Java I/O File Handling

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

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

public class CreateFile {
   public static void main(String[] args) {
      File file = new File("test.txt");

      try {
         if (file.createNewFile()) {
            System.out.println("File created: " + file.getName());
         } else {
            System.out.println("File already exists: " + file.getName());
         }
      } catch (IOException e) {
         System.out.println("An error occurred while creating the file.");
         e.printStackTrace();
      }
   }
}
```

Output:
File created: test.txt

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

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

public class CheckFileExists {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the file path to check: ");
      String filePath = sc.nextLine();

      File file = new File(filePath);

      if (file.exists() && file.isFile()) {
```

```
            System.out.println("File exists at: " + file.getAbsolutePath());
        } else {
            System.out.println("File does not exist at the specified path.");
        }

        sc.close();
    }
}
```

Output:
Enter the file path to check: C:\Users \Documents\report.txt
File exists at: C:\Users \Documents\report.txt

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

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

public class WriteHelloWorld {
    public static void main(String[] args) {
        String fileName = "hello.txt";

        try (FileWriter writer = new FileWriter(fileName)) {
            writer.write("Hello, World!");
            System.out.println("Successfully wrote to " + fileName);
        } catch (IOException e) {
            System.out.println("An error occurred while writing to the file.");
            e.printStackTrace();
        }
    }
}
```

Output:
Successfully wrote to hello.txt

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

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
```

```
public class ReadFileLineByLine {
   public static void main(String[] args) {
      String fileName = "input.txt";  // Change this to your file name

      try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
         String line;
         System.out.println("Contents of " + fileName + ":");
         while ((line = br.readLine()) != null) {
            System.out.println(line);
         }
      } catch (IOException e) {
         System.out.println("An error occurred while reading the file.");
         e.printStackTrace();
      }
   }
}
```

Output:
Hello, World!
Welcome to file reading in Java.
This is the third line.

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

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

public class AppendToFile {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the filename to append to: ");
      String fileName = sc.nextLine();

      System.out.print("Enter the line of text to append: ");
      String textToAppend = sc.nextLine();

      try (FileWriter writer = new FileWriter(fileName, true)) { // true enables append
mode
         writer.write(textToAppend + System.lineSeparator());
         System.out.println("Successfully appended text to " + fileName);
      } catch (IOException e) {
```

```
            System.out.println("An error occurred while appending to the file.");
            e.printStackTrace();
        }

        sc.close();
    }
}
```

Output:
Enter the filename to append to: notes.txt
Enter the line of text to append: This is a new line added to the file.
Successfully appended text to notes.txt

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

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class FileStatistics {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the filename to analyze: ");
        String fileName = sc.nextLine();

        int lineCount = 0;
        int wordCount = 0;
        int charCount = 0;

        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
            String line;

            while ((line = br.readLine()) != null) {
                lineCount++;


                String[] words = line.trim().split("\\s+");
                if (!line.trim().isEmpty()) {
                    wordCount += words.length;
                }
```

```
            charCount += line.length();
        }

        System.out.println("File: " + fileName);
        System.out.println("Number of lines: " + lineCount);
        System.out.println("Number of words: " + wordCount);
        System.out.println("Number of characters: " + charCount);

    } catch (IOException e) {
        System.out.println("An error occurred while reading the file.");
        e.printStackTrace();
    }

    sc.close();
  }
}
```

Output:
Enter the filename to analyze: sample.txt
File: sample.txt
Number of lines: 5
Number of words: 40
Number of characters: 230

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

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class CopyFile {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter source filename: ");
    String sourceFile = sc.nextLine();

    System.out.print("Enter destination filename: ");
    String destFile = sc.nextLine();
```

```
        try (FileReader fr = new FileReader(sourceFile);
             FileWriter fw = new FileWriter(destFile)) {

            int ch;
            while ((ch = fr.read()) != -1) {
                fw.write(ch);
            }

            System.out.println("File copied successfully from " + sourceFile + " to " +
destFile);

