Third Year Diploma Courses in Computer Science & Engineering, Computer Engineering, Computer Technology and Information Technology Branch.

Java Programming

As per MSBTE 'I' Scheme Syllabus
JPR-22412

Unit- V
Java Applets & Graphics Programming

Total Marks- 10

Contents:

5.1 Introduction to applets Applet, Applet life cycle (skeleton), Applet tag, Adding Applet To HTML file, passing parameter to applet, embedding <applet>tags in java code, adding controls to applets.

5.2 Graphics Programming Graphics classes, lines, rectangles, ellipse, circle, arcs, polygons, color & fonts, setColor(), getColor(), setForeGround(), setBackGround(), font class, variable defined by font class: name, pointSize, size, style, font methods: getFamily(), getFont(), getFontname(), getSize(), getStyle(), getAllFonts() & getavailablefontfamilyname() of the graphics environment class.

Prof. Gunwant V. Mankar
B.E(IT), M.Tech(CSE), AMIE, MIAEng, MSCI
HOD** in CSE Dept**.
(BTC- School of Diploma in Engineering, Ballarpur)
e-mail:- info@gunwantmankar.com
Website:- www.gunwantmankar.com

Unit- V:- Java Applets & Graphics Programming

5.1. Introduction to applets

Applet Basics-

Basically, an applet is dynamic and interactive java program that inside the web page or applets are small java programs that are primarily used in internet computing. The java application programs run on command prompt using java interpreter whereas the java applets can be transported over the internet form one computer to another and run using the appletviewer or any web browser that supports java.

An applet is like application program which can perform arithmetic operations, display graphics, play sounds accept user input, create animation and play interactive games. To run an applet, it must be included in HTML tags for web page. Web browser is a program to view web page.

Every applet is implemented by creating sub class of Applet class. Following diagram shows the inheritance hierarchy of Applet class.

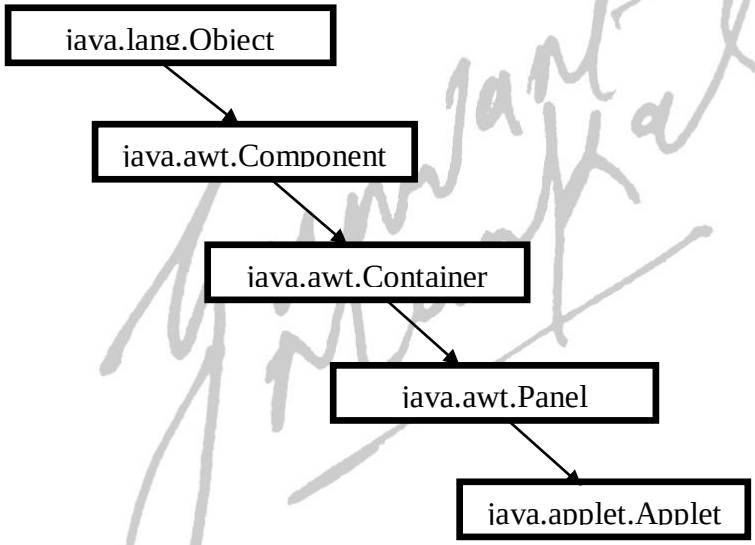
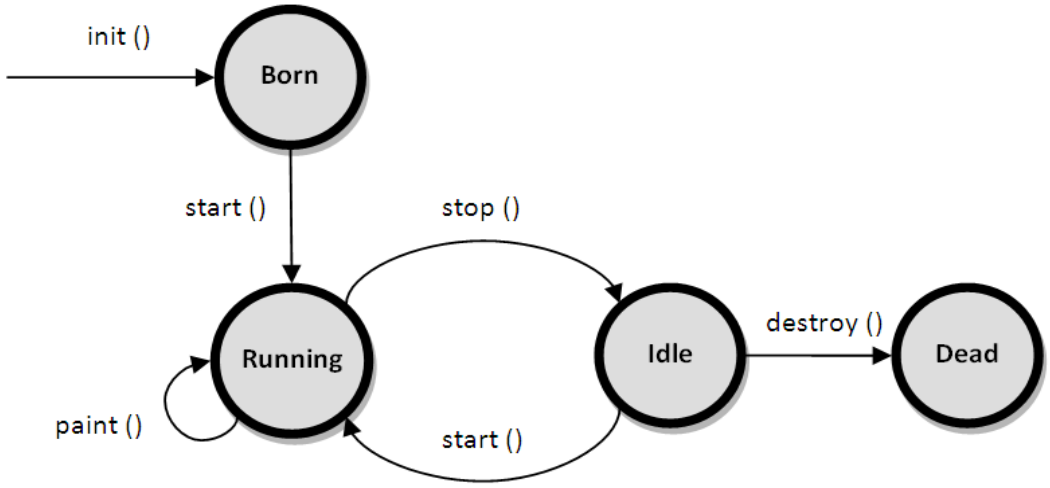


