Date :-

Exp No :- (11) a)

Write a program to read from and write to files using Byte Stream

```
import java.io.*;
public class ByteStream {
  public static void main (String[] args) throws IOException
  {
    FileInputStream sourceStream = null;
    FileOutputStream targetStream = null;
    try {
        sourceStream = new FileInputStream("sourcefile.txt");
        targetStrean = new FileOutputStream("targetfile.txt");
        int temp;
        while ((temp = sourceStream.read()) != -1)
            targetStream.write((byte)temp);

    }
    finally
    {
      if (sourceStrean != null)
          sourceStrean.close();
      if (targetStrean != null)
          targetStrean.close();
    }
}

}
```

Output:-

Contents of the file are copied.

Date :-

Exp No :- (11) b)

Write a program to both read and write to files using character streams.

```
import java.io.*;
public class CharacterStream {
  public static void main (String[] args) throws IOException
  {
    FileReader SourceStream = null;
    File try {
        sourceStream = new FileReader ("test.txt");
        int temp;
        while ((temp = sourceStream.read()) != -1)
            System.out.println ((char) temp);
    }
    finally
    {
      if (sourceStream != null)
          sourceStream.close();

    }

  }
}
```

Output:

Shows counts of file test.txt:

Date :-

Exp No:-

(11) b)

Date:-

Exp No:- 12

6) Write a java program that displays the number of characters, lines and words in a text file.

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
class CountFile {
public static void main(String[] args) throws IOException
{
  BufferedReader br = null;
  FileReader fr = null;
  int charCount = 0;
  int WordCount = 0;
  int LineCount = 0;
  try {
     fr = new FileReader("file1.txt");
     br = new BufferedReader(fr);
     String currentline = br.readline();
    while (currentline != null)
    {
      LineCount++;
      String[] words = currentline.split(" ");
      WordCount = wordCount + words.length;
      for (String word: words)
      {
        charCount = charCount + word.length();
      }
      currentline = br.readLine();
}
```

Output:-

Number of characters in file1 : 841
Number of words in file1 : 143
Number of lines in file1 : 14

```
System.out.println("Number of characters in file1 :", +
                    charCount);
System.out.println(" Number of words in file1:" + WordCount);
System.out.println(" Number of Lines in file1: " + LineCount);

catch (Exception e)
{
  System.out.println(" File not found");
}
finally
{
  br.close();
}
}
```

Date:-

Exp NO:- (13)

Write a program to check the Characters of a file after getting the filename from the user.

```
import java.io.File;
class fileProperty {
    public static void main(String[] args) {
        String fname = args[0];
        File f = new File(fname);
        System.out.println("File name:" + f.getName());
        System.out.println("Path:" + f.getPath());
        System.out.println("Absolute Path:" + f.getAbsolutePath());
        System.out.println("Parent:" + f.getParent());
        System.out.println("Exists:" + f.exists());
        if (f.exists()) {
            System.out.println("Is Writable:" + f.canWrite());
            System.out.println("Is Readable:" + f.canRead());
            System.out.println("Is a Directory:" + f.isDirectory());
            System.out.println("File Size in bytes:" + f.length());
        }
    }
}
```

Output:-

File name: file.txt
Path : file.txt
Absolute Path : C:\Users\Yash\CVR\src\file.txt
Parent : null
Exists : true
Is Writable : true
Is Readable : true
Is a directory : false
Full size in byts : 20

Title:- 8) Develop a Java Application demonstrating serialization and de-serialization.

Exp No:- (14)

```
import java.io.*;
class Demo implements java.io.Serializable {
    public int a;
    public String b;
    public Demo(int a, String b) {
        this.a = a;
        this.b = b;
    }
}

class Test {
public static void main(String[] args) {
    Demo object = new Demo(1, "Yash");
    String filename = "file.ser";
    try {
        FileOutputStream file = new FileOutputStream(filena);
        ObjectOutputStream out = new ObjectOutputStream(file);
        out.writeObject(object);
        out.close();
        file.close();
        System.out.println("Object has been Serialized");
    }
    catch(IOException ex) {
        System.out.println("IOException is caught");
    }
    Demo object1 = null;
```

try {
FileInputStream file = new FileInputStream(filename);
ObjectInputStream in = new ObjectInputStream(file);
Object1 = (Demo) in.readObject();
in.close();
file.close();
System.out.println("Object has been deserialized");
System.out.println("a = " + object1.a);
System.out.println("b = " + object1.b);
}
catch (IOException en) {
System.out.println("IOException is caught");
}
catch (ClassNotFoundExceptia en) {
System.out.println("classNotFandExceptia is caught");
}
}

Output:-

Object has been serialized

Object has been deserialized

a = 1

b = Yash

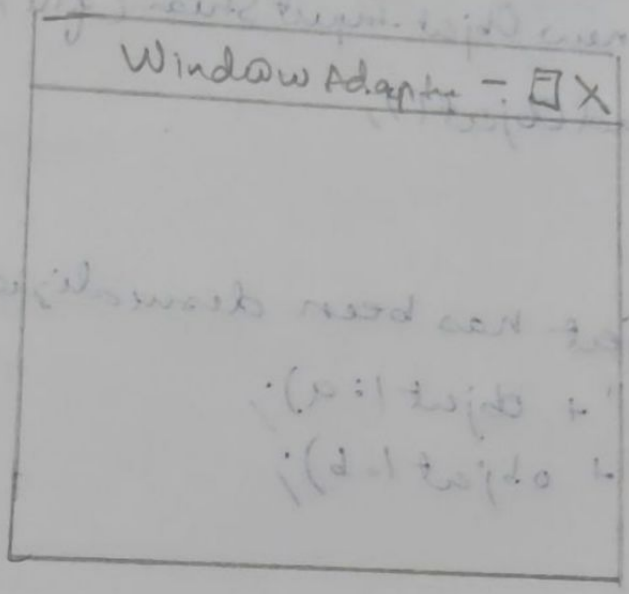
Date:-

Exp No:-
(5) a:)

Q) Write a program to demonstrate adapter classes

```
import java.awt.*;
import java.awt.event.*;
public class AdapterExample {
    Frame f;
    AdapterExample() {
        f = new Frame("Window Adapter");
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                f.dispose();
            }
        });
        f.setSize(400, 400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        new AdapterExample();
    }
}
```

Output:-



Date:-

Exp No:- 5) b)

a) Write a program to demonstrate inner classes.

```
class Outer {
    class inner {
        public void show()
        {
            System.out.println("In a nested class Method");
        }
    }
}
class Main {
    public static void main(String[] args)
    {
        Outer.Inner in = new Outer().new Inner();
        in.show();
    }
}
```

Output:-

In a Nested class Method.

Date :-

Exp No :- (16)

a) Develop an application to demonstrate applet life cycle

```
/*
<applet code = "Lifecycle.class" width = 300 height = 250>
</applet>
*/
import java.awt.*;
import java.applet.*;
public class Lifecycle extends Applet {
    String Output = " ";
    public void init()
    {
        this.setBackground(Color.yellow);
        this.setForeground(Color.red);
        Output = Output + "init";
    }
    public void start()
    {
        Output = Output + " start";
    }
    public void stop()
    {
        Output = Output + " Stop";
    }
    public void destroy()
    {
        Output = Output + " destroy";
        System.out.println(" Applet Destroying");
    }
```

```
public void paint( Graphics g)
{
   Font f = new Font("TimesNewRoman", Font.BOLD,
                     28);
        g.SetFont(f)
        g.drawString(output,100,100);
}
```

Output :-