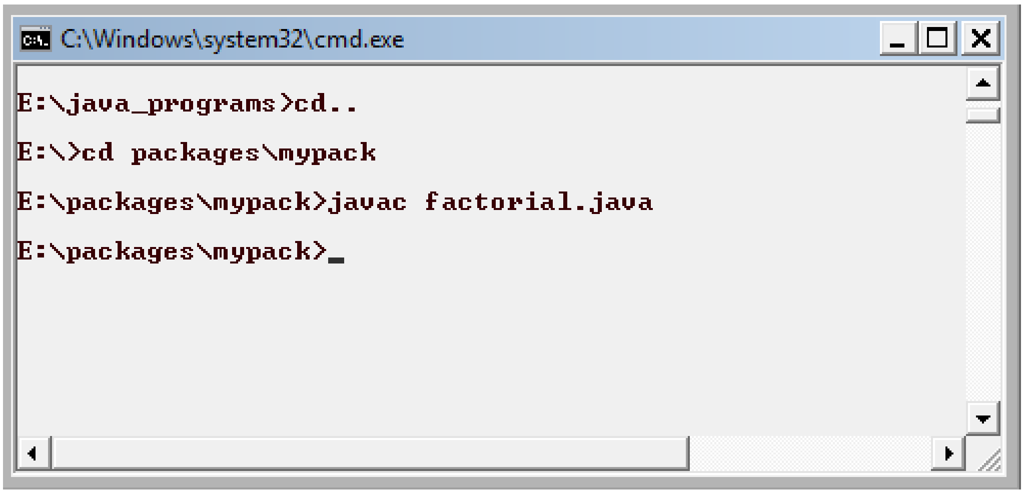
EXERCISE 12

a). Write a JAVA program illustrate class path

PROGRAM:

```
package mypack;
    public class Factorial{
          public int fact(int a){
                if(a == 1)
                     return 1;
                 else
                      return a*fact(a-1);
                }
        }
```

OUTPUT: compile the program of the specific class path



```
E:\java_programs>cd..

E:\>cd packages\mypack

E:\packages\mypack>javac factorial.java

E:\packages\mypack>_
```

This program will not run it does not contain main method.

b). Write a case study on including in class path in your os environment of your package.

Classpath is used for storing the path of the third-party and user-defined classes. Whenever we execute/compile any class file, jdk tools javac and java, search the package/class file in the user classpath which is the current directory by default. If the classes are not in the current directory,then we need to set the classpath.

The classpath can be set in two ways:

1. It is an environment variable which can be set using the *System* utility in the control panel or at the DOS prompt as shown.

```
Set CLASSPATH = %CLASSPATH%;c:\pack;
```

%Classpath% is used to keep the existing path intact and append our new path to it. Now L3 of both the above cases will execute.

2. Use classpath option –classpath or –cp of javac/java tools to override the user-defined classpath and find the user-defined specific package/classes used in the Java source files.

```
//syntax: javac –cp path of the directory/package used in java
source file
followed by name of the java source file
C:\pack\packexample> javac –cp c:\javaeg DemoClass.java
```

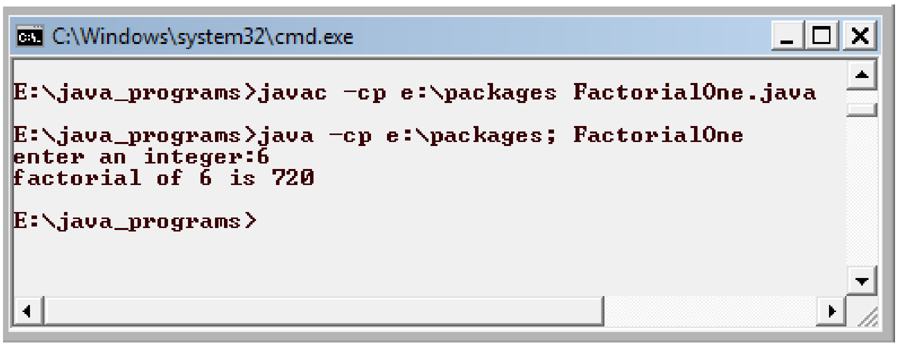
-cp specifies that the user-defined package/classes used in DemoClass.java will be found at c:\javaeg.

