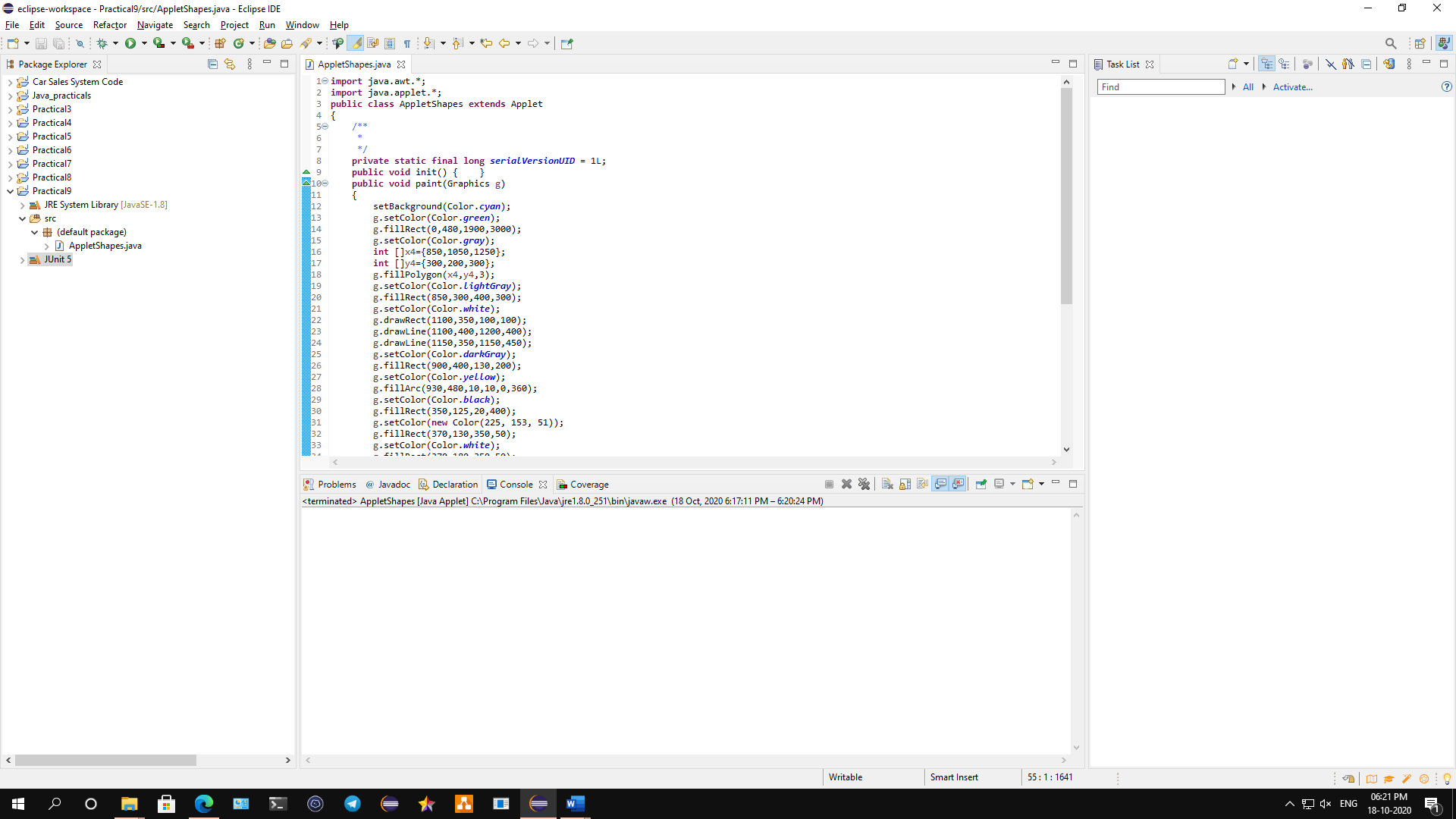
**Practical No 9**

**A) To design an applet to display shapes like: House, Ashok Chakra, Indian flag.**

**Aim: Write a program to design an applet to display shapes like: House, Ashok Chakra, Indian flag.**

**Description:**

An applet is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal. Four methods in the Applet class gives you the framework on which you build any serious applet they are: init, start, stop, destroy and paint. In this program we first defined the java awt and applet package followed by appletshapes which extends applet class. Then we define the init function and paint which is used to show shapes and graphics. Then we set background as cyan and green colours. Then we defined the required colours we need to complete the program. In the last we defined the maths funtions to make circle, rectangle and tringle shape. The output will be display showing shape like house, ashok chakra and the Indian flag.



