**24) Program to draw Circle, Rectangle, Line in Applet.**

**Source Code:**

import java.awt.Color;

import java.awt.Graphics;

import javax.swing.JApplet;

public class shapes extends JApplet {

@Override

public void paint(Graphics g){

setSize(500,500);

g.drawLine(50,50,100,50);

g.setColor(Color.LIGHT\_GRAY);

g.fillRect(50, 70, 60, 60);

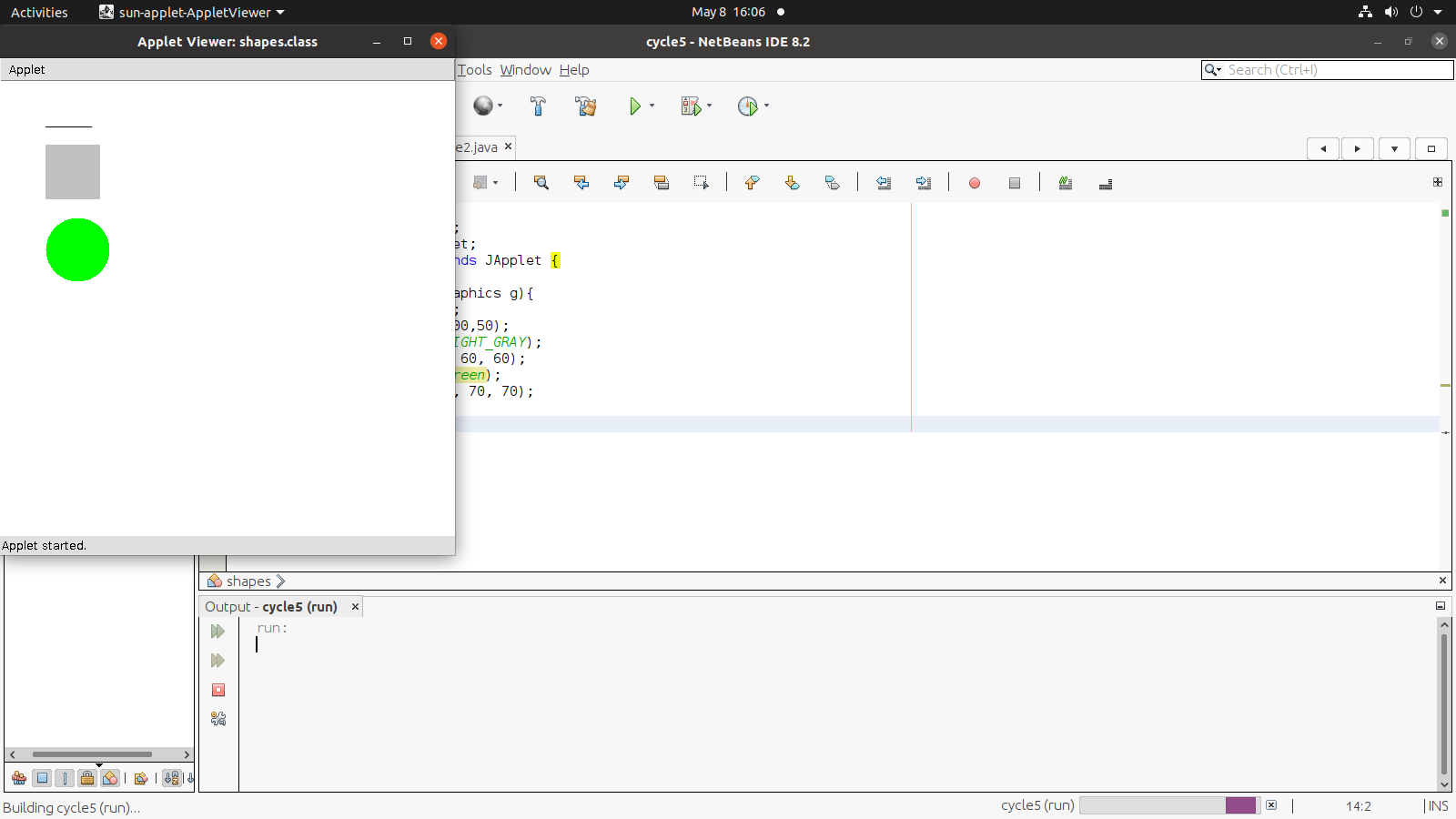
g.setColor(Color.green);

g.fillOval(50, 150, 70, 70);

}

}

**Output:**



**25) Implement a simple calculator using AWT components.**

**Source Code:**

import java.awt.\*;

import java.awt.event.\*;

public class Calculator implements ActionListener{

Frame f=new Frame();

Label l1=new Label("First Number");

Label l2=new Label("Second Number");

Label l3=new Label("Result");

TextField t1=new TextField();

TextField t2=new TextField();