Fig. Chain of classes inherited by Applet class in java

5.1.1 Differentiate between applet and application (4 points). [W-14, S-15, W-15]

Applet	Application
Applet does not use main() method for initiating execution of code	Application use main() method for initiating execution of code
Applet cannot run independently	Application can run independently
Applet cannot read from or write to files in local computer	Application can read from or write to files in local computer
Applet cannot communicate with other servers on network	Application can communicate with other servers on network
Applet cannot run any program from local computer.	Application can run any program from local computer.
Applet are restricted from using libraries from other language such as C or C++	Application are not restricted from using libraries from other language

5.1.2 Applet life Cycle [W-14, S-15]



Applets are small applications that are accessed on an Internet server, transported over the Internet, automatically installed, and run as part of a web document.

The applet states include:

- **Born or initialization state**
- **Running state**
- **Idle state**
- **Dead or destroyed state**

a) Born or initialization state [S-15]

Applet enters the initialization state when it is first loaded. This is done by calling the init() method of Applet class. At this stage the following can be done:

- Create objects needed by the applet
- Set up initial values
- Load images or fonts
- Set up colors

Initialization happens only once in the life time of an applet.

```
public void init()
        {
        //implementation
        }
```

b) Running state: [S-15]

Applet enters the running state when the system calls the start() method of Applet class. This occurs automatically after the applet is initialized. start() can also be called if the applet is already in idle state. start() may be called more than once. start() method may be overridden to create a thread to control the applet.

```
public void start()
        {
        //implementation
        }
```

c) Idle or stopped state:

An applet becomes idle when it is stopped from running. Stopping occurs automatically when the user leaves the page containing the currently running applet. stop() method may be overridden to terminate the thread used to run the applet.

```
public void stop()
        {
        //implementation
        }
```

d) Dead state:

An applet is dead when it is removed from memory. This occurs automatically by invoking the destroy method when we quit the browser. Destroying stage occurs only once in the lifetime of an applet. destroy() method may be overridden to clean up resources like threads.

```
public void destroy()
        {
        //implementation
        }
```

e) Display state: [S-15]

Applet is in the display state when it has to perform some output operations on the screen. This happens after the applet enters the running state. paint() method is called for this. If anything is to be displayed the paint() method is to be overridden.

```
public void paint(Graphics g)
        {
        //implementation
        }
```

5.1.3 Applet Tag & Attributes [W-15, S-16]

APPLET Tag:

The APPLET tag is used to start an applet from both an HTML document and from an applet viewer.

The syntax for the standard APPLET tag:

<APPLET

[CODEBASE = *codebaseURL*]
CODE = *appletFile*
[ALT = *alternateText*]
[NAME = *appletInstanceName*]
WIDTH = *pixels* HEIGHT = *pixels*
[ALIGN = *alignment*]
[VSPACE = *pixels*] [HSPACE = *pixels*]>
[< PARAM NAME = *AttributeName1* VALUE = *AttributeValue*>]
[<PARAM NAME = *AttributeName2* VALUE = *AttributeValue*>]
. . .

</APPLET>

CODEBASE is an optional attribute that specifies the base URL of the applet code or the directory that will be searched for the applet‟s executable class file.

CODE is a required attribute that give the name of the file containing your applet‟s compiled class file which will be run by web browser or appletviewer.

ALT: Alternate Text. The ALT tag is an optional attribute used to specify a short text message that should be displayed if the browser cannot run java applets.

NAME is an optional attribute used to specifies a name for the applet instance.

WIDTH AND HEIGHT are required attributes that give the size(in pixels) of the applet display area.

ALIGN is an optional attribute that specifies the alignment of the applet. The possible value is: LEFT, RIGHT, TOP, BOTTOM, MIDDLE, BASELINE, TEXTTOP, ABSMIDDLE, and ABSBOTTOM.