**Conclusion: We have designed an applet to display shapes like: House, Ashok Chakra, Indian flag.**

**Code:**

**import** java.awt.\*;

**import** java.applet.\*;

**public** **class** AppletShapes **extends** Applet

{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **void** init() { }

**public** **void** paint(Graphics g)

{

setBackground(Color.***cyan***);

g.setColor(Color.***green***);

g.fillRect(0,480,1900,3000);

g.setColor(Color.***gray***);

**int** []x4={850,1050,1250};

**int** []y4={300,200,300};

g.fillPolygon(x4,y4,3);

g.setColor(Color.***lightGray***);

g.fillRect(850,300,400,300);

g.setColor(Color.***white***);

g.drawRect(1100,350,100,100);

g.drawLine(1100,400,1200,400);

g.drawLine(1150,350,1150,450);

g.setColor(Color.***darkGray***);

g.fillRect(900,400,130,200);

g.setColor(Color.***yellow***);

g.fillArc(930,480,10,10,0,360);

g.setColor(Color.***black***);

g.fillRect(350,125,20,400);

g.setColor(**new** Color(225, 153, 51));

g.fillRect(370,130,350,50);

g.setColor(Color.***white***);

g.fillRect(370,180,350,50);

g.setColor(**new** Color(19, 136, 8));

g.fillRect(370,230,350,50);

g.setColor(**new** Color(0, 0, 128));

g.drawOval(500,180,50,50);

**int** n1=525;

**int** d1=205;

**double** n2,d2;

**double** angle= 0.0;

**double** line=0.0;

**int** r=23;

**for**(**int** i=1;i<=24;i++)

{

angle=(**double**)line\*(3.14/180);

n2=n1+(**double**)r\*Math.*cos*(angle);

d2=d1+(**double**)r\*Math.*sin*(angle);

g.drawLine(n1,d1,(**int**)n2,(**int**)d2);

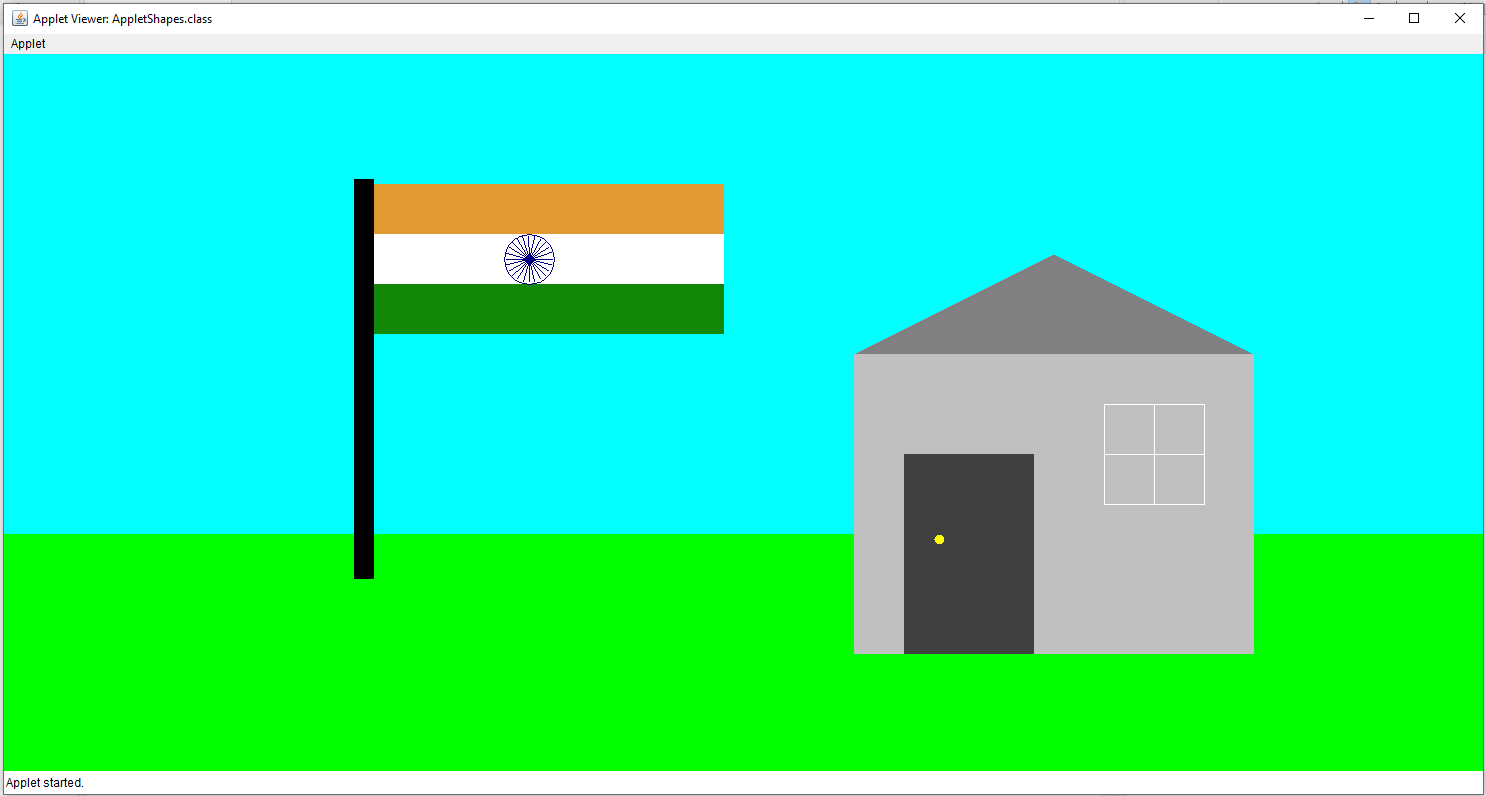
line=line+(360/24);