        } catch (IOException e) {
            System.out.println("An error occurred during file copying.");
            e.printStackTrace();
        }

        sc.close();
    }
}
```

Output:
Enter source filename: source.txt
Enter destination filename: destination.txt
File copied successfully from source.txt to destination.txt

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

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

public class ListFilesInDirectory {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter directory path: ");
        String dirPath = sc.nextLine();

        File directory = new File(dirPath);

        if (directory.exists() && directory.isDirectory()) {
            String[] contents = directory.list();

            if (contents != null && contents.length > 0) {
```

```
            System.out.println("Contents of directory '" + dirPath + "':");
            for (String item : contents) {
                System.out.println(item);
            }
        } else {
            System.out.println("The directory is empty.");
        }
    } else {
        System.out.println("The specified path is not a valid directory.");
    }

    sc.close();
  }
}
```

Output:

```
Enter directory path: C:\Users\Documents
Contents of directory 'C:\Users\Documents':
report.pdf
photos
notes.txt
projects
```

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

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

public class ListTxtFiles {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter folder path: ");
    String folderPath = sc.nextLine();

    File folder = new File(folderPath);

    if (folder.exists() && folder.isDirectory()) {

      FilenameFilter txtFilter = new FilenameFilter() {
        public boolean accept(File dir, String name) {
          return name.toLowerCase().endsWith(".txt");
```

```
            }
        };

        String[] txtFiles = folder.list(txtFilter);

        if (txtFiles != null && txtFiles.length > 0) {
            System.out.println(".txt files in folder \"" + folderPath + "\":");
            for (String fileName : txtFiles) {
                System.out.println(fileName);
            }
        } else {
            System.out.println("No .txt files found in the folder.");
        }
    } else {
        System.out.println("The specified path is not a valid directory.");
    }

    sc.close();
  }
}
```

Output:

```
Enter folder path: C:\Users\Documents
.txt files in folder "C:\Users\Documents":
notes.txt
todo.txt
report.txt
```

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

```
import java.io.*;

class Student implements Serializable {
  private static final long serialVersionUID = 1L;

  private String name;
  private int age;
  private String department;

  public Student(String name, int age, String department) {
    this.name = name;
    this.age = age;
```

```
        this.department = department;
    }

    @Override
    public String toString() {
        return "Student{name='" + name + "', age=" + age + ", department='" + department +
"'}";
    }
}

public class SerializeDeserializeStudent {
    public static void main(String[] args) {
        String filename = "student.ser";



        Student student = new Student("Alice", 20, "Computer Science");

        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(filename))) {
            oos.writeObject(student);
            System.out.println("Student object serialized to " + filename);
        } catch (IOException e) {
            System.out.println("Error serializing object");
            e.printStackTrace();
        }

        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(filename))) {
            Student deserializedStudent = (Student) ois.readObject();
            System.out.println("Deserialized Student object:");
            System.out.println(deserializedStudent);
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Error deserializing object");
            e.printStackTrace();
        }
    }
}
```

Output:
Student object serialized to student.ser
Deserialized Student object:
Student{name='Alice', age=20, department='Computer Science'}

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

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

public class ReadFileWithScanner {
   public static void main(String[] args) {
      Scanner inputScanner = new Scanner(System.in);
      System.out.print("Enter the filename to read: ");
      String fileName = inputScanner.nextLine();

      File file = new File(fileName);

      try (Scanner fileScanner = new Scanner(file)) {
         System.out.println("Tokens in the file:");
         while (fileScanner.hasNext()) {
            String token = fileScanner.next();
            System.out.println(token);
         }
      } catch (FileNotFoundException e) {
         System.out.println("File not found: " + fileName);
      }

      inputScanner.close();
   }
}
```

Output:
If example.txt contains:
Hello, world!
This is a test.

Output will be:
Tokens in the file:
Hello,
world!
This
is
a
test.