VSPACE AND HSPACE attributes are optional, VSPACE specifies the space, in pixels, about and below the applet. HSPACE VSPACE specifies the space, in pixels, on each side of the applet

PARAM NAME AND VALUE: The PARAM tag allows you to specifies applet-specific arguments in an HTML page applets access there attributes with the get Parameter()method.

Q. Explain <PARAM> tag of applet with suitable example. [S-15]

To pass parameters to an applet <PARAM… > tag is used. Each <PARAM…> tag has a name attribute and a value attribute. Inside the applet code, the applet can refer to that parameter by name to find its value.

The syntax of <PARAM…> tag is as follows

<PARAM NAME = name1 VALUE = value1>

To set up and handle parameters, two things must be done.

1. Include appropriate <PARAM..> tags in the HTML document.
2. Provide code in the applet to parse these parameters.

Parameters are passed on an applet when it is loaded. Generally init() method in the applet is used to get hold of the parameters defined in the <PARAM…> tag.

The getParameter() method, which takes one string argument representing the name of the parameter and returns a string containing the value of that parameter.

Example

```
import java.awt.*;
import java.applet.*;
```

```
public class hellouser extends Applet
{
String str;
public void init()
{
str = getParameter("username");
str = "Hello "+ str;
}
public void paint(Graphics g)
{
g.drawString(str,10,100);
}
}

<HTML>
<Applet code = “hellouser.class” width = 400 height = 400>
<PARAM NAME = "username" VALUE = abc>
</Applet>
</HTML>
```

Q. How can parameter be passed to an applet? Write an applet to accept user name in the form of parameter and print ‘Hello<username>’. [W-15]

- **Passing Parameters to Applet**

User defined parameters can be supplied to an applet using <PARAM…..> tags. PARAM tag names a parameter the Java applet needs to run, and provides a value for that parameter.

PARAM tag can be used to allow the page designer to specify different colors, fonts, URLs or other data to be used by the applet.

- **To set up and handle parameters, two things must be done.**

1. Include appropriate <PARAM..>tags in the HTML document.

The Applet tag in HTML document allows passing the arguments using param tag.

The syntax of <PARAM…> tag

<Applet code=”AppletDemo” height=300 width=300>
<PARAM NAME = name1 VALUE = value1>
</Applet>

NAME: attribute name
VALUE: value of attribute named by corresponding PARAM NAME.

2. Provide code in the applet to parse these parameters.

The Applet access their attributes using the getParameter method.

The syntax is : **String getParameter(String name);**

- **Program for an applet to accept user name in the form of parameter and print 'Hello<username>' [W-15]**

```
import java.awt.*;
import java.applet.*;

public class hellouser extends Applet
{
String str;
public void init()
{
str = getParameter("username");
str = "Hello "+ str;
}
public void paint(Graphics g)
{
g.drawString(str,10,100);
}
}
```

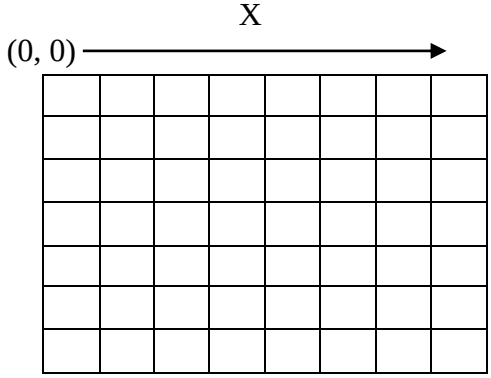
```
<HTML>
<Applet code = "hellouser.class" width = 400 height = 400>
<PARAM NAME = "username" VALUE = abc>
</Applet>
</HTML>
```

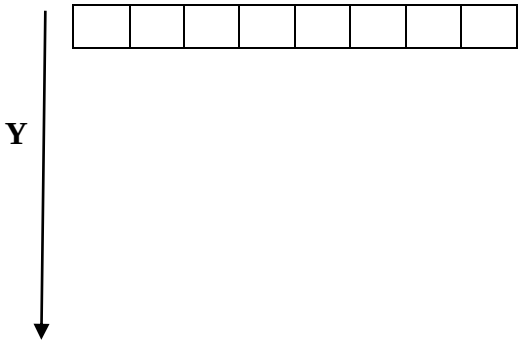
5.2. Graphics Programming

Graphics can be drawn with the help of java. java applets are written to draw lines, figures of different shapes, images and text in different styles even with the colours in display.