}

}

}

**Output:**

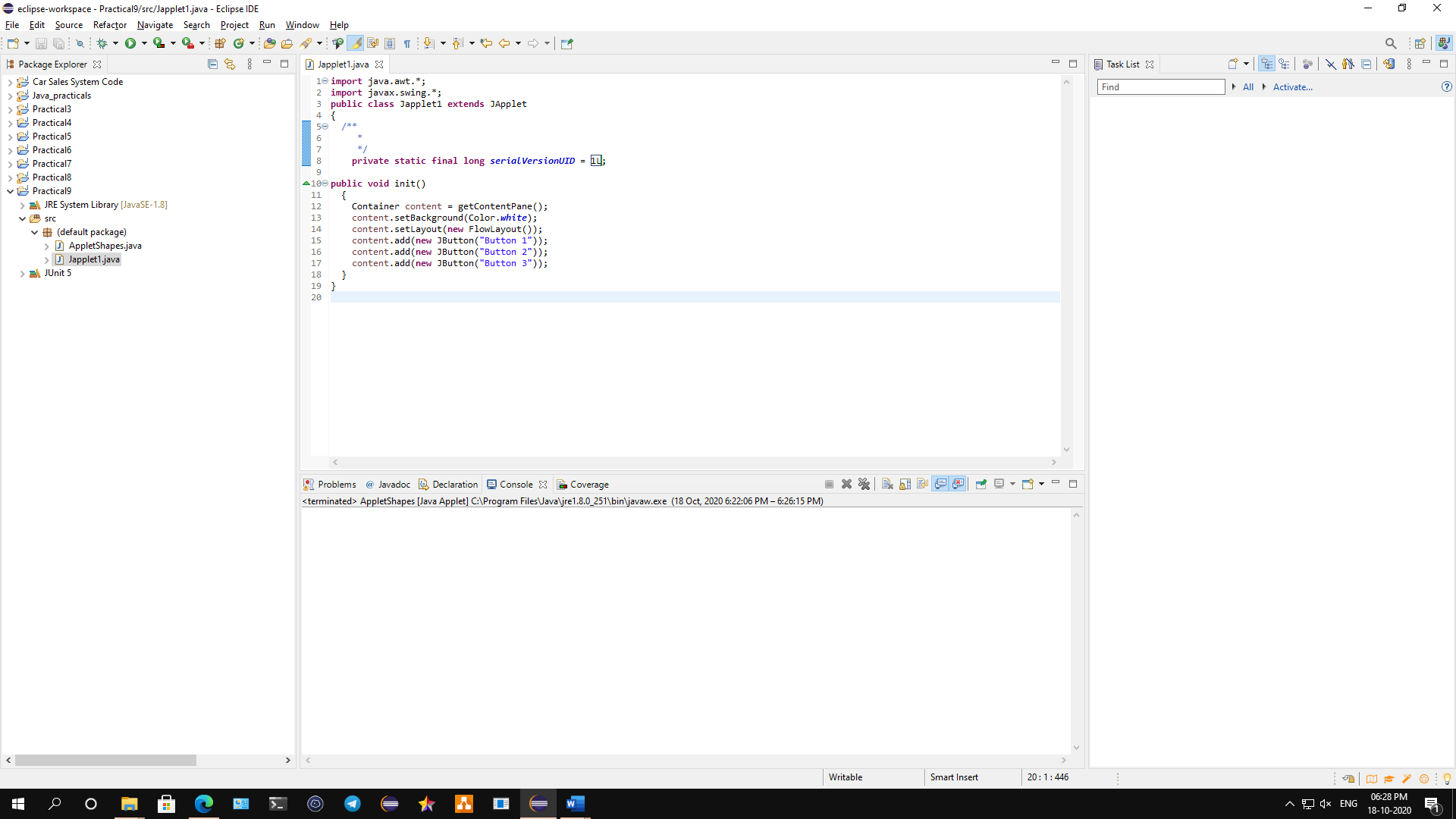


**B) To demonstrate JApplet.**

**Aim: Write a program to demonstrate JApplet.**

**Description:**

A JApplet is a Swing JPanel with a mission. It is a GUI container that has some extra structure to allow it to be used in the “alien” environment of a web browser. Applets also have a lifecycle that lets them act more like an application than a static component. We have imported the java awt and swing class, then we defined Japplet1 class which extends Japplet class, then we define the init function which show container and set background as white then we define a new button as button 1, button 2 and button 3. Then the output will shown with gui that show Button 1, Button 2 and Button 3.



