

```
import java.io.File;
import java.io.FileNotFoundException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class FileOperationsMenu {

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

        while (true) {
            System.out.println("\n--- File Operations Menu ---");
            System.out.println("1. Check if file exists");
            System.out.println("2. Get last modification date and time");
            System.out.println("3. Rename a file");
            System.out.println("4. Create a directory/folder");
            System.out.println("5. Check if file can be read");
            System.out.println("6. Exit");
            System.out.print("Enter your choice (1-6): ");

            String choice = scanner.nextLine();

            switch (choice) {
                case "1":
                    checkFileExists(scanner);
                    break;
                case "2":
                    getLastModification(scanner);
                    break;
                case "3":
                    renameFile(scanner);
                    break;
                case "4":
                    createDirectory(scanner);
                    break;
                case "5":
                    checkFileReadPermission(scanner);
                    break;
                case "6":
                    System.out.println("Exiting program.");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice. Please enter a number between 1
and 6.");
            }
        }
    }

    private static void checkFileExists(Scanner scanner) {
        System.out.print("Enter full file path to check existence: ");
        String filePath = scanner.nextLine();
        File file = new File(filePath);

        try {
            if (!file.exists()) {
                throw new FileNotFoundException("File not found at: " + filePath);
            }
            System.out.println("File found at: " + filePath);
        } catch (FileNotFoundException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }

    private static void getLastModification(Scanner scanner) {
```

```
System.out.print("Enter full file path to get last modification date: ");
String filePath = scanner.nextLine();
File file = new File(filePath);

if (file.exists()) {
    long lastModified = file.lastModified();
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    System.out.println("Last modified: " + sdf.format(new Date(lastModified)));
} else {
    System.out.println("File does not exist.");
}
}

private static void renameFile(Scanner scanner) {
    System.out.print("Enter full path of the existing file to rename: ");
    String oldFilePath = scanner.nextLine();
    File oldFile = new File(oldFilePath);

    if (!oldFile.exists()) {
        System.out.println("Original file does not exist.");
        return;
    }

    System.out.print("Enter new full path and filename: ");
    String newFilePath = scanner.nextLine();
    File newFile = new File(newFilePath);

    boolean success = oldFile.renameTo(newFile);
    if (success) {
        System.out.println("File renamed successfully to: " + newFilePath);
    } else {
        System.out.println("Failed to rename file.");
    }
}

private static void createDirectory(Scanner scanner) {
    System.out.print("Enter full directory path to create: ");
    String dirPath = scanner.nextLine();
    File dir = new File(dirPath);

    if (!dir.exists()) {
        boolean created = dir.mkdir();
        if (created) {
            System.out.println("Directory created at: " + dir.getAbsolutePath());
        } else {
            System.out.println("Failed to create directory.");
        }
    } else {
        System.out.println("Directory already exists.");
    }
}

private static void checkFileReadPermission(Scanner scanner) {
    System.out.print("Enter full file path to check read permission: ");
    String filePath = scanner.nextLine();
    File file = new File(filePath);

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

    if (file.canRead()) {
        System.out.println("File can be read.");
    } else {
        System.out.println("File cannot be read.");
    }
}
```

```
    }  
  }  
}
```

--- File Operations Menu ---

1. Check if file exists
2. Get last modification date and time
3. Rename a file
4. Create a directory/folder
5. Check if file can be read
6. Exit

Enter your choice (1-6): 4

Enter full directory path to create: NewFolder

Directory created at: /home/AI-B2/Documents/Siddharth/OOPS/Exercise 9/NewFolder

--- File Operations Menu ---

1. Check if file exists
2. Get last modification date and time
3. Rename a file
4. Create a directory/folder
5. Check if file can be read
6. Exit

Enter your choice (1-6): 1

Enter full file path to check existence: NewFolder

File found at: NewFolder

--- File Operations Menu ---

1. Check if file exists
2. Get last modification date and time
3. Rename a file
4. Create a directory/folder
5. Check if file can be read
6. Exit

Enter your choice (1-6): 2

Enter full file path to get last modification date: NewFolder

Last modified: 23/10/2025 09:40:05

--- File Operations Menu ---

1. Check if file exists
2. Get last modification date and time
3. Rename a file
4. Create a directory/folder
5. Check if file can be read
6. Exit

Enter your choice (1-6): 3

Enter full path of the existing file to rename: NewFolder

Enter new full path and filename: NewFolderRenamed

File renamed successfully to: NewFolderRenamed

--- File Operations Menu ---

1. Check if file exists
2. Get last modification date and time
3. Rename a file
4. Create a directory/folder
5. Check if file can be read
6. Exit

Enter your choice (1-6): 5

Enter full file path to check read permission: NewFolderRenamed

File can be read.

--- File Operations Menu ---

1. Check if file exists
2. Get last modification date and time
3. Rename a file
4. Create a directory/folder
5. Check if file can be read
6. Exit

Enter your choice (1-6): 6

Exiting program.