1. Write a Program for Create a new text file named test.txt

```
import java.io.*;
public class CreateFile {
    public static void main(String[] args) throws IOException {
        File f = new File("test.txt");
        if (f.createNewFile()) System.out.println("File created");
        else System.out.println("File already exists");
    }
}
```

Output

```
File created
```

---

2. Write a Program for Check whether a file exists at a given path

```
import java.io.*;
public class CheckFileExists {
    public static void main(String[] args) {
        File f = new File("test.txt");
        if (f.exists()) System.out.println("File exists");
        else System.out.println("File does not exist");
    }
}
```

Output

```
File exists
```

---

3. Write a Program for "Hello, World!" into a file using FileWriter

```
import java.io.*;
public class WriteFile {
    public static void main(String[] args) throws IOException {
        FileWriter w = new FileWriter("hello.txt");
        w.write("Hello, World!");
        w.close();
        System.out.println("Written");
    }
}
```

Output

```
Written
```

---

4. Write a Program for Read a file line by line using BufferedReader

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

Output

```
Hello, World!
```

---

5. Write a Program for Append a line of text to an existing file

```
import java.io.*;
public class AppendFile {
    public static void main(String[] args) throws IOException {
        FileWriter w = new FileWriter("hello.txt", true);
        w.write("\nThis is appended text");
        w.close();
        System.out.println("Appended");
    }
}
```

Output

```
Appended
```

---

6. Write a Program for Count the number of lines, words, and characters in a file

```
import java.io.*;
public class CountFile {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader("hello.txt"));
        String line;
        int lines = 0, words = 0, chars = 0;
        while ((line = br.readLine()) != null) {
            lines++;
            words += line.split("\\s+").length;
            chars += line.length();
        }
        br.close();
        System.out.println("Lines: " + lines);
        System.out.println("Words: " + words);
        System.out.println("Characters: " + chars);
    }
}
```

Output

```
Lines: 2
Words: 5
Characters: 35
```

---

7. Write a Program for Copy content from one file to another

```
import java.io.*;
public class CopyFile {
    public static void main(String[] args) throws IOException {
        FileReader fr = new FileReader("hello.txt");
        FileWriter fw = new FileWriter("copy.txt");
        int c;
        while ((c = fr.read()) != -1) fw.write(c);
        fr.close();
        fw.close();
        System.out.println("Copied");
    }
}
```

Output

```
Copied
```

---

8. Write a Program for List all files in a directory

```
import java.io.*;
public class ListFiles {
    public static void main(String[] args) {
        File dir = new File(".");
        for (String name : dir.list()) System.out.println(name);
    }
}
```

Output (example)

```
test.txt
hello.txt
copy.txt
```

---

9. Write a Program for Filter and display only .txt files

```
import java.io.*;
public class FilterTxt {
    public static void main(String[] args) {
        File dir = new File(".");
        String[] files = dir.list((d, name) -> name.endsWith(".txt"));
        for (String name : files) System.out.println(name);
    }
}
```

Output

```
test.txt

hello.txt

copy.txt
```

---

10. Write a Program for Serialize and deserialize a Student object

```
import java.io.*;

class Student implements Serializable {

   String name; int age;

   Student(String n, int a) { name=n; age=a; }

}

public class SerDesStudent {

   public static void main(String[] args) throws Exception {

      Student s = new Student("John", 20);

      ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("student.ser"));

      oos.writeObject(s);

      oos.close();

      ObjectInputStream ois = new ObjectInputStream(new FileInputStream("student.ser"));

      Student s2 = (Student) ois.readObject();

      ois.close();

      System.out.println(s2.name + " " + s2.age);

   }

}
```

Output

```
John 20
```

---

11. Write a Program for Read a file using Scanner and display tokens

```
import java.io.*;

import java.util.*;

public class ScannerRead {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(new File("hello.txt"));
        while (sc.hasNext()) System.out.println(sc.next());
        sc.close();
    }
}
```

Output

```
Hello,
World!
This
is
appended
text
```

---

12. Write a Program for Search for a specific word and count occurrences

```
import java.io.*;
public class SearchWord {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader("hello.txt"));
        String line; int count = 0;
        while ((line = br.readLine()) != null)
            for (String w : line.split("\\s+"))
                if (w.equalsIgnoreCase("Hello,")) count++;
        br.close();
        System.out.println("Occurrences: " + count);
    }
}
```

Output

```
Occurrences: 1
```

---

13. Write a Program for Create, move, and delete a file

```
import java.nio.file.*;

public class FileOps {

    public static void main(String[] args) throws Exception {

        Path p = Paths.get("sample.txt");

        Files.createFile(p);

        Files.move(p, Paths.get("moved.txt"), StandardCopyOption.REPLACE_EXISTING);

        Files.delete(Paths.get("moved.txt"));

        System.out.println("Done");

    }

}
```

Output

```
Done
```

---

14. Write a Program for all lines using Files.readAllLines()

```
import java.nio.file.*;

import java.util.*;

public class ReadAll {

    public static void main(String[] args) throws Exception {

        List<String> lines = Files.readAllLines(Paths.get("hello.txt"));

        for (String l : lines) System.out.println(l);

    }

}
```

Output

```
Hello, World!

This is appended text
```

---

15. Write a Program for write and append using Files.write()

```
import java.nio.file.*;

import java.nio.file.StandardOpenOption;

import java.util.*;

public class FilesWrite {
    public static void main(String[] args) throws Exception {
        Files.write(Paths.get("data.txt"), Arrays.asList("First line"));
        Files.write(Paths.get("data.txt"), Arrays.asList("Second line"), StandardOpenOption.APPEND);
        System.out.println("Written");
    }
}
```

Output

```
Written
```

---

16. Write a Program for Walk through a directory tree

```
import java.nio.file.*;
public class WalkDir {
    public static void main(String[] args) throws Exception {
        Files.walk(Paths.get(".")).forEach(System.out::println);
    }
}
```

Output (example)

```
.
./test.txt
./hello.txt
./copy.txt
```

---

17. Write a Program for Copy a file with REPLACE_EXISTING

```
import java.nio.file.*;
public class FilesCopy {
    public static void main(String[] args) throws Exception {
        Files.copy(Paths.get("hello.txt"), Paths.get("copy2.txt"),
StandardCopyOption.REPLACE_EXISTING);
        System.out.println("Copied");
    }
}
```

Output

```
Copied
```

---

18. Write a Program for Print size of a file

```
import java.nio.file.*;
public class FileSize {
    public static void main(String[] args) throws Exception {
        long size = Files.size(Paths.get("hello.txt"));
        System.out.println("Size: " + size + " bytes");
    }
}
```

Output

```
Size: 35 bytes
```

---

19. Write a Program for Serialize Employee object

```
import java.io.*;
class Employee implements Serializable {
    String name; int id;
    Employee(String n, int i) { name=n; id=i; }
}
public class SerEmployee {
    public static void main(String[] args) throws Exception {
        Employee e = new Employee("Alice", 101);
        ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("employee.ser"));
        oos.writeObject(e);
        oos.close();
        System.out.println("Serialized");
    }
}
```

Output

```
Serialized
```

20. Write a Program for Deserialize Employee object

```
import java.io.*;

class Employee implements Serializable {

   String name; int id;

   Employee(String n, int i) { name=n; id=i; }

}

public class DesEmployee {

   public static void main(String[] args) throws Exception {

      ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"));

      Employee e = (Employee) ois.readObject();

      ois.close();

      System.out.println(e.name + " " + e.id);

   }

}
```

Output

```
Alice 101
```