**Conclusion: We have demonstrated JApplet in details.**

**Code:**

**import** java.awt.\*;

**import** javax.swing.\*;

**public** **class** Japplet1 **extends** JApplet

{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **void** init()

{

Container content = getContentPane();

content.setBackground(Color.***white***);

content.setLayout(**new** FlowLayout());

content.add(**new** JButton("Button 1"));

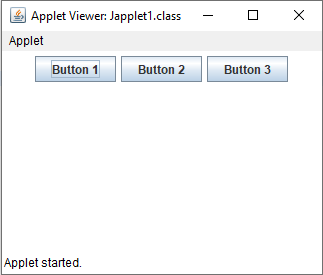
content.add(**new** JButton("Button 2"));

content.add(**new** JButton("Button 3"));

}

}

**Output:**

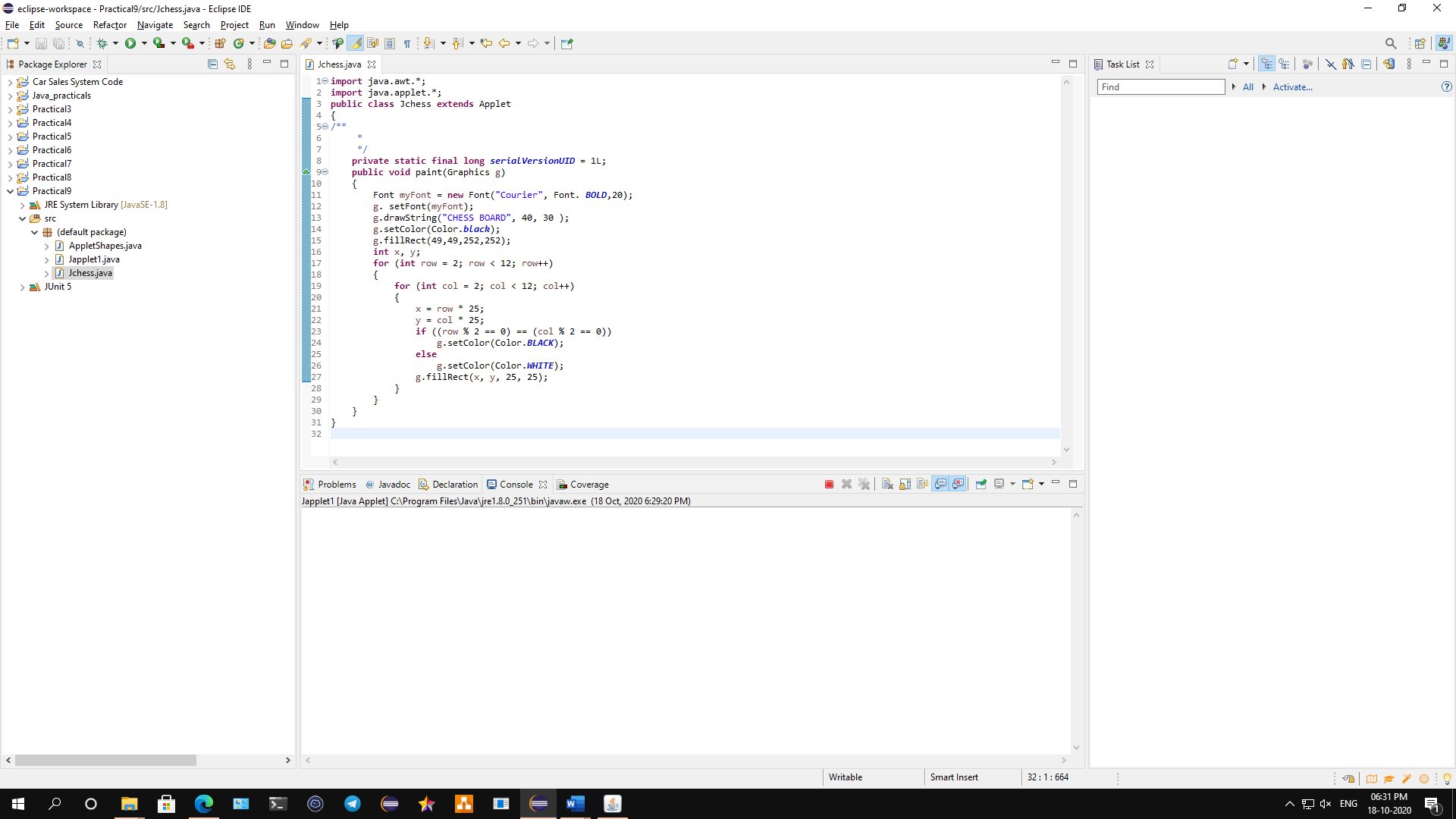


**C) To Develop Chess Board using Applet.**

**Aim: Write a program to develop a Chess Board using Applet.**

**Description:**

An applet is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal. Four methods in the Applet class gives you the framework on which you build any serious applet they are: init, start, stop, destroy and paint. In this program we import the awt and swing classes followed by the Jchess class which extends the applet class, then we defined the main function as paint which used to define the shape and graphics. Then we set Courier as default font and colours as black and white in order to make a chess board, then we define the rows and columns and the chess board has been made in java using a applet.