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

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class WordCountInFile {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);

System.out.print("Enter filename: ");
String fileName = sc.nextLine();

System.out.print("Enter word to search: ");
String wordToFind = sc.nextLine();

int count = 0;

try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
String line;

while ((line = br.readLine()) != null) {
        String[] words = line.split("[\\s\\p{Punct}]+");

        for (String word : words) {
        if (word.equalsIgnoreCase(wordToFind)) {
                count++;
        }
        }
}

System.out.println("The word \"" + wordToFind + "\" occurs " + count + " times in the
file.");

} catch (IOException e) {
System.out.println("An error occurred while reading the file.");
e.printStackTrace();
}

sc.close();
```

```
    }
}
```

Output:
Enter filename: notes.txt
Enter word to search: java
The word "java" occurs 5 times in the file.

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

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

public class FileOperations {
   public static void main(String[] args) {
      Path sourcePath = Paths.get("myfile.txt");
      Path targetPath = Paths.get("subfolder", "myfile_moved.txt");

      try {
         if (!Files.exists(sourcePath)) {
            Files.createFile(sourcePath);
            System.out.println("File created: " + sourcePath.toAbsolutePath());
         } else {
            System.out.println("File already exists: " + sourcePath.toAbsolutePath());
         }

if (!Files.exists(targetPath.getParent())) {
            Files.createDirectories(targetPath.getParent());
         }

         Files.move(sourcePath, targetPath, StandardCopyOption.REPLACE_EXISTING);
         System.out.println("File moved to: " + targetPath.toAbsolutePath());

         Files.delete(targetPath);
         System.out.println("File deleted: " + targetPath.toAbsolutePath());

      } catch (IOException e) {
         System.out.println("An error occurred during file operations.");
         e.printStackTrace();
      }
   }
}
```

Output:
File created: /path/to/your/project/myfile.txt
File moved to: /path/to/your/project/subfolder/myfile_moved.txt
File deleted: /path/to/your/project/subfolder/myfile_moved.txt

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

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;
import java.util.List;
import java.util.Scanner;

public class ReadAllLinesExample {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the filename to read: ");
      String fileName = sc.nextLine();

      Path filePath = Paths.get(fileName);

      try {
         List<String> allLines = Files.readAllLines(filePath);
         System.out.println("Contents of " + fileName + ":");
         for (String line : allLines) {
            System.out.println(line);
         }
      } catch (IOException e) {
         System.out.println("Error reading the file.");
         e.printStackTrace();
      }

      sc.close();
   }
}
```

Output:
Enter the filename to read: example.txt
Contents of example.txt:

Hello, World!
This is a sample file.
Have a nice day!

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

```
import java.nio.file.*;
import java.io.IOException;
import java.util.Scanner;
import java.util.List;
import java.util.Arrays;
import java.nio.charset.StandardCharsets;

public class WriteAndAppendFile {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter filename: ");
      String fileName = sc.nextLine();

      System.out.print("Enter text to write/append: ");
      String text = sc.nextLine();

      Path filePath = Paths.get(fileName);

      try {
         if (!Files.exists(filePath)) {
            Files.write(filePath, Arrays.asList(text), StandardCharsets.UTF_8);
            System.out.println("Data written to new file: " + fileName);
         } else {
            Files.write(filePath, Arrays.asList(text), StandardCharsets.UTF_8,
StandardOpenOption.APPEND);
            System.out.println("Data appended to file: " + fileName);
         }
      } catch (IOException e) {
         System.out.println("Error writing/appending to file.");
         e.printStackTrace();
      }

      sc.close();
   }
```

```
}
```

Output:
Enter filename: notes.txt
Enter text to write/append: This is a new note.
Data written to new file: notes.txt

Enter filename: notes.txt
Enter text to write/append: Another note.
Data appended to file: notes.txt

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

```
import java.nio.file.*;
import java.io.IOException;
import java.util.Scanner;
import java.util.stream.Stream;

public class WalkDirectoryTree {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter directory path to walk: ");
      String dirPath = sc.nextLine();