Every applet has its own area of the screen known as canvas, where it creates the display in the area specified the size of applet's space is decided by the attributes of <APPLET...> tag.

A java applet draws graphical image inside its space using the coordinate system shown in following fig., which shows java's coordinate system has the origin (0, 0) in the upper-left corner, positive x values are to be right, and positive y values are to the bottom. The values of coordinates x and y are in pixels.





Q. Write a simple applet which display message 'Welcome to Java'. [W-15]

Program:

```
import java. applet.*;
import java.awt.*;

public class Welcome extends Applet
   {
   public void paint( Graphics g)
            {
             g.drawString("Welcome to java",25,50);
            }
   }

/*<applet code= WelcomeJava width= 300 height=300>
</applet>*/
```

- **Step to run an Applet**

1. Write a java applet code and save it with as a class name declared in a program by extension as a .java.
   e.g. from above java code file we can save as a **Welcome.java**

2. Compile the java file in command prompt jdk as shown below
   C:\java\jdk1.7.0\bin> javac Welcome.java

3. After successfully compiling java file, it will create the .class file, e.g Welcome.class. then we have to write applet code to add this class into applet.

4. Applet code

   ```
   <html>
   <Applet code= " Welcome.class"  width= 500  height=500>
   </applet>
   </html>
   ```

5. Save this file with Welcome.html in 'bin' library folder.

6. Now write the following steps in command prompt jdk.

   C:\java\jdk1.7.0\bin> appletviewer Welcome.java

C:\java\jdk1.7.0\bin> appletviewer Welcome.html
(Shows output in applet viewer)
OR
C:\java\jdk1.7.0\bin> Welcome.html
(Shows output in internet browser)

5.2.1. Graphics Class

The Graphics class of java includes methods for drawing different types of shapes, from simple lines to polygons to text in a variety of fonts.

The paint( ) method and a Graphics object is used to display text. To draw shapes, drawing methods in Graphics class is used which arguments representing end points, corners, or starting locations of a shape as a values in the applet's coordinate system.

Method	Description
clearRect( )	Erases a rectangular area of the canvas
copyArea( )	Copies a rectangular area of the canvas to another area
drawArc( )	Draws a hollow arc.
drawLine( )	Draws a straight line
drawOval( )	Draws a hollow oval
drawPolygon( )	Draws a hollow polygon
drawRect( )	Draws a hollow rectangle
drawRoundRect( )	Draws a hollow rectangle with rounded corners.
drawstring( )	Displays a text string
fillArc( )	Draws a filled arc
fillOval( )	Draws a filled arc
fillPolygon( )	Draws a filled polygon
fillRect( )	Draws a filled rectangle
fillRoundRect( )	Draws filled rectangle with rounded corners
getColor( )	Retrieves the current drawing color
getFont( )	Retrieves the currently used font
getFontMetrics( )	Retrieves information about the current font.
setColor( )	Sets the drawing color
setFont( )	Seta fonts.

5.2.2. drawString( ) [S-15]

Displaying String:
drawString() method is used to display the string in an applet window

Syntax:

void drawString(String message, int x, int y);
where message is the string to be displayed beginning at x, y

Example:

g.drawString("WELCOME", 10, 10);

5.2.3. Lines and Rectangle.

5.2.3.1. drawLine( )

The drawLine ( ) method is used to draw line which takes two pair of coordinates (x1,y1) and (x2, y2) as arguments and draws a line between them. The graphics object g is passed to paint( ) method.

The syntax is

g.drawLine(x1,y1,x2,y2);

e.g. g.drawLine(20,20,80,80);

5.2.3.2. drawRect( ) [W-14, S-15,W-15, S-16]

The drawRect() method display an outlined rectangle

Syntax: void drawRect(int top, int left, int width, int height)

This method takes four arguments, the first two represents the x and y co-ordinates of the top left corner of the rectangle and the remaining two represent the width and height of rectangle.

Example: g.drawRect(10,10,60,50);

Q. Design an Applet program which displays a rectangle filled with red color and message as "Hello Third year Students" in blue color. [S-16]

Program-

```
import  java.awt.*;
import  java.applet.*;

public class DrawRectangle extends Applet
        {
        public void paint(Graphics g)
                {
                        g.setColor(Color.red);
                        g.fillRect(10,60,40,30);
```

```
                    g.setColor(Color.blue);
                    g.drawString("Hello Third year Students",70,100);
                }
        }

/* <applet code="DrawRectangle.class" width="350" height="300"> </applet> */
```

5.2.4. Circle and Ellipse

5.2.4.1. drawOval( ) [W-14, S-15, W-15, S-16]

To draw an Ellipses or circles used drawOval() method can be used.