**Conclusion: We have developed a Chess Board using Applet.**

**Code:**

**import** java.awt.\*;

**import** java.applet.\*;

**public** **class** Jchess **extends** Applet

{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **void** paint(Graphics g)

{

Font myFont = **new** Font("Courier", Font. ***BOLD***,20);

g. setFont(myFont);

g.drawString("CHESS BOARD", 40, 30 );

g.setColor(Color.***black***);

g.fillRect(49,49,252,252);

**int** x, y;

**for** (**int** row = 2; row < 12; row++)

{

**for** (**int** col = 2; col < 12; col++)

{

x = row \* 25;

y = col \* 25;

**if** ((row % 2 == 0) == (col % 2 == 0))

g.setColor(Color.***BLACK***);

**else**

g.setColor(Color.***WHITE***);

g.fillRect(x, y, 25, 25);

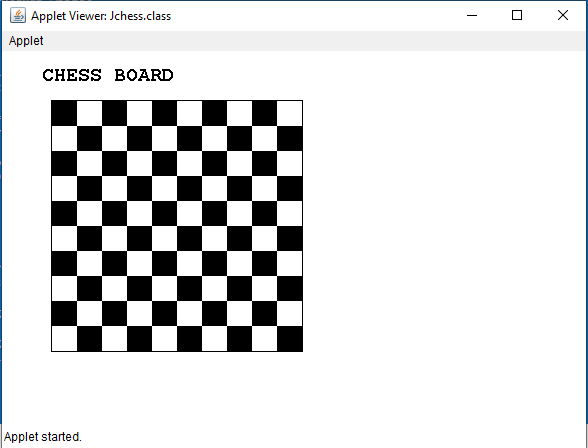
}

}

}

}

**Output:**

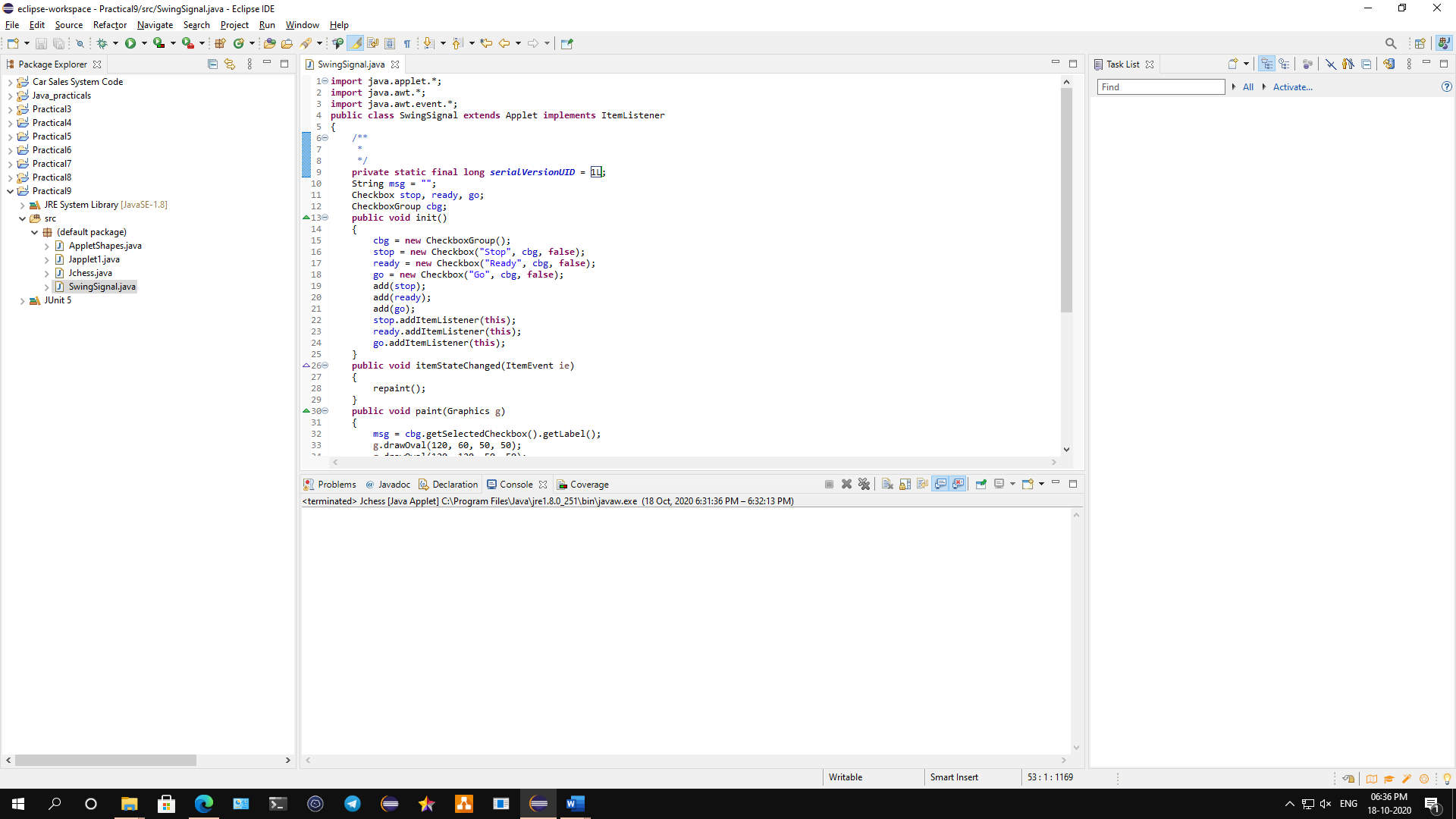


**D) Traffic Signal Simulator using Swing Components.**

**Aim: Write a program to simulate Traffic Signal using Swing Components.**

**Description:**

**Swing** is a part of the **JFC (Java Foundation Classes).** Building [Graphical User Interface](https://www.geeksforgeeks.org/what-is-the-difference-between-gui-and-cui/) in Java requires the use of Swings. **Swing Framework** contains a large set of components which allow a high level of customization and provide rich functionalities, and is used to create window-based applications. Java swing components are lightweight, platform-independent, provide powerful components like tables, scroll panels, buttons, list, colour chooser, etc. In this program we imported the applet, awt and the swing classes followed by swingsignal that extends applet and implement itemlisteners, then we define the main init funtions and then we define 3 oval shapes and gave them 3 colours as red, green and blue. The output will show simulating traffic signal.



