Ass6_8

Q1

```
public class Q1 {
    private String name;

    public Q1(String name) {
        this.name = name;
    }

    public void show() {
        Q1 b = new Q1("Inner Object");
        b.display();
    }

    public void display() {
        Q1 a=new Q1("Inner Display Object");
    }

    @Override
    protected void finalize() throws Throwable {
        System.out.println("Garbage collected: " + name);
        super.finalize();
    }

    public static void main(String[] args) {
        Q1 obj = new Q1("Main Object");
        obj.show();
        System.gc();
    }
}
```

Q2

```
package ass6;
```

```
public class Q2 {
    private String name;

    public Q2(String name) {
        this.name = name;
    }

    @Override
    protected void finalize() throws Throwable {
        System.out.println("Garbage collected: " + name);
        super.finalize();
    }
    public static void main(String[] args) {
        Q2 obj1 = new Q2("Object 1");
        Q2 obj2 = new Q2("Object 2");
        obj1 = obj2;
        obj2 = null;
        System.gc();
    }
}
```

Q3

```
package ass6;
public class Q3 {
    private String name;

    public Q3(String name) {
        this.name = name;
    }

    @Override
    protected void finalize() throws Throwable {
        System.out.println("Garbage collected: " + name);
        super.finalize();
    }
```

```
    public static void main(String[] args) {
        Q3 obj = new Q3("Object");
        obj = null;
        System.gc();
    }
}
```

Q4

```
package ass6;

public class Q4 {
    private String name;

    public Q4(String name) {
        this.name = name;
    }

    @Override
    protected void finalize() throws Throwable {
        System.out.println("Garbage collected: " + name);
        super.finalize();
    }

    public static void main(String[] args) {
        new Q4("Anonymous");
        System.gc();
    }
}
```

Q5

```
package ass6;
import java.util.Random;

class DataContainer {
```

```
    private int intValue;
    private double doubleValue;

    DataContainer() {
        this.intValue = 0;
        this.doubleValue = 0.0;
    }

    void setData(int i, double d) {
        this.intValue = i;
        this.doubleValue = d;
    }

    void updateIntValue(int i) {
        this.intValue = i;
    }

    void updateDoubleValue(double d) {
        this.doubleValue = d;
    }

    void printData() {
        System.out.println("Integer Value: " + intValue);
        System.out.println("Double Value: " + doubleValue);
    }
}

public class MemoryAllocation {

    public static void main(String[] args) {
        DataContainer obj1 = new DataContainer();
        obj1.setData(new Random().nextInt(100), new Random().
        obj1.printData();

        DataContainer obj2 = new DataContainer();
        obj2.setData(new Random().nextInt(100), new Random().
        obj2.printData();
```

```
        obj1 = null;
        obj2 = null;

        Runtime runtime = Runtime.getRuntime();
 System.out.println("Total Memory before garbage collection:
System.out.println("Free Memory before garbage collection: " +

        runtime.gc();
        System.out.println("Total Memory: " + runtime.totalMem
        System.out.println("Free Memory: " + runtime.freeMemo
  System.out.println("Used Memory: " + (runtime.totalMemory()
    }
}
```

Q6

```
package ass6;

import java.util.ArrayList;

public class Q6 {
    public static void main(String[] args) {
        long startTime = System.currentTimeMillis();

        ArrayList<Object> objects = new ArrayList<>();

        try {
            while (true) {
                objects.add(new Object());
                if (System.currentTimeMillis() - startTime > 
                    printMemoryUsage(startTime);
                    startTime = System.currentTimeMillis();
                }
            }
        } catch (OutOfMemoryError e) {
            System.out.println("Out of memory");
            printMemoryUsage(startTime);
        }
```

```
    }

    private static void printMemoryUsage(long startTime) {
        long totalMemory = Runtime.getRuntime().totalMemory()
        long freeMemory = Runtime.getRuntime().freeMemory();
        long usedMemory = totalMemory - freeMemory;

System.out.println("Timestamp: " + (System.currentTimeMillis(
        System.out.println("Total memory: " + totalMemory + "
        System.out.println("Free memory: " + freeMemory + " by
        System.out.println("Used memory: " + usedMemory + " by
        System.out.println("------------------------");
    }
}

Terminal:
javac ass6/Q6.java
java -XX:+UseG1GC ass6/Q6
```

