



[\*] Untitled1

```
1 //name pushpendra rawat
2 //rollno 2001109
3 //sec MCA 20
4
5 import java.io.*;
6 import java.io.IOException;
7 import java.util.*;
8
9 public class FileOrg {
10     public static void main(String[] args) {
11         try {
12             File myObj = new File("filename");
13             if (myObj.createNewFile())
14             {
15                 System.out.println("File created with the name " + myObj.getName());
16
17                 // To read the write content on the File.....
18
19                 FileWriter myWriter = new FileWriter("file name");
20
21                 System.out.println("Enter Content");
22                 Scanner input = new Scanner(System.in);
23                 String str = input.nextLine();
24                 myWriter.write(str);
25                 myWriter.close();
26
27                 // To show the output of the file.
28
29                 System.out.println("The content of the files are as follows");
30                 String line = null;
31                 FileReader fileReader = new FileReader("file name");
32
33                 BufferedReader bufferedReader = new BufferedReader(fileReader);
34
35                 while((line = bufferedReader.readLine()) != null)
36                     ;
37             }
```

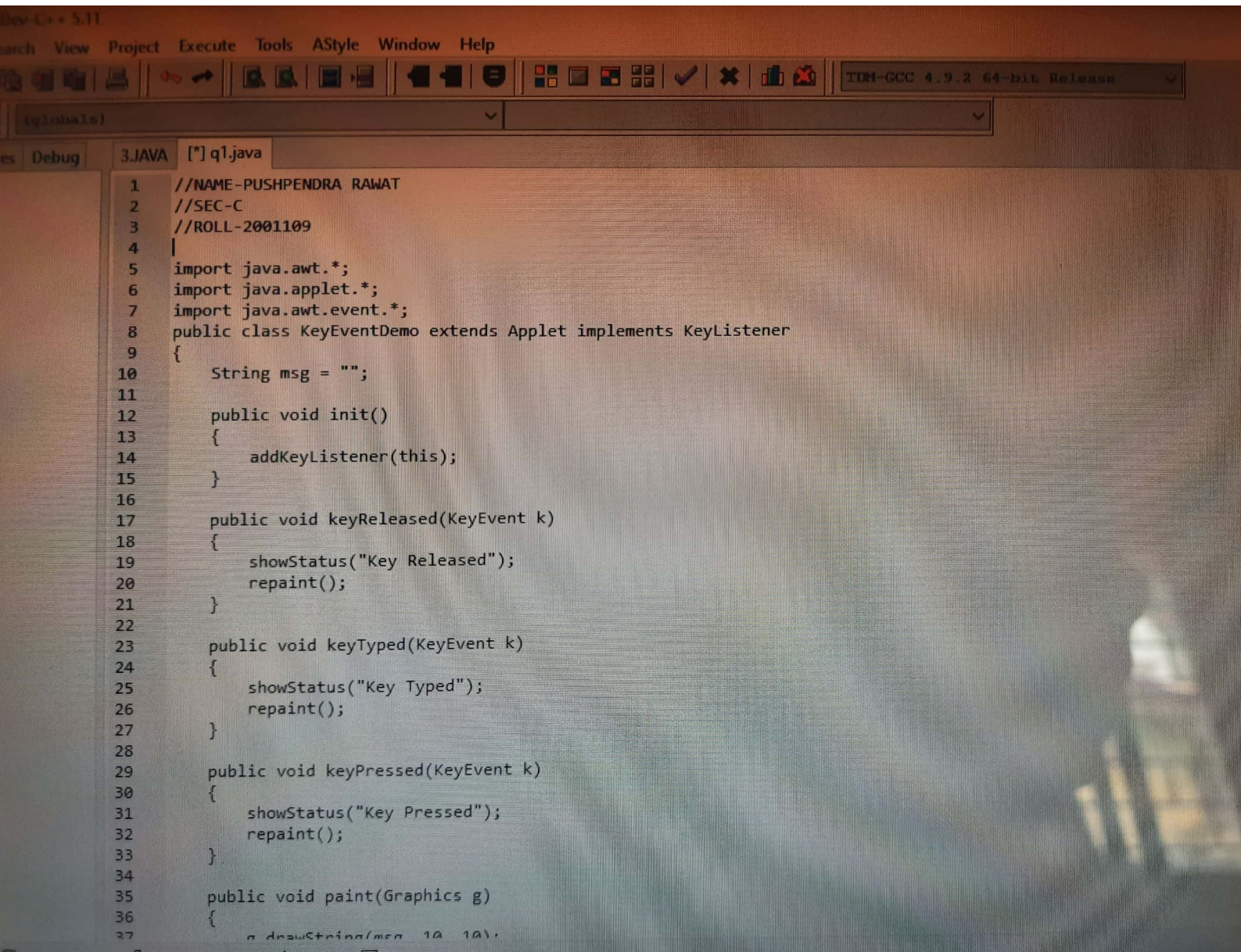


```

24 Scanner input = new Scanner(System.in);
25 String str = input.nextLine();
26 myWriter.write(str);
27 myWriter.close();
28
29 // To show the output of the file.
30
31 System.out.println("The content of the files are as follows");
32 String line = null;
33 FileReader fileReader = new FileReader("file name");
34
35 BufferedReader bufferedReader = new BufferedReader(fileReader);
36
37 while((line = bufferedReader.readLine()) != null)
38 {
39     System.out.println(line);
40 }
41 bufferedReader.close();
42
43 else {
44     System.out.println("File already exists.");
45 }
46
47 catch (IOException e) {
48     System.out.println("An error occurred.");
49     e.printStackTrace();
50
51 }
52
53 }