**Conclusion: We have simulated a Traffic Signal using Swing Components.**

**Code:**

**import** java.applet.\*;

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** SwingSignal **extends** Applet **implements** ItemListener

{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

String msg = "";

Checkbox stop, ready, go;

CheckboxGroup cbg;

**public** **void** init()

{

cbg = **new** CheckboxGroup();

stop = **new** Checkbox("Stop", cbg, **false**);

ready = **new** Checkbox("Ready", cbg, **false**);

go = **new** Checkbox("Go", cbg, **false**);

add(stop);

add(ready);

add(go);

stop.addItemListener(**this**);

ready.addItemListener(**this**);

go.addItemListener(**this**);

}

**public** **void** itemStateChanged(ItemEvent ie)

{

repaint();

}

**public** **void** paint(Graphics g)

{

msg = cbg.getSelectedCheckbox().getLabel();

g.drawOval(120, 60, 50, 50);

g.drawOval(120, 120, 50, 50);

g.drawOval(120, 180, 50, 50);

**if**(msg.equals("Stop"))

{

g.setColor(**new** Color(204, 50, 50));

g.fillOval(120, 60, 50, 50);

}

**else** **if**(msg.equals("Ready"))

{

g.setColor(**new** Color(231, 180, 22));

g.fillOval(120, 120, 50, 50);

}

**else**

{

g.setColor(**new** Color(45, 201, 55));

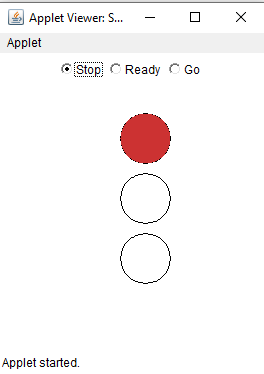
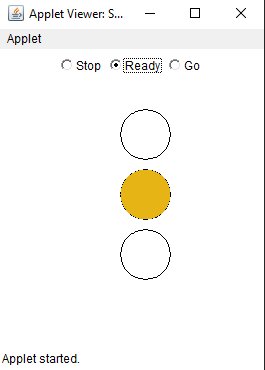
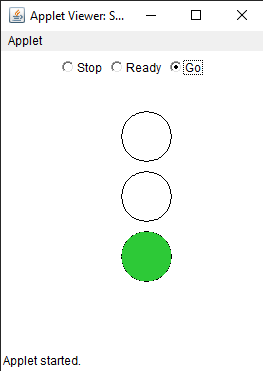
g.fillOval(120, 180, 50, 50);

