Aim: Introduce java Applet, awt, swings.

What is AWT in Java?

The full form of AWT is Abstract Window Toolkit. It has no full version. 2.
It is an API used to develop window-based applications in Java.

What is Swing in Java?

Swing is a graphical user interface (GUI) and a part of Oracle's Java Foundation Classes that are used to design different applications.

No.	Java AWT	Java Swing
1)	AWT components are **platform-dependent**.	Java swing components are **platform-independent**.
2)	AWT components are **heavyweight**.	Swing components are **lightweight**.
3)	AWT **doesn't support pluggable look and feel**.	Swing **supports pluggable look and feel**.
4)	AWT provides **less components** than Swing.	Swing provides **more powerful components** such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
5)	AWT **doesn't follows MVC**(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.	Swing **follows MVC**.

Java Applet

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Advantage of Applet

There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

Plugin is required at client browser to execute applet.

Programs:

12: a) Develop an applet that displays a simple message in center of the screen.
b)Develop a simple calculator using Swings.

a) Develop an applet that displays a simple message in center of the screen.

```
/* Develop an applet that displays a simple message( information ) */
import java.awt.*;
import java.applet.*;
/*<applet code=weeka height=400 width=400>
</applet> */
class weeka extends Applet
{
 public static void init()
 {
  setForeground(Color.yellow);
 }
 public static void paint(Graphics g)
 {
  g.drawString(" Hello World ",80,40);
 }
}
```

b) Create a simple calculator using Java Swing.

12b).Java Swing is a GUI (graphical user Interface) widget toolkit for Java. Java Swing is a part of Oracle's Java foundation classes . Java Swing is an API for providing graphical user interface elements to Java Programs. Swing was created to provide more powerful and flexible components than Java AWT (Abstract Window Toolkit).

methods used :

1. **add(Component c)** : adds component to container.
2. **addActionListenerListener(ActionListener d)** : add actionListener for specified component
3. **setBackground(Color c)** : sets the background color of the specified container
4. **setSize(int a, int b)** : sets the size of container to specified dimensions.
5. **setText(String s)** : sets the text of the label to s.

6. **getText()** : returns the text of the label.

```
// Java program to create a simple calculator
// with basic +, -, /, * using java swing elements

import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
class calculator extends JFrame implements ActionListener {
	// create a frame
	static JFrame f;

	// create a textfield
	static JTextField l;

	// store operator and operands
	String s0, s1, s2;

	// default constructor
	calculator()
	{
		s0 = s1 = s2 = "";
	}

	// main function
	public static void main(String args[])
	{
		// create a frame
		f = new JFrame("calculator");

		try {
			// set look and feel

	UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
		}
		catch (Exception e) {
			System.err.println(e.getMessage());
		}

		// create a object of class
		calculator c = new calculator();

		// create a textfield
		l = new JTextField(16);

		// set the textfield to non editable
		l.setEditable(false);

		// create number buttons and some operators
		JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, bm, be, beq, beq1;

		// create number buttons
		b0 = new JButton("0");
```

```
b1 = new JButton("1");
b2 = new JButton("2");
b3 = new JButton("3");
b4 = new JButton("4");
b5 = new JButton("5");
b6 = new JButton("6");
b7 = new JButton("7");
b8 = new JButton("8");
b9 = new JButton("9");

// equals button
beq1 = new JButton("=");

// create operator buttons
ba = new JButton("+");
bs = new JButton("-");
bd = new JButton("/");
bm = new JButton("*");
beq = new JButton("C");

// create . button
be = new JButton(".");

// create a panel
JPanel p = new JPanel();

// add action listeners
bm.addActionListener(c);
bd.addActionListener(c);
bs.addActionListener(c);
ba.addActionListener(c);
b9.addActionListener(c);
b8.addActionListener(c);
b7.addActionListener(c);
b6.addActionListener(c);
b5.addActionListener(c);
b4.addActionListener(c);
b3.addActionListener(c);
b2.addActionListener(c);
b1.addActionListener(c);
b0.addActionListener(c);
be.addActionListener(c);
beq.addActionListener(c);
beq1.addActionListener(c);

// add elements to panel
p.add(l);
p.add(ba);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(bs);
p.add(b4);
p.add(b5);
```

```
		p.add(b6);
		p.add(bm);
		p.add(b7);
		p.add(b8);
		p.add(b9);
		p.add(bd);
		p.add(be);
		p.add(b0);
		p.add(beq);
		p.add(beq1);

		// set Background of panel
		p.setBackground(Color.blue);

		// add panel to frame
		f.add(p);

		f.setSize(200, 220);
		f.show();
}
public void actionPerformed(ActionEvent e)
{
		String s = e.getActionCommand();

		// if the value is a number
		if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
			// if operand is present then add to second no
			if (!s1.equals(""))
				s2 = s2 + s;
			else
				s0 = s0 + s;

			// set the value of text
			l.setText(s0 + s1 + s2);
		}
		else if (s.charAt(0) == 'C') {
			// clear the one letter
			s0 = s1 = s2 = "";

			// set the value of text
			l.setText(s0 + s1 + s2);
		}
		else if (s.charAt(0) == '=') {

			double te;

			// store the value in 1st
			if (s1.equals("+"))
				te = (Double.parseDouble(s0) + Double.parseDouble(s2));
			else if (s1.equals("-"))
				te = (Double.parseDouble(s0) - Double.parseDouble(s2));
			else if (s1.equals("/"))
				te = (Double.parseDouble(s0) / Double.parseDouble(s2));
			else
```

```
                    te = (Double.parseDouble(s0) * Double.parseDouble(s2));

                // set the value of text
                l.setText(s0 + s1 + s2 + "=" + te);

                // convert it to string
                s0 = Double.toString(te);

                s1 = s2 = "";
            }
            else {
                // if there was no operand
                if (s1.equals("") || s2.equals(""))
                    s1 = s;
                // else evaluate
                else {
                    double te;

                    // store the value in 1st
                    if (s1.equals("+"))
                        te = (Double.parseDouble(s0) +
Double.parseDouble(s2));
                    else if (s1.equals("-"))
                        te = (Double.parseDouble(s0) -
Double.parseDouble(s2));
                    else if (s1.equals("/"))
                        te = (Double.parseDouble(s0) /
Double.parseDouble(s2));
                    else
                        te = (Double.parseDouble(s0) *
Double.parseDouble(s2));

                    // convert it to string
                    s0 = Double.toString(te);

                    // place the operator
                    s1 = s;

                    // make the operand blank
                    s2 = "";
                }

                // set the value of text
                l.setText(s0 + s1 + s2);
            }
        }
}
```