deserialized: 2 EFG 95

11. **Print all files in a directory**

```
import java.io.; import java.util.;
public class FH11{
public static void main(String[] args){
Scanner sc=new Scanner(System.in);
File dir=new File(sc.nextLine());
```

```
File[] files=dir.listFiles(File::isFile);
for(File f:files) System.out.println("File: " + f.getName());
}
}
```

Output

File: file1.txt

File: file2.java

File: notes.txt

12. **Delete a file**

```
import java.io.*;
public class FH12{
public static void main(String[] args){
File f=new File("delete.txt");
if(f.exists())System.out.println("Deleted: " + f.delete());
else System.out.println("File not found");
}
}
```

Output

Deleted: true

13. **Search word in file**

```
import java.io.; import java.util.;
public class FH13{
public static void main(String[] args)throws Exception{
Scanner sc=new Scanner(System.in);
String word=sc.next();
BufferedReader br=new BufferedReader(new FileReader("notes.txt"));
String line; boolean found=false;
while((line=br.readLine())!=null){
if(line.contains(word)){found=true;break;}
}
br.close();
System.out.println("Found: " + found);
}
}
```

Output Found: true

14. **Replace "Java" with "Python"**

```
import java.io.*;
public class FH14{
public static void main(String[] args)throws Exception{
BufferedReader br=new BufferedReader(new FileReader("story.txt"));
StringBuilder sb=new StringBuilder(); String line;
while((line=br.readLine())!=null){
sb.append(line.replace("Java","Go")).append("\n");
}
br.close();
FileWriter fw=new FileWriter("updated_story.txt");
fw.write(sb.toString());
fw.close();
}
}
```

Output

all "Java" replaced with "Go".