}

}

}

**Output:**

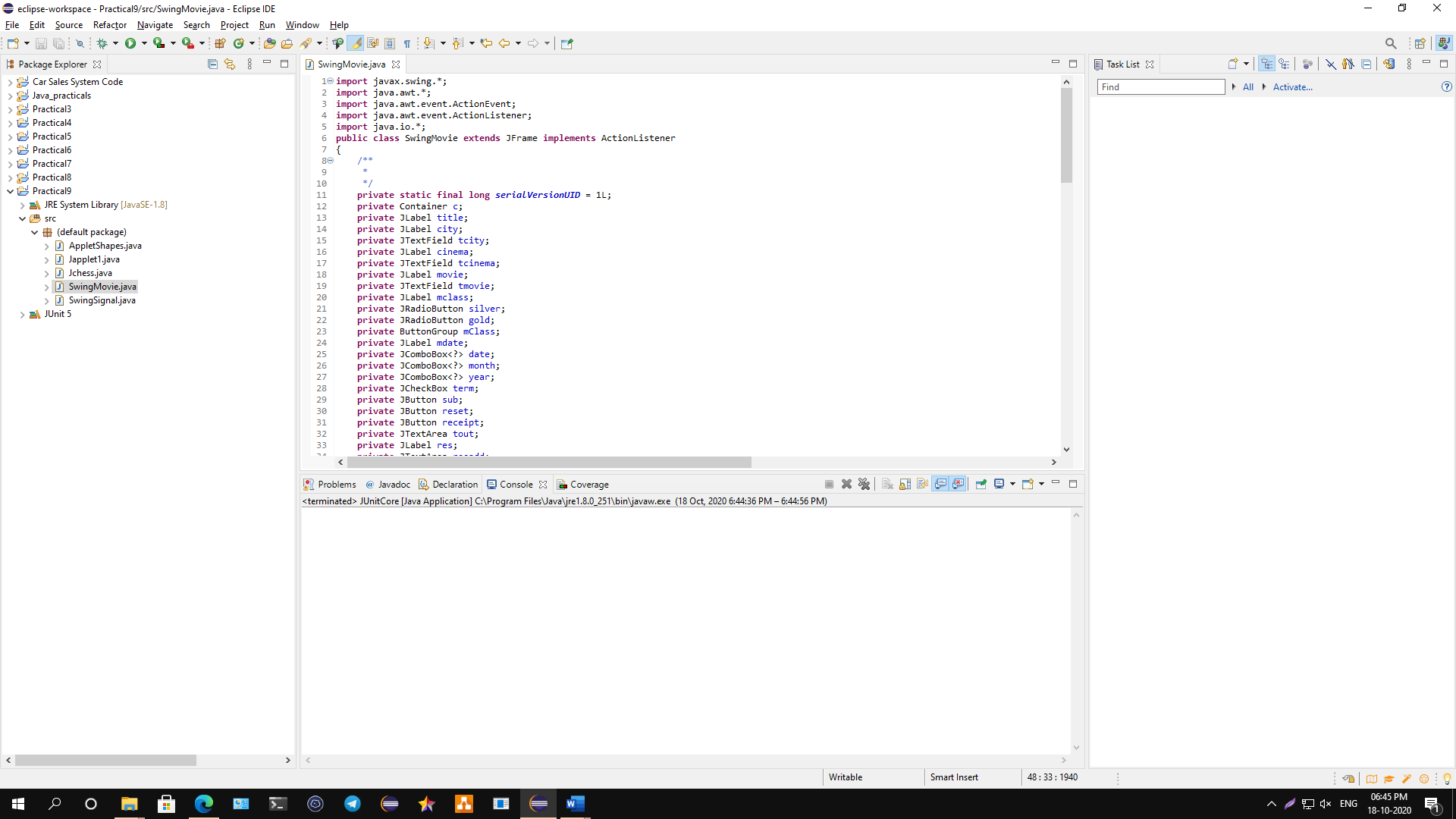
  

**E) To design a form using Swings Component for Movie Ticket Booking application.**

**Aim: Write a program to design a form using Swings Component for Movie Ticket Booking application.**

**Description:**

**Swing** is a part of the **JFC (Java Foundation Classes).** Building [Graphical User Interface](https://www.geeksforgeeks.org/what-is-the-difference-between-gui-and-cui/) in Java requires the use of Swings. **Swing Framework** contains a large set of components which allow a high level of customization and provide rich functionalities, and is used to create window-based applications. Java swing components are lightweight, platform-independent, provide powerful components like tables, scroll panels, buttons, list, colour chooser, etc. In this program we imported java swing, awt and io classes followed by swingmovie class which extends Jframe that implements actionlistner classes, then we defined the main private classes which is required to make a box and list. We then defined the containers and text required to make an application. We also defined title name city class and so on. We also use try and catch keywords for exception handling then we use the new class for obtaining the data. The application will run and we can book a ticket.



**Conclusion: We have designed a form using Swings Component for Movie Ticket Booking application.**

**Code:**

**import** javax.swing.\*;

**import** java.awt.\*;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.\*;

**public** **class** SwingMovie **extends** JFrame **implements** ActionListener

{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** Container c;

**private** JLabel title;

**private** JLabel city;

**private** JTextField tcity;

**private** JLabel cinema;

**private** JTextField tcinema;

**private** JLabel movie;

**private** JTextField tmovie;

**private** JLabel mclass;

**private** JRadioButton silver;

**private** JRadioButton gold;

**private** ButtonGroup mClass;

**private** JLabel mdate;

**private** JComboBox<?> date;

**private** JComboBox<?> month;

**private** JComboBox<?> year;

**private** JCheckBox term;

**private** JButton sub;

**private** JButton reset;

**private** JButton receipt;

**private** JTextArea tout;

**private** JLabel res;

**private** JTextArea resadd;

**private** String dates[] = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10","11", "12", "13", "14", "15","16", "17", "18", "19", "20","21", "22", "23", "24", "25","26", "27", "28", "29", "30","31"};

**private** String months[] = { "Jan", "feb", "Mar", "Apr","May", "Jun", "July", "Aug","Sup", "Oct", "Nov", "Dec" };

**private** String years[] = { "1995", "1996", "1997", "1998","1999", "2000", "2001", "2002","2003", "2004", "2005", "2006","2007", "2008", "2009", "2010","2011", "2012", "2013", "2014","2015", "2016", "2017", "2018", "2019", "2020" };

**public** SwingMovie() { setTitle("Movies"); setBounds(300, 90, 900, 600); setDefaultCloseOperation(***EXIT\_ON\_CLOSE***); setResizable(**false**);

c = getContentPane(); c.setLayout(**null**);

title = **new** JLabel("Movie Booking");

title.setFont(**new** Font("Arial", Font.***PLAIN***, 30));

title.setSize(300, 30);

title.setLocation(300, 30);

c.add(title);

city = **new** JLabel("City");

city.setFont(**new** Font("Arial", Font.***PLAIN***, 20));

city.setSize(100, 20);

city.setLocation(100, 100);

c.add(city);

tcity = **new** JTextField(); tcity.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); tcity.setSize(190, 20); tcity.setLocation(200, 100); c.add(tcity);

cinema = **new** JLabel("Cinema"); cinema.setFont(**new** Font("Arial", Font.***PLAIN***, 20)); cinema.setSize(100, 20); cinema.setLocation(100, 150); c.add(cinema);

