PRACTICAL-12

APPLET LIFECYCLE

CODE:-

```
import java.applet.Applet;
import java.awt.Graphics;

public class LifeCycleApplet extends Applet {

    public void init() {
        System.out.println("Applet initialized");
    }
    public void start() {
        System.out.println("Applet started");
    }
    public void paint(Graphics g) {
        System.out.println("Painting the applet");
        g.drawString("Hello, Applet!", 20, 20);
    }
    public void stop() {
        System.out.println("Applet stopped");
    }
    public void destroy() {
        System.out.println("Applet destroyed");
    }
}
```

OUTPUT:

```
Applet initialized
Applet started
Painting the applet
Applet stopped
Applet destroyed
```

PRACTICAL-13

READING AND WRITING FROM APARTICULAR FILE

CODE: Reading

```
import java.io.*;
public class FileReaderExample {
public static void main(String[] args) {
String filePath = "example.txt";
try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
String line
while ((line = reader.readLine()) != null) {
System.out.println(line);
 }
 } catch (IOException e) {
 System.out.println("An error occurred while reading the file");
  e.printStackTrace();
   }
  }
}
```

OUTPUT:-

```
Hello, world!
This is a test.
Java file reading example.
```

CODE:Writing

```
import java.io.*;
public class WriteToFileExample {
public static void main(String[] args) {
String fileName = "output.txt";
String content = "Hello, this is a sample text written to a file.";
try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {
writer.write(content);
writer.newLine();
writer.write("This is another line of text.");
 System.out.println("Data written to file successfully.");
 } catch (IOException e) {
 System.out.println("An error occurred while writing to the file.");
 e.printStackTrace();
}
}
}
```

OUTPU

```
Data written to file successfully.

Hello, this is a sample text written to a file.
This is another line of text.
```

PRACTICAL-14

STRING BUFFER CLASS AND ITS METHODS

CODE:

```
public class StringBufferExample {
   public static void main(String[] args) {
      StringBuffer sb = new StringBuffer("Hello");
      sb.append(" World!");
      System.out.println("After append: " + sb);
      sb.insert(6, "Beautiful");
      System.out.println("After insert: " + sb);
      sb.replace(6, 15, "Amazing");
      System.out.println("After replace: " + sb);
      sb.delete(6, 13);
      System.out.println("After delete: " + sb);
      sb.reverse();
      System.out.println("After reverse: " + sb);
      sb.reverse();
      System.out.println("After reversing back: " + sb);
      System.out.println("Current capacity: " + sb.capacity());
      sb.setLength(5);
      System.out.println("After setting length to 5: " + sb);
      sb.ensureCapacity(50);
      System.out.println("Capacity after ensureCapacity(50): " + sb.capacity());
      System.out.println("Character at index 1: " + sb.charAt(1));
      System.out.println("Current length: " + sb.length());
   }
}
```

OUTPUT:

```
After append: Hello World!
After insert: Hello Beautiful World!
After replace: Hello Amazing World!
After delete: Hello World!
After reverse: IdiroW olleH
After reversing back: Hello World!
Current capacity: 21
After setting length to 5: Hello
Capacity after ensureCapacity(50): 50
Character at index 1: e
Current length: 5
```