Java I/O File Handling -

- 1. Write a program to create a new text file named test.txt.

ANSWER

```
import java.io.File;
import java.io.IOException;

public class CreateFileExample {
  public static void main(String[] args) {
    try {
      File file = new File("test.txt");
      if (file.createNewFile()) {
        System.out.println("File created: " + file.getName());
      } else {
        System.out.println("File already exists!");
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

- 2. Write a program to check whether a file exists at a given path.

ANSWER

```
import java.io.File;

public class FileExistCheck {
  public static void main(String[] args) {
    File file = new File("test.txt");
    if (file.exists()) {
      System.out.println("File exists at: " + file.getAbsolutePath());
```

```
        } else {
            System.out.println("File does not exist!");
        }
    }
}
```

- 3. Write a Java program to write "Hello, World!" into a file using FileWriter.

ANSWER

```
import java.io.FileWriter;
import java.io.IOException;

public class WriteFile {
    public static void main(String[] args) {
        try (FileWriter fw = new FileWriter("hello.txt")) {
            fw.write("Hello, World!");
            System.out.println("Data written successfully!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- 4. Write a program to read the content of a file line by line using BufferedReader.

ANSWER

```
import java.io.*;

public class ReadFileBuffered {
    public static void main(String[] args) {
        try (BufferedReader br = new BufferedReader(new FileReader("hello.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- 5. Write a program to append a line of text to an existing file.

ANSWER

```
import java.io.FileWriter;
import java.io.IOException;

public class AppendFile {
    public static void main(String[] args) {
        try (FileWriter fw = new FileWriter("hello.txt", true)) {
            fw.write("\nThis is appended text.");
            System.out.println("Data appended successfully!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- 6. Write a program to count the number of lines, words, and characters in a file.

ANSWER

```
import java.io.*;
```

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

- 7. Write a program to copy content from one file to another using FileReader and FileWriter.

ANSWER

```
import java.io.*;

public class CopyFile {
  public static void main(String[] args) {
    try (FileReader fr = new FileReader("hello.txt");
       FileWriter fw = new FileWriter("copy.txt")) {
            int ch;
            while ((ch = fr.read()) != -1) {
                fw.write(ch);
            }
            System.out.println("File copied successfully!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- 8. Write a program that lists all the files in a directory.

ANSWER

```
import java.io.File;

public class ListFiles {
    public static void main(String[] args) {
        File dir = new File(".");
        File[] files = dir.listFiles();
        if (files != null) {
            for (File f : files) {
                if (f.isFile())
                    System.out.println(f.getName());
            }
        }
    }
}
```

- 9. Write a program to filter and display only .txt files from a folder using FilenameFilter.

ANSWER

```
import java.io.*;

public class TxtFileFilter {
    public static void main(String[] args) {
        File dir = new File(".");
        FilenameFilter filter = (d, name) -> name.endsWith(".txt");
        File[] files = dir.listFiles(filter);
        if (files != null) {
            for (File f : files) {
                System.out.println(f.getName());
            }
        }
    }
}
```

- 10. Write a program to serialize and deserialize a Student object to and from a file.

ANSWER

```
import java.io.*;

class Student implements Serializable {
    int id;
    String name;
    Student(int id, String name) {
        this.id = id;
        this.name = name;
    }
}
```

```
public class SerializeStudent {
    public static void main(String[] args) {
        // Serialization
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("student.ser"))) {
            Student s = new Student(1, "Samarth");
            oos.writeObject(s);
            System.out.println("Student serialized!");
        } catch (IOException e) { e.printStackTrace(); }

        // Deserialization
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("student.ser"))) {
            Student s = (Student) ois.readObject();
            System.out.println("Deserialized Student: " + s.id + " - " + s.name);
        } catch (Exception e) { e.printStackTrace(); }
    }
}
```

- 11. Write a program to read a file using Scanner and display the tokens.

ANSWER

```
import java.io.File;
import java.util.Scanner;

public class ReadWithScanner {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(new File("hello.txt"))) {
            while (sc.hasNext()) {
                System.out.println(sc.next());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- 12. Write a program to search for a specific word in a file and count its occurrences.