tcinema = **new** JTextField(); tcinema.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); tcinema.setSize(150, 20); tcinema.setLocation(200, 150); c.add(tcinema);

movie = **new** JLabel("Movie"); movie.setFont(**new** Font("Arial", Font.***PLAIN***, 20)); movie.setSize(100, 20); movie.setLocation(100, 200); c.add(movie);

tmovie = **new** JTextField(); tmovie.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); tmovie.setSize(150, 20); tmovie.setLocation(200, 200); c.add(tmovie);

mclass = **new** JLabel("Class"); mclass.setFont(**new** Font("Arial", Font.***PLAIN***, 20)); mclass.setSize(100, 20); mclass.setLocation(100, 250); c.add(mclass);

silver = **new** JRadioButton("Silver"); silver.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); silver.setSelected(**true**); silver.setSize(75, 20); silver.setLocation(200, 250); c.add(silver);

gold = **new** JRadioButton("Gold"); gold.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); gold.setSelected(**false**); gold.setSize(80, 20); gold.setLocation(275, 250); c.add(gold);

mClass = **new** ButtonGroup(); mClass.add(silver); mClass.add(gold);

mdate = **new** JLabel("Date"); mdate.setFont(**new** Font("Arial", Font.***PLAIN***, 20)); mdate.setSize(100, 20); mdate.setLocation(100, 300); c.add(mdate);

date = **new** JComboBox<Object>(dates);

date.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); date.setSize(50, 20); date.setLocation(200, 300); c.add(date);

month = **new** JComboBox<Object>(months);

month.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); month.setSize(60, 20); month.setLocation(250, 300); c.add(month);

year = **new** JComboBox<Object>(years);

year.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); year.setSize(60, 20); year.setLocation(320, 300); c.add(year);

term = **new** JCheckBox("Accept Terms And Conditions."); term.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); term.setSize(250, 20); term.setLocation(150, 400); c.add(term);

sub = **new** JButton("Submit");

sub.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); sub.setSize(100, 20); sub.setBackground(Color.***green***); sub.setLocation(150, 450); sub.addActionListener(**this**); c.add(sub);

reset = **new** JButton("Reset");

reset.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); reset.setSize(100, 20); reset.setBackground(Color.***white***); reset.setLocation(270, 450); reset.addActionListener(**this**); c.add(reset);

receipt = **new** JButton("Generate receipt"); receipt.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); receipt.setSize(200, 20);

receipt.setBackground(Color.***orange***); receipt.setLocation(540, 450); receipt.addActionListener(**this**); c.add(receipt);

tout = **new** JTextArea();

tout.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); tout.setSize(300, 250); tout.setLocation(500, 100); tout.setLineWrap(**true**); tout.setEditable(**false**); c.add(tout);

res = **new** JLabel("");

res.setFont(**new** Font("Arial", Font.***PLAIN***, 20)); res.setSize(500, 25); res.setLocation(100, 500); c.add(res);

resadd = **new** JTextArea();

resadd.setFont(**new** Font("Arial", Font.***PLAIN***, 15)); resadd.setSize(200, 75); resadd.setLocation(580, 175); resadd.setLineWrap(**true**); c.add(resadd);

setVisible(**true**);

}

@Override

**public** **void** actionPerformed(ActionEvent e) { String receipt = ""; **if** (e.getSource() == sub) { **if** (term.isSelected()) { String data1;

String data = "City : " + tcity.getText() + "\n" + "Cinema : " + tcinema.getText() + "\n"+ "Movie : " +

tmovie.getText() + "\n";

**if** (silver.isSelected()) data1 = "Class : Silver" + "\n";

**else** data1 = "Class : Gold" + "\n"; String data2 = "Date : " + (String)date.getSelectedItem()

+ "/" + (String)month.getSelectedItem()

+ "/" + (String)year.getSelectedItem() + "\n";

receipt = data + data1 + data2; tout.setText(receipt); tout.setEditable(**false**);

res.setText("Registration Successfully..");

}

**else** { tout.setText(""); resadd.setText("");

res.setText("Please accept the" + " terms & conditions..");

}}

**else** **if** (e.getSource() == reset) { String def = ""; tcity.setText(def); tcinema.setText(def); tmovie.setText(def);

tout.setText(def); term.setSelected(**false**); date.setSelectedIndex(0); month.setSelectedIndex(0); year.setSelectedIndex(0); resadd.setText(def);

}

**else** **if** (e.getSource() == receipt) { FileWriter writer = **null**; **try** { writer = **new** FileWriter("movie-receipt.txt");

} **catch** (IOException ioException) { ioException.printStackTrace();

}

BufferedWriter buffer = **new** BufferedWriter(writer); **try** { buffer.write(receipt);

} **catch** (IOException ioException) { ioException.printStackTrace();

}

**try** { buffer.close();

} **catch** (IOException ioException) { ioException.printStackTrace();

}}}}

**class** Registration

{

**private** **static** SwingMovie *f*;

**public** **static** **void** main(String[] args) **throws** Exception

{

*setF*(**new** SwingMovie());

}

**public** **static** SwingMovie getF() {

**return** *f*;

}

**public** **static** **void** setF(SwingMovie f) {

Registration.*f* = f;

}

}

**Output:**

