

1. Write a Program for Sort Students by rollNo (Comparable)

```
import java.util.*;

class Student implements Comparable<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<>();

        list.add(new Student(3,"A",80));

        list.add(new Student(1,"B",90));

        list.add(new Student(2,"C",85));

        Collections.sort(list);

        System.out.println(list);

    }

}
```

Output

[1 B 90, 2 C 85, 3 A 80]

2. Write a Program for Product by price (Comparable)

```
import java.util.*;

class Product implements Comparable<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<>();

        list.add(new Product("P1",200));

        list.add(new Product("P2",100));

        list.add(new Product("P3",150));

        Collections.sort(list);

        System.out.println(list);

    }

}

```

Output

[P2 100.0, P3 150.0, P1 200.0]

3. Write a Program for Employee by name (Comparable)

```

import java.util.*;

class Employee implements Comparable<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<>();

        list.add(new Employee("Zara"));

        list.add(new Employee("John"));

        list.add(new Employee("Adam"));

        Collections.sort(list);
    }
}

```

```
        System.out.println(list);
    }
}
```

Output

[Adam, John, Zara]

4. Write a Program for Books by bookId descending (Comparable)

```
import java.util.*;

class Book implements Comparable<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<>();
        list.add(new Book(2,"B"));
        list.add(new Book(3,"C"));
        list.add(new Book(1,"A"));
        Collections.sort(list);
        System.out.println(list);
    }
}
```

Output

[3 C, 2 B, 1 A]

5. Write a Program for Sort custom objects & show before/after

```

import java.util.*;

class Item implements Comparable<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<>();
        list.add(new Item(3,"Pen"));
        list.add(new Item(1,"Book"));
        list.add(new Item(2,"Pencil"));

        System.out.println(list);
        Collections.sort(list);
        System.out.println(list);
    }
}

```

Output

[3 Pen, 1 Book, 2 Pencil]

[1 Book, 2 Pencil, 3 Pen]

6. Write a Program for Students by marks descending (Comparator)

```

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<>();

        list.add(new Student2("A",80));

        list.add(new Student2("B",90));

        list.add(new Student2("C",85));

        list.sort((a,b)->b.marks-a.marks);

        System.out.println(list);

    }

}

```

Output

[B 90, C 85, A 80]

7. Write a Program for 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<>();

        list.add(new Product2("B",150));

        list.add(new Product2("A",200));

        list.add(new Product2("C",100));

        list.sort(Comparator.comparingDouble(p->p.price));

        System.out.println(list);

        list.sort((a,b)->Double.compare(b.price,a.price));

        System.out.println(list);

        list.sort(Comparator.comparing(p->p.name));
    }
}

```

```

        System.out.println(list);
    }
}

```

Output

```

[C 100.0, B 150.0, A 200.0]
[A 200.0, B 150.0, C 100.0]
[A 200.0, B 150.0, C 100.0]

```

8. Write a Program for 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<>();
        list.add(new Emp("A",LocalDate.of(2022,5,1)));
        list.add(new Emp("B",LocalDate.of(2020,3,10)));
        list.add(new Emp("C",LocalDate.of(2021,7,15)));
        list.sort(Comparator.comparing(e->e.date));
        System.out.println(list);
    }
}

```

Output

```

[B 2020-03-10, C 2021-07-15, A 2022-05-01]

```

9. Write a Program for 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<>();
        list.add(new City("X",5000));
        list.add(new City("Y",2000));
        list.add(new City("Z",8000));
        list.sort((a,b)->b.pop-a.pop);
        System.out.println(list);
    }
}
```

Output

[Z 8000, X 5000, Y 2000]

10. Write a Program for Strings by length

```
import java.util.*;

public class Main10{
    public static void main(String[] args){
        List<String> list=Arrays.asList("aaa","b","cccc");
        list.sort(new Comparator<String>(){
            public int compare(String a,String b){return a.length()-b.length();}
        });
        System.out.println(list);
    }
}
```

```
}  
}
```

Output

[b, aaa, cccc]

11. Write a Program for Student Comparable by name, Comparator by marks