c). Write a JAVA program that import and use the defined your package in the previous Problem

PROGRAM:

```
import mypack.*;
import java.util.Scanner;
 class FactorialOne{
        public static void main(String args[]){
              Scanner sc = new Scanner(System.in);
              System.out.print("enter an integer:");
              int a = sc.nextInt();
              Factorial i = new Factorial();
              System.out.println("factorial of " +a+ " is " +i.fact(a));
          }
    }
```

OUTPUT:



```
E:\java_programs>javac -cp e:\packages FactorialOne.java

E:\java_programs>java -cp e:\packages; FactorialOne
enter an integer:6
factorial of 6 is 720

E:\java_programs>
```

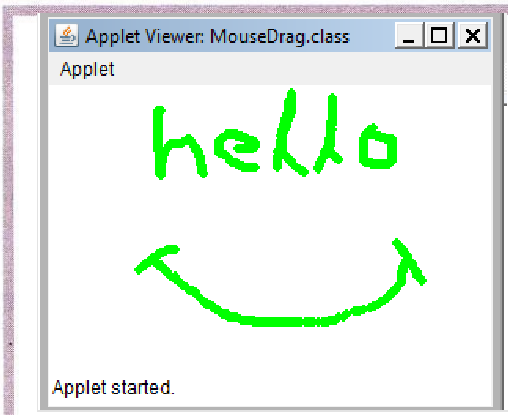
EXERCISE 13

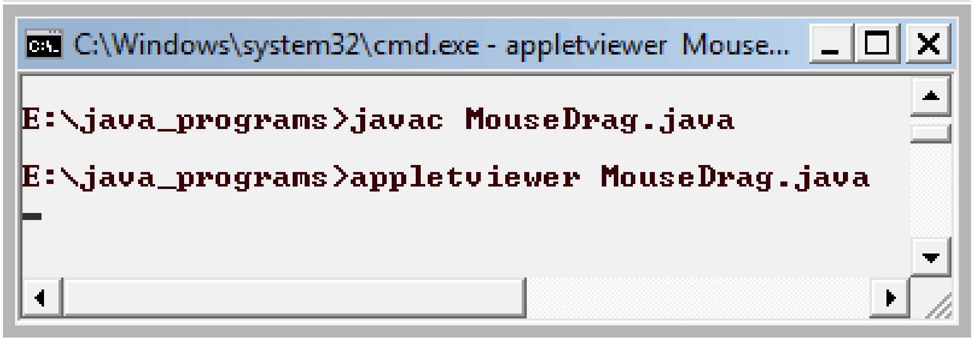
a).Write a JAVA program to paint like paint brush in applet.

PROGRAM:

```
/*<applet code="MouseDrag.class" width="300" height="300"> </applet>*/
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class MouseDrag extends Applet implements MouseMotionListener{
            public void init(){
            addMouseMotionListener(this);
            setBackground(Color.white);
            }

            public void mouseDragged(MouseEvent me){
            Graphics g=getGraphics();
            g.setColor(Color.green);
            g.fillOval(me.getX(),me.getY(),6,6);
      }
public void mouseMoved(MouseEvent me){}
  }
```

OUTPUT:





```
E:\java_programs>javac MouseDrag.java

E:\java_programs>appletviewer MouseDrag.java
_
```

b) Write a JAVA program to display analog clock using Applet.

AIM: JAVA program to display analog clock using Applet.

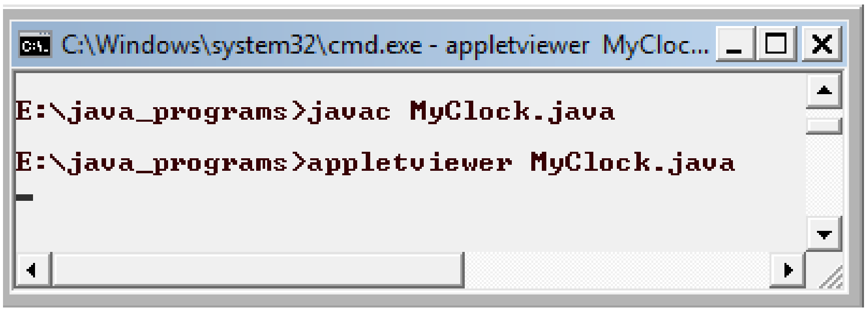
PROGRAM:

```
/*<applet code="MyClock.class" width="300" height="300">  </applet>*/
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.text.*;
public class MyClock extends Applet implements Runnable {

   int width, height;
   Thread t = null;
   boolean threadSuspended;
   int hours=0, minutes=0, seconds=0;
   String timeString = "";

   public void init() {
      width = getSize().width;
      height = getSize().height;
      setBackground( Color.black );
   }
```