Q7

```
package ass6;
public class Q7 {

    static class Student {
        private String name;
        private int id;

        public Student(String name, int id) {
            this.name = name;
            this.id = id;
        }

        @Override
        protected void finalize() throws Throwable {
System.out.println("Student " + name + " with ID " + id + " i
            super.finalize();
        }
```

```
    }

    public static void main(String[] args) {
        Runtime runtime = Runtime.getRuntime();

        long startTime = System.currentTimeMillis();

        for (int i = 0; i < 100000; i++) {
            Student student = new Student("Student" + i, i);
        }

        long memoryUsage = runtime.totalMemory() - runtime.fr
System.out.println("Memory used before garbage collection: " +

        System.gc();

        long memoryUsageAfterGC = runtime.totalMemory() - run
System.out.println("Memory used after garbage collection: " +

        long endTime = System.currentTimeMillis();
        long timeElapsed = endTime - startTime;
        System.out.println("Time elapsed: " + timeElapsed + "
    }
}
```

Ass8

Q1

```
package ass8;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.*;

public class Q1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Enter your diary entry:");
        String entry = scanner.nextLine();

        File file = new File("diary.txt");

        try {
            if (file.exists()) {
System.out.println("The file 'diary.txt' already exists. Exis
"will not be overwritten. Do you want to append to it? (Y/N)"

                String response = scanner.nextLine().trim().t

                if (response.equals("Y") || response.equals("
                    // Open a FileWriter in append mode to wr
                    FileWriter writer = new FileWriter(file,

                    writer.write(getCurrentDate() + "\n" + en

                    writer.close();

                    System.out.println("Diary entry added suc
                }
                else{
                    System.out.println("Exiting without makin
                    return;
                }
            } else {
                if (file.createNewFile()) {
                    System.out.println("File 'diary.txt' crea
                } else {
                    System.out.println("Failed to create file
                    return;
                }
            }


        } catch (IOException e) {
   System.out.println("An error occurred while writing to the
```

```
        } finally {
            scanner.close();
        }
    }
    private static String getCurrentDate() {
        Date date = new Date();
        return new SimpleDateFormat("dd-MM-yyyy ").format(dat
//        return new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").
    }
}
```

Q2

```
package ass8;

import java.io.*;

public class Q2 {
    public static void main(String[] args) {
        File file = new File("diary1.txt");
        if (!file.exists()) {
            System.out.println("The file "+file+" does not ex
            return;
        }

        try (FileReader fileReader = new FileReader(file)) {
            int character;
            System.out.println("Diary entries:");
            while ((character = fileReader.read()) != -1) {
                System.out.print((char) character);
            }
        } catch (IOException e) {
   System.out.println("An error occurred while reading the fi
        }
    }
}
```

Q4

```
package ass8;

import java.io.File;

public class Q4 {

    public static void main(String[] args) {
        String directoryPath = "src/ass8";

        File directory = new File(directoryPath);

        if (!directory.exists()) {
            System.out.println("The directory does not exist.
            return;
        }

        File[] files = directory.listFiles();

        if (files.length == 0) {
            System.out.println("The directory is empty.");
            return;
        }

        System.out.println("Files and directories in " + dire
        for (File file : files) {
            System.out.println(file.getName());
        }
    }
}
```

Q5

```
package ass8;
import java.io.File;
import java.io.FilenameFilter;
import java.util.Scanner;
```

```
public class Q5 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the directory path:");
        String dirPath = scanner.nextLine();

        File dir = new File(dirPath);
        if (!dir.exists() || !dir.isDirectory()) {
            System.out.println("Invalid directory path.");
            return;
        }

        File[] files = dir.listFiles(new FilenameFilter() {
            @Override
            public boolean accept(File dir, String name) {
                return name.toLowerCase().endsWith(".txt");
            }
        });
        if (files == null || files.length == 0) {
            System.out.println("No text files found in the di
        } else {
            System.out.println("Text files in " + dirPath + "
            for (File file : files) {
                System.out.println(file.getName());
            }
        }
    }
}
```