      Path startPath = Paths.get(dirPath);

      if (!Files.exists(startPath) || !Files.isDirectory(startPath)) {
         System.out.println("Invalid directory path.");
         sc.close();
         return;
      }

      try (Stream<Path> stream = Files.walk(startPath)) {
         System.out.println("Files in directory tree:");
         stream
            .filter(Files::isRegularFile)  // Only files, skip directories
            .forEach(path -> System.out.println(path.toString()));
      } catch (IOException e) {
         System.out.println("Error walking the directory tree.");
         e.printStackTrace();
```

```
        }

        sc.close();
    }
}
```

Output:
Enter directory path to walk: C:\Users\Documents
Files in directory tree:
C:\Users\Documents\report.pdf
C:\Users\Documents\notes.txt
C:\Users\Documents\projects\project1.docx
C:\Users\Documents\projects\project2.xlsx

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

```
import java.nio.file.*;
import java.io.IOException;
import java.util.Scanner;

public class CopyFileWithReplace {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter source file path: ");
        String sourcePathStr = sc.nextLine();

        System.out.print("Enter destination file path: ");
        String destPathStr = sc.nextLine();

        Path sourcePath = Paths.get(sourcePathStr);
        Path destPath = Paths.get(destPathStr);

        try {
            Files.copy(sourcePath, destPath, StandardCopyOption.REPLACE_EXISTING);
            System.out.println("File copied successfully with replace option.");
        } catch (IOException e) {
            System.out.println("An error occurred during file copy.");
            e.printStackTrace();
        }
```

```
        sc.close();
    }
}
```

Output:
Enter source file path: C:\Users\Documents\file1.txt
Enter destination file path: C:\Users\Documents\backup\file1.txt
File copied successfully with replace option.

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

```
import java.nio.file.*;
import java.io.IOException;
import java.util.Scanner;

public class FileSizeChecker {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the file path: ");
        String filePathStr = sc.nextLine();

        Path filePath = Paths.get(filePathStr);

        try {
            if (Files.exists(filePath) && Files.isRegularFile(filePath)) {
                long size = Files.size(filePath);
                System.out.println("Size of file '" + filePath.getFileName() + "' is: " + size + "
bytes.");
            } else {
                System.out.println("File does not exist or is not a regular file.");
            }
        } catch (IOException e) {
            System.out.println("An error occurred while checking the file size.");
            e.printStackTrace();
        }

        sc.close();
    }
}
```

Output:

Enter the file path: example.txt
Size of file 'example.txt' is: 2048 bytes.

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

```
import java.io.*;

class Employee implements Serializable {
   private static final long serialVersionUID = 1L;

   private String name;
   private int id;
   private double salary;

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

   public String toString() {
      return "Employee{name='" + name + "', id=" + id + ", salary=" + salary + "}";
   }
}

public class SerializeEmployee {
   public static void main(String[] args) {
      Employee emp = new Employee("John Doe", 101, 75000);

      try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream("employee.ser"))) {
         oos.writeObject(emp);
         System.out.println("Employee object serialized to employee.ser");
      } catch (IOException e) {
         System.out.println("Error serializing Employee object");
         e.printStackTrace();
      }
   }
}
```

Output:

Employee object serialized to employee.ser

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

```
import java.io.*;

class Employee implements Serializable {
   private static final long serialVersionUID = 1L;

   private String name;
   private int id;
   private double salary;


   public Employee() {}

   public String toString() {
      return "Employee{name='" + name + "', id=" + id + ", salary=" + salary + "}";
   }
}

public class DeserializeEmployee {
   public static void main(String[] args) {
      try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream("employee.ser"))) {
         Employee emp = (Employee) ois.readObject();
         System.out.println("Deserialized Employee object:");
         System.out.println(emp);
      } catch (IOException | ClassNotFoundException e) {
         System.out.println("Error deserializing Employee object");
         e.printStackTrace();
      }
   }
}
```

Output:
Deserialized Employee object:
Employee{name='John Doe', id=101, salary=75000.0}