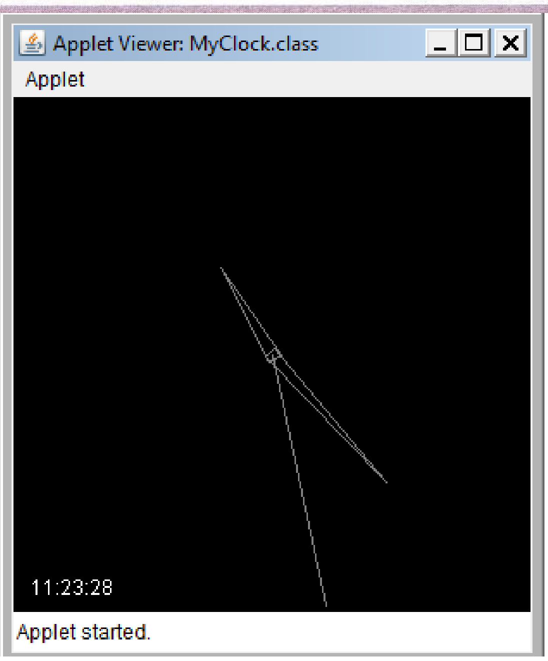
```
public void start() {
  if ( t == null ) {
    t = new Thread( this );
    t.setPriority( Thread.MIN_PRIORITY );
    threadSuspended = false;
    t.start();
  }
  else {
    if ( threadSuspended ) {
      threadSuspended = false;
      synchronized( this ) {
        notify();
      }
    }
  }
}

public void stop() {
  threadSuspended = true;
}

public void run() {
  try {
    while (true) {

      Calendar cal = Calendar.getInstance();
      hours = cal.get( Calendar.HOUR_OF_DAY );
      if ( hours > 12 ) hours -= 12;
      minutes = cal.get( Calendar.MINUTE );
      seconds = cal.get( Calendar.SECOND );

      SimpleDateFormat formatter
        = new SimpleDateFormat( "hh:mm:ss", Locale.getDefault() );
      Date date = cal.getTime();
      timeString = formatter.format( date );

      // Now the thread checks to see if it should suspend itself
      if ( threadSuspended ) {
        synchronized( this ) {
          while ( threadSuspended ) {
            wait();
          }
        }
      }
      repaint();
      t.sleep( 1000 );  // interval specified in milliseconds
    }
```

```
        }
        catch (Exception e) { }
    }

    void drawHand( double angle, int radius, Graphics g ) {
        angle -= 0.5 * Math.PI;
        int x = (int)( radius*Math.cos(angle) );
        int y = (int)( radius*Math.sin(angle) );
        g.drawLine( width/2, height/2, width/2 + x, height/2 + y );
    }

    void drawWedge( double angle, int radius, Graphics g ) {
        angle -= 0.5 * Math.PI;
        int x = (int)( radius*Math.cos(angle) );
        int y = (int)( radius*Math.sin(angle) );
        angle += 2*Math.PI/3;
        int x2 = (int)( 5*Math.cos(angle) );
        int y2 = (int)( 5*Math.sin(angle) );
        angle += 2*Math.PI/3;
        int x3 = (int)( 5*Math.cos(angle) );
        int y3 = (int)( 5*Math.sin(angle) );
        g.drawLine( width/2+x2, height/2+y2, width/2 + x, height/2 + y );
        g.drawLine( width/2+x3, height/2+y3, width/2 + x, height/2 + y );
        g.drawLine( width/2+x2, height/2+y2, width/2 + x3, height/2 + y3 );
    }

    public void paint( Graphics g ) {

        g.setColor( Color.gray );
        drawWedge( 2*Math.PI * hours / 12, width/5, g );
        drawWedge( 2*Math.PI * minutes / 60, width/3, g );
        drawHand( 2*Math.PI * seconds / 60, width/2, g );
        g.setColor( Color.white );
        g.drawString( timeString, 10, height-10 );
    }
}
```

OUTPUT:



```
E:\java_programs>javac MyClock.java

E:\java_programs>appletviewer MyClock.java
_
```



11:23:28

Applet started.

c). Write a JAVA program to create different shapes and fill colors using Applet.