TextField t3=new TextField();

Button b1=new Button("Add");

Button b2=new Button("Sub");

Button b3=new Button("Mul"); Button b4=new Button("Div");

Button b5=new Button("Cancel"); Calculator(){ l1.setBounds(50,100,100,20); l2.setBounds(50,140,100,20); l3.setBounds(50,180,100,20); t1.setBounds(200,100,100,20); t2.setBounds(200,140,100,20); t3.setBounds(200,180,100,20); b1.setBounds(50,250,50,20); b2.setBounds(110,250,50,20); b3.setBounds(170,250,50,20); b4.setBounds(230,250,50,20); b5.setBounds(290,250,50,20);

f.add(l1); f.add(l2); f.add(l3); f.add(t1); f.add(t2); f.add(t3); f.add(b1);

f.add(b2); f.add(b3); f.add(b4); f.add(b5);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

f.setLayout(null); f.setVisible(true);

f.setSize(400,350); }

public void actionPerformed(ActionEvent e){

int n1=Integer.parseInt(t1.getText());

int n2=Integer.parseInt(t2.getText());

if(e.getSource()==b1){ t3.setText(String.valueOf(n1+n2));

} if(e.getSource()==b2){ t3.setText(String.valueOf(n1-n2));

} if(e.getSource()==b3){ t3.setText(String.valueo f(n1\*n2));

} if(e.getSource()==b4){ t3.setText(String.valueOf(n1/n2));

} if(e.getSource()==b5){

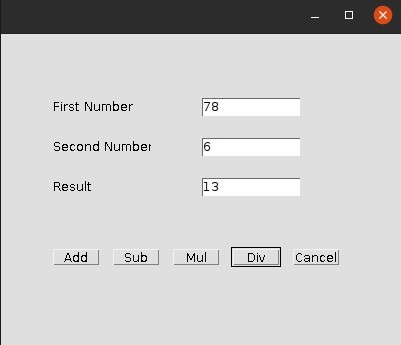
System.exit(0);

} } public static void main(String...s)

{ new Calculator();

} }

# Output:



**26)** **Program to list the sub directories and files in each directory and search for a file name.**

**Source Code:**

import java.io.File;

import java.util.Scanner;

public class DirectoryLister {

public static void main(String[] args) {

System.out.println("Program to list the sub directories and files in each directory and search for a filename.");

System.out.println("");

Scanner scanner = new Scanner(System.in);

// Ask user for the root directory

System.out.print("Enter the root directory: ");

String rootDir = scanner.nextLine();

// Ask user for the file name to search

System.out.print("Enter the file name to search for: ");

String fileNameToSearch = scanner.nextLine();

// Start listing and searching

System.out.println("Listing files and directories under: " + rootDir);

boolean found = listDirectory(new File(rootDir), fileNameToSearch);

if (!found) {

System.out.println("File not found: " + fileNameToSearch);

}

scanner.close();

}

public static boolean listDirectory(File dir, String fileNameToSearch) {

boolean found = false;

if (!dir.isDirectory()) {

System.out.println(dir.getAbsolutePath() + " is not a directory.");

return found;

}

File[] files = dir.listFiles();

if (files != null) {

System.out.println("Directory: " + dir.getName());

// Collect file names in a string builder for the required format

StringBuilder fileNames = new StringBuilder();

for (File file : files) {

if (file.isDirectory()) {

found = listDirectory(file, fileNameToSearch) || found; // Recursively list subdirectories

} else {

fileNames.append(file.getName()).append("\t");

if (file.getName().equalsIgnoreCase(fileNameToSearch)) {

System.out.println("Found the file: " + file.getAbsolutePath());

found = true;

}

}

}

// Print all file names in the current directory

if (fileNames.length() > 0) {

System.out.println(fileNames.toString().trim());

}

System.out.println(); // Add a newline after each directory's listing

} else {

System.out.println("Could not access directory: " + dir.getAbsolutePath());