Q6

```
package ass8;

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

```
public class Q6 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the name of the file to delet
        String fileName = scanner.nextLine();

        File fileToDelete = new File(fileName);

        if (fileToDelete.exists()) {
            if (fileToDelete.delete()) {
                System.out.println("File deleted successfully
            } else {
                System.out.println("Unable to delete the file
            }
        } else {
            System.out.println("File does not exist: " + file
        }

        scanner.close();
    }
}
```

Q7

```
package ass8;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class Q7 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Enter the source file path:");
        String srcFilePath = scanner.nextLine();

        System.out.println("Enter the destination file path:"
        String destFilePath = scanner.nextLine();

        File srcFile = new File(srcFilePath);
        File destFile = new File(destFilePath);

        if (!srcFile.exists() || !srcFile.isFile()) {
            System.out.println("Source file does not exist or
            return;
        }

        if (destFile.exists()) {
            System.out.println("Destination file already exis
            String overwrite = scanner.nextLine();
            if (!overwrite.equalsIgnoreCase("yes")) {
                System.out.println("File copy cancelled.");
                return;
            }
        }

        try (
            FileReader in = new FileReader(srcFile);
            FileWriter out = new FileWriter(destFile)
        ) {
            int c;
            while ((c = in.read()) != -1) {
                out.write(c);
            }
            System.out.println("File content copied successfu
        } catch (IOException e) {
    System.err.println("Error occurred while copying file con
        }
    }
}
```

Q8

```
package ass8;

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

public class Q8 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the current file name: ");
        String srcName = scanner.nextLine();

        System.out.print("Enter the new file name: ");
        String destName = scanner.nextLine();

        File srcFile = new File(srcName);
        File destFile = new File(destName);

        if (srcFile.renameTo(destFile)) {
            System.out.println("File renamed successfully.");
        } else {
            System.err.println("Error renaming file.");
        }

        scanner.close();
    }
}
```

Q9

```
package ass8;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.file.Files;
```

```
import java.nio.file.Paths;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Q9 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the file name: ");
        String fileName = scanner.nextLine();

        File file = new File(fileName);
        try {

        if (!file.exists()) {
            throw new FileNotFoundException("File not found:

        }

        System.out.println("File metadata:");
        System.out.println("Name: " + file.getName());
        System.out.println("Path: " + file.getPath());
        System.out.println("Absolute path: " + file.getAbsolu
        System.out.println("Parent: " + file.getParent());
        System.out.println("Exists: " + file.exists());
        System.out.println("Is file: " + file.isFile());
        System.out.println("Is directory: " + file.isDirector
        System.out.println("Is hidden: " + file.isHidden());
        System.out.println("Length: " + file.length() + " byt

        Date lastModifiedDate = new Date(file.lastModified())
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM
        System.out.println("Last modified: " + dateFormat.for

        System.out.println("Readable: " + Files.isReadable(Pa
        System.out.println("Writable: " + Files.isWritable(Pa
        System.out.println("Executable: " + Files.isExecutabl
```

```
        }catch (FileNotFoundException  e) {
            System.err.println(e);
        }finally {
            scanner.close();
        }
    }
}
```

Q10

```
package ass8;

import java.io.File;
import java.util.Scanner;
class DirectoryNotFoundException extends Exception {
    public DirectoryNotFoundException(String message) {
        super(message);
    }
}
public class Q10 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the directory path: ");
        String dirPath = scanner.nextLine();

        File dir = new File(dirPath);
    try {
        if (!dir.exists() || !dir.isDirectory()) {
            throw new DirectoryNotFoundException("Directory n
        }

        System.out.println("Directory listing:");
        listFilesAndDirectories(dir, "");
    }catch (DirectoryNotFoundException e) {
        System.err.println(e);
    }finally {
```

```
        scanner.close();
    }
    }

    private static void listFilesAndDirectories(File dir, Str
        File[] files = dir.listFiles();

        for (File file : files) {
            if (file.isDirectory()) {
                System.out.println(indent + "[" + file.getNam
                listFilesAndDirectories(file, indent + "  ");
            } else {
                System.out.println(indent + file.getName());
            }
        }
    }
}
```