```







```
Debug | 3.JAVA | [*] q1.java
4 |
5 | import java.awt.*;
6 | import java.applet.*;
7 | import java.awt.event.*;
8 | public class KeyEventDemo extends Applet implements KeyListener
9 | {
10 |     String msg = "";
11 |
12 |     public void init()
13 |     {
14 |         addKeyListener(this);
15 |     }
16 |
17 |     public void keyReleased(KeyEvent k)
18 |     {
19 |         showStatus("Key Released");
20 |         repaint();
21 |     }
22 |
23 |     public void keyTyped(KeyEvent k)
24 |     {
25 |         showStatus("Key Typed");
26 |         repaint();
27 |     }
28 |
29 |     public void keyPressed(KeyEvent k)
30 |     {
31 |         showStatus("Key Pressed");
32 |         repaint();
33 |     }
34 |
35 |     public void paint(Graphics g)
36 |     {
37 |         g.drawString(msg, 10, 10);
38 |     }
39 | }
```

```
5.11
File Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
3.JAVA [*] q1.java [*] Untitled3
1 //name-pushpendra rawat
2
3 import java.io.*;
4 import java.net.*;
5
6 public class udp_client
7 {
8     public static void main(String args[])
9     {
10         DatagramSocket sock = null;
11         int port = 7777;
12         String s;
13
14         BufferedReader cin = new BufferedReader(new InputStreamReader(System.in));
15
16         try
17         {
18             sock = new DatagramSocket();
19             InetAddress host = InetAddress.getByName("localhost");
20
21             while(true)
22             {
23                 //take input and send the packet
24                 echo("Enter message to send : ");
25                 s = (String)cin.readLine();
26                 byte[] b = s.getBytes();
27
28                 DatagramPacket dp = new DatagramPacket(b , b.length , host , port);
29                 sock.send(dp);
30
31                 //now receive reply
32                 //buffer to receive incoming data
33                 byte[] buffer = new byte[65536];
34                 DatagramPacket reply = new DatagramPacket(buffer, buffer.length);
35                 sock.receive(reply);
36
37             }
38         }
39     }
40 }
```



```

22 while(true)
23 {
24     //take input and send the packet
25     echo("Enter message to send : ");
26     s = (String)cin.readLine();
27     byte[] b = s.getBytes();
28
29     DatagramPacket dp = new DatagramPacket(b , b.length , host , port);
30     sock.send(dp);
31
32     //now receive reply
33     //buffer to receive incoming data
34     byte[] buffer = new byte[65536];
35     DatagramPacket reply = new DatagramPacket(buffer, buffer.length);
36     sock.receive(reply);
37
38     byte[] data = reply.getData();
39     s = new String(data, 0, reply.getLength());
40
41     //echo the details of incoming data - client ip : client port - client message
42     echo(reply.getAddress().getHostAddress() + " : " + reply.getPort() + " - " + s);
43 }
44
45 catch(IOException e)
46 {
47     System.err.println("IOException " + e);
48 }
49
50 //simple function to echo data to terminal
51 public static void echo(String msg)
52 {
53     System.out.println(msg);
54 }

```