ANSWER

```
import java.io.*;
import java.util.*;

public class WordSearch {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter word to search: ");
        String word = sc.next();
        int count = 0;

        try (BufferedReader br = new BufferedReader(new FileReader("hello.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] words = line.split("\\s+");
                for (String w : words) {
                    if (w.equalsIgnoreCase(word)) count++;
                }
            }
        }
        System.out.println("Word '" + word + "' found " + count + " times.");
    }
}
```

- 13. Write a program to create, move, and delete a file using Files and Paths.

ANSWER

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

public class FileOperations {
    public static void main(String[] args) throws Exception {
        Path file = Paths.get("sample.txt");

        // Create
        Files.createFile(file);
        System.out.println("File created!");

        // Move
        Path target = Paths.get("moved_sample.txt");
        Files.move(file, target, StandardCopyOption.REPLACE_EXISTING);
        System.out.println("File moved!");

        // Delete
        Files.delete(target);
        System.out.println("File deleted!");
    }
}
```

- 14. Write a program to read all lines of a file using Files.readAllLines() and print them.

ANSWER

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

import java.util.*;


public class ReadAllLines {

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

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

    lines.forEach(System.out::println);

  }

}
```

- 15. Write a program to write data into a file using Files.write() and append using StandardOpenOption.APPEND.

ANSWER

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

import java.nio.charset.StandardCharsets;

import java.nio.file.StandardOpenOption;

import java.util.*;


public class WriteAndAppend {

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

    Path path = Paths.get("file.txt");


    Files.write(path, "First line\n".getBytes(StandardCharsets.UTF_8));


    Files.write(path, Arrays.asList("Appended line"), StandardCharsets.UTF_8,
StandardOpenOption.APPEND);


    System.out.println("Data written and appended!");

  }

}
```

- 16. Write a program to walk through a directory tree and display file names using Files.walk().

ANSWER

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


public class WalkDirectory {

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

    Files.walk(Paths.get("."))

        .forEach(System.out::println);

  }

}
```

- 17. Write a program to copy a file using Files.copy() with REPLACE_EXISTING option.

ANSWER

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

public class CopyFileNIO {
    public static void main(String[] args) throws Exception {
        Path source = Paths.get("hello.txt");
        Path dest = Paths.get("backup.txt");
        Files.copy(source, dest, StandardCopyOption.REPLACE_EXISTING);
        System.out.println("File copied with REPLACE_EXISTING!");
    }
}
```

- 18. Write a program to check and print the size of a file in bytes using Files.size().

ANSWER

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

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

- 19. Write a program to serialize a class Employee and store it in employee.ser.

ANSWER

```
import java.io.*;
```

```
class Employee implements Serializable {
    int id;
    String name;
    double salary;

    Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }
}

public class EmployeeSerialization {
    public static void main(String[] args) {
        // Serialize
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("employee.ser"))) {
            Employee e = new Employee(101, "Samarth", 50000);
            oos.writeObject(e);
            System.out.println("Employee serialized!");
        } catch (IOException ex) { ex.printStackTrace(); }

        // Deserialize
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"))) {
            Employee e = (Employee) ois.readObject();
            System.out.println("Deserialized Employee: " + e.id + " " + e.name + " " + e.salary);
        } catch (Exception ex) { ex.printStackTrace(); }
    }
}
```

- 20. Write a program to deserialize the employee.ser file and display the object data.

ANSWER

```
import java.io.*;

class Employee implements Serializable {
  int id;
  String name;
  double salary;

  Employee(int id, String name, double salary) {
    this.id = id;
    this.name = name;
    this.salary = salary;
  }
}

public class EmployeeSerialization {
  public static void main(String[] args) {
    // Serialize
    try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream("employee.ser"))) {
      Employee e = new Employee(101, "Samarth", 50000);
      oos.writeObject(e);
      System.out.println("Employee serialized!");
    } catch (IOException ex) { ex.printStackTrace(); }

    // Deserialize
```

```
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"))) {

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

            System.out.println("Deserialized Employee: " + e.id + " " + e.name + " " + e.salary);

        } catch (Exception ex) { ex.printStackTrace(); }

    }

}
```