AIM: JAVA program to create different shapes and fill colors using Applet.

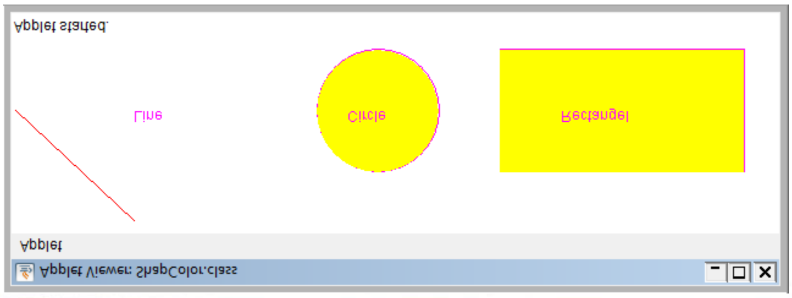
ALGORITHM:

1. Create the class named ‘ShapColor’ which inherits the Applet class.
2. Initialize the the variable values of ‘x’, ‘y’ and ‘r’.
3.Create a function named ‘pait()’ with thr Graphics instance as a parameters.
4.Use this instance ‘g’ to call the drawing function like ‘setColor()’, ‘drawLine()’, ‘fillOval()’ and many other by passing the necessary requirements.

PROGRAM:

```
/*<applet ALIGN = "CENTER"  CODE = "ShapColor.class" WIDTH = 800 HEIGHT = 200>
</applet>*/
import java.applet.*;
import java.awt.*;

public class  ShapColor extends Applet{
  int x=300,y=100,r=50;

  public void paint(Graphics g){
  g.setColor(Color.red);  //Drawing line color is red
  g.drawLine(3,100,100,10);
  g.setColor(Color.magenta);
  g.drawString("Line",100,100);

  g.drawOval(x-r,y-r,100,100);
  g.setColor(Color.yellow);  //Fill the yellow color in circle
  g.fillOval( x-r,y-r, 100, 100 );
  g.setColor(Color.magenta);
  g.drawString("Circle",275,100);

  g.drawRect(400,50,200,100);
  g.setColor(Color.yellow);  //Fill the yellow color in rectangel
  g.fillRect( 400, 50, 200, 100 );
  g.setColor(Color.magenta);
  g.drawString("Rectangel",450,100);
  }
}
```

OUTPUT:



Exercise - 14 (Event Handling)

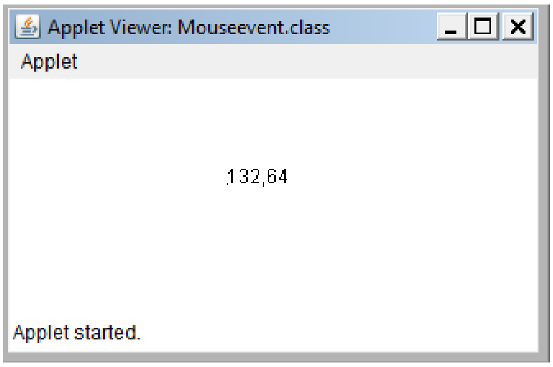
14 a).Write a JAVA program that display the x and y position of the cursor movement using Mouse.

AIM: JAVA program that display the x and y position of the cursor movement using Mouse.