```
import java.util.*;  
  
class Stu implements Comparable<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<>();  
        list.add(new Stu("B",85));  
        list.add(new Stu("A",90));  
        list.add(new Stu("C",80));  
        Collections.sort(list);  
        System.out.println(list);  
        list.sort((a,b)->b.marks-a.marks);  
        System.out.println(list);  
    }  
}
```

Output

[A 90, B 85, C 80]

[A 90, B 85, C 80]

12. Write a Program for Book Comparable (ID), Comparator (title then author)


```

import java.util.*;

class Book2 implements Comparable<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<>();
        list.add(new Book2(2,"Java","X"));
        list.add(new Book2(1,"Python","Y"));
        list.add(new Book2(3,"C","Z"));

        Collections.sort(list);

        System.out.println(list);

        list.sort(Comparator.comparing((Book2 b)->b.title).thenComparing(b->b.author));

        System.out.println(list);
    }
}

```

Output

[1 Python Y, 2 Java X, 3 C Z]

[3 C Z, 2 Java X, 1 Python Y]

13. Write a Program for 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<>();
        list.add(new Emp2("A",30000,"HR"));
        list.add(new Emp2("B",40000,"IT"));
        list.add(new Emp2("C",35000,"Admin"));
        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));
        System.out.println(list);
    }
}

```

Output

[A 30000.0 HR, C 35000.0 Admin, B 40000.0 IT]

14. Write a Program for Comparator.comparing() method references

```

import java.util.*;

class Person {
    String name; int age;
    Person(String n,int a){name=n;age=a;}
    public String toString(){return name+" "+age;}
}

public class Main14 {
    public static void main(String[] args) {
        List<Person> list=new ArrayList<>();
        list.add(new Person("A",25));
        list.add(new Person("B",20));
        list.sort(Comparator.comparing(Person::getName));
    }
}

```

```
}
```

15. Write a Program for 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<>((a,b)->a.age-b.age);

        set.add(new Person2("A",25));
        set.add(new Person2("B",20));
        set.add(new Person2("C",30));

        System.out.println(set);
    }
}
```

Output

[B 20, A 25, C 30]

File Handling & Serialization

1. Write a Program for 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("A\nB\nC\nD\nE\n");

        fw.close();
    }
}
```

Output (content of student.txt)

A
B
C
D
E

2. Write a Program for 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(line);

        br.close();

    }

}
```

Output

A
B
C
D
E

3 . Write a Program for 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("F\n");

        fw.close();

    }

}
```

}

Output

A

B

C

D

E

F

4. Write a Program for 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("Lines: "+lines);

        System.out.println("Words: "+words);

    }

}
```

Output

Lines: 3

Words: 15

5. Write a Program for 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. Write a Program for 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(f.getAbsolutePath());
            System.out.println(f.getName());
            System.out.println(f.canWrite());
            System.out.println(f.canRead());
            System.out.println(f.length());
        }else System.out.println("File not found");
    }
}

```

Output

```

/full/path/report.txt
report.txt
true
true
120

```

7. Write a Program for 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());

        fw.close();

    }

}

```

Output

Hello File

8. Write a Program for 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<>();

        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. Write a Program for 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(1,"A",90));

        oos.close();

    }

}

```

Output

student.ser created with object data.

10. Write a Program for 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(s.id+" "+s.name+" "+s.marks);

    }

}

```

Output

1 A 90

11. Write a Program for 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(f.getName());

    }

}

```



```
}  
}
```

Output

```
file1.txt  
file2.java  
notes.txt
```

12. Write a Program for 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(f.delete());  
        else System.out.println("File not found");  
    }  
}
```

Output

```
true
```

13. Write a Program for 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);  
    }  
}
```

```
}  
}
```

Output

true

14. Write a Program for 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","Python")).append("\n");  
        }  
        br.close();  
        FileWriter fw=new FileWriter("updated_story.txt");  
        fw.write(sb.toString());  
        fw.close();  
    }  
}
```

Output

all "Java" replaced with "Python".