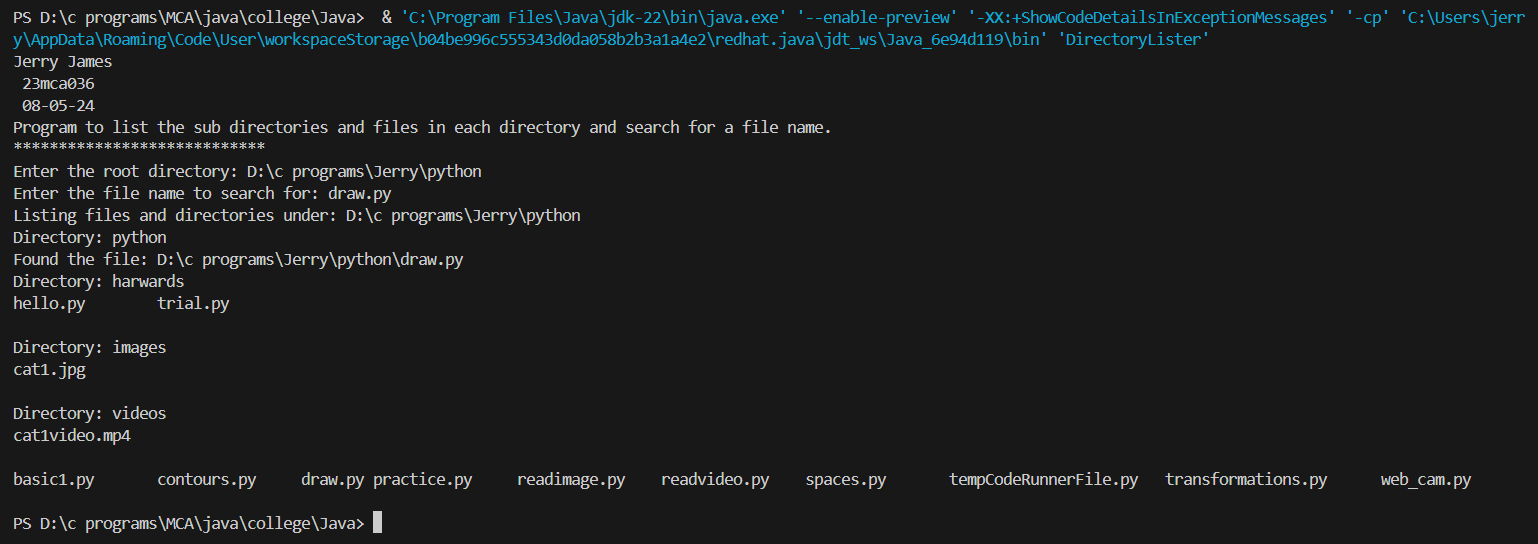
}

return found;

}

}

**Output:**



**27)** **Write a program to write to a file, then read from the file and display the contents on the console.**

**Source Code:**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class FileReadWrite {

public static void main(String[] args) {

System.out.println("Write a program to write to a file, then read from the file and display the contents on the console. ");

System.out.println("");

Scanner scanner = new Scanner(System.in);

// Ask user for the file name

System.out.print("Enter the file name to write to and read from: ");

String fileName = scanner.nextLine();

// Write to the file

System.out.println("Enter text to write to the file (type 'exit' to finish):");

try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {

String line;

while (!(line = scanner.nextLine()).equalsIgnoreCase("exit")) {

writer.write(line);

writer.newLine();

}

} catch (IOException e) {

System.err.println("Error writing to the file: " + e.getMessage());

}

// Read from the file and display the contents

System.out.println("\nContents of the file:");

try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {

String line;

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

System.err.println("Error reading from the file: " + e.getMessage());

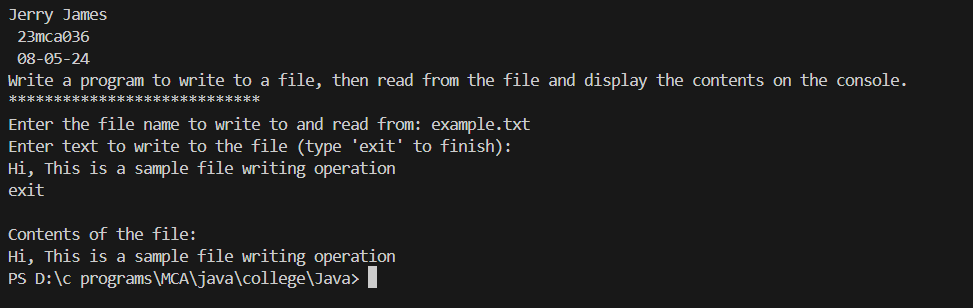
}

scanner.close();

}

}

**Output:**



**28) Write a program to copy one file to another.**

**Source Code:**

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Scanner;

public class FileCopy {

public static void main(String[] args) {

System.out.println("Write a program to copy one file to another. ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Scanner scanner = new Scanner(System.in);

// Ask user for the source file name

System.out.print("Enter the source file name: ");

String sourceFileName = scanner.nextLine();

// Ask user for the destination file name

System.out.print("Enter the destination file name: ");

String destinationFileName = scanner.nextLine();

// Copy the contents from source file to destination file

try (FileInputStream fis = new FileInputStream(sourceFileName);

FileOutputStream fos = new FileOutputStream(destinationFileName)) {

byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = fis.read(buffer)) != -1) {

fos.write(buffer, 0, bytesRead);

}

System.out.println("File copied successfully.");

} catch (IOException e) {

System.err.println("Error during file copy: " + e.getMessage());

}

scanner.close();

}

}

**Output:**