PROGRAM:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
 /* <applet code = "Mouseevent.class" width = 400 height = 200> </applet> */

public class Mouseevent extends Applet
{
  int x=0;
  int y=0;
  public void init()
 {
   addMouseListener(new mymouselistener());
 }
 public void start()
 {
 }
 public void paint(Graphics g)
 {
           g.drawLine(x,y,x,y);
    g.drawString(x + ","+ y, x,y);
 }
public class mymouselistener extends MouseAdapter
 {
  public void mouseClicked(MouseEvent e)
  {
  x = e.getX();
  y = e.getY();
  repaint();
     }    }   }
```

OUTPUT:



```
132,64
```

Applet started.

b).Write a JAVA program that identifies key-up key-down event user entering text in a Applet.

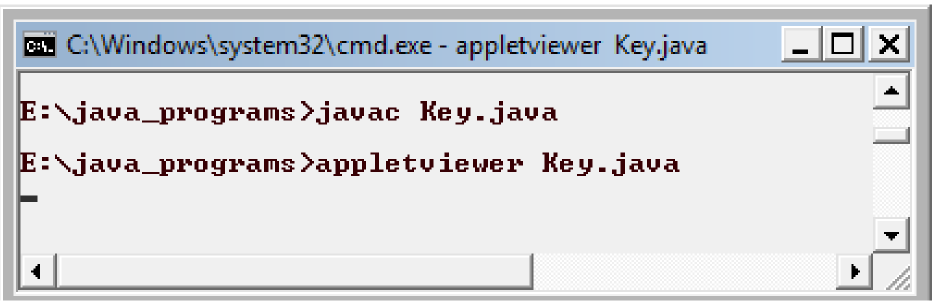
AIM: JAVA program that identifies key-up key-down event user entering text in a Applet.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Key.class" width=800 height=200>
</applet>
*/
public class Key extends Applet
implements KeyListener
{
        int X=20,Y=30;
        String msg="Type here: ";
        public void init()
        {
                addKeyListener(this);
                requestFocus();
                setBackground(Color.green);
                setForeground(Color.black);
        }
        public void keyPressed(KeyEvent k)
        {
                showStatus("KeyDown");
                int key=k.getKeyCode();
                switch(key)
                {
                        case KeyEvent.VK_UP:
                                showStatus("Move to Up");
                                break;
                        case KeyEvent.VK_DOWN:
                                showStatus("Move to Down");
                                break;
                        case KeyEvent.VK_LEFT:
                                showStatus("Move to Left");
                                break;
                        case KeyEvent.VK_RIGHT:
                                showStatus("Move to Right");
                                break;
                }
                repaint();
        }
```

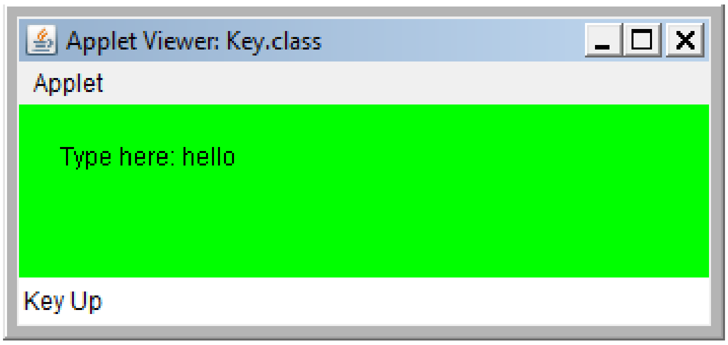
```
    public void keyReleased(KeyEvent k)
    {

        showStatus("Key Up");
    }
    public void keyTyped(KeyEvent k)
    {
        msg+=k.getKeyChar();
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(msg,X,Y);
    }
}
```

OUTPUT:



C:\Windows\system32\cmd.exe - appletviewer Key.java

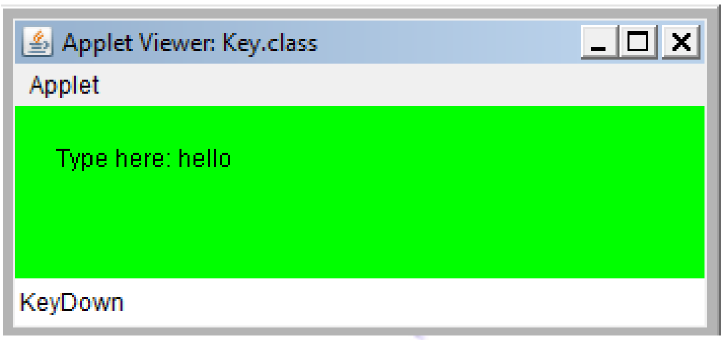
```
E:\java_programs>javac Key.java

E:\java_programs>appletviewer Key.java
_
```



Applet Viewer: Key.class

Type here: hello

Key Up



Applet Viewer: Key.class

Type here: hello

KeyDown