Syntax: void drawOval(int top, int left, int width, int height)

The ellipse is drawn within a bounding rectangle whose upper-left corner is specified by top and left and whose width and height are specified by width and height to draw a circle or filled circle, specify the same width and height the following program draws several ellipses and circle.

Example: g.drawOval(10,10,50,50);

5.2.4.2. fillOval ( ) [W-14]

Draws an oval within a bounding rectangle whose upper left corner is specified by top, left. Width and height of the oval are specified by width and height.

Syntax- void fillOval(int top, int left, int width, int height):

Example g.fillOval(10,10,50,50);

Q. Write a simple applet program which display three concentric circle. [S-16]

Program-

```
import java.awt.*;
import java.applet.*;

public class CircleDemo extends Applet
        {
            public void paint (Graphics g)
                    {

                    g.drawOval(100,100,190,190);
                    g.drawOval(115,115,160,160);
                    g.drawOval(130,130,130,130);
                    }
        }
```

```
/*<applet code="CircleDemo.class" height=300 width=200>
 </applet>*/
```

(OR)

HTML Source:

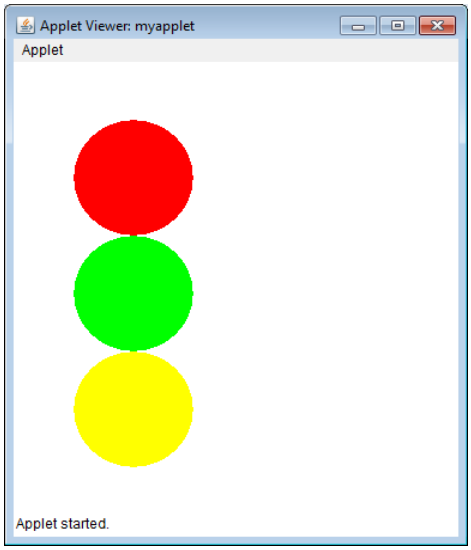
```
<html> <applet code="CircleDemo.class" height=300 width=200>
</applet>
</html>
```

Q. Write a program to design an applet to display three circles filled with three different colors on screen. [W-14, W-15]

Program-

```
import  java.awt.*;
import  java.applet.*;

public class myapplet extends Applet
        {
                public void paint(Graphics g)
                {
                g.setColor(Color.red);
                g.fillOval(50,50,100,100);

                g.setColor(Color.green);
                g.fillOval(50,150,100,100);

                g.setColor(Color.yellow);
                g.fillOval(50,250,100,100);

                }
        }

/*<applet code=myapplet width= 300 height=300>
</applet>*/
```

Output



5.2.5. Drawing Arcs

5.2.5.1. drawArc( ) [S-15, W-15]

It is used to draw arc

Syntax:

void drawArc(int x, int y, int w, int h, int start_angle, int sweep_angle);

where x, y starting point, w& h are width and height of arc, and start_angle is starting angle of arc sweep_angle is degree around the arc

Example:

```
g.drawArc(10, 10, 30, 40, 40, 90);
```

5.2.6. Drawing polygons

5.2.6.1. drawPolygon( ) [W-14, W-15]

drawPolygon() method is used to draw arbitrarily shaped figures.

Syntax- void drawPolygon(int[ ] xPoints, int[ ] yPoints, int numPoints):

The polygon‟s end points are specified by the co-ordinates pairs contained within the x and y arrays. The number of points define by x and y is specified by numPoints.

Example-

```
int x[ ] = {10, 170, 80};
int y[ ] = {20, 40, 140};
int n = 3;

g.drawPolygon(x, y, n);
```

Q. Write the syntax and example for each of following graphics methods:

1) drawPoly ( ) 2) drawRect ( ) 3) drawOval ( ) 4) fillOval ( )

For syntax refer above 5.2.3.2 and all.....

Example for including all methods in a one program

```
import java.applet.*;
import java.awt.*;

public class DrawGraphics extends Applet
{
        public void paint(Graphics g)
        {
        int x[ ] = {10, 170, 80};
        int y[ ] = {20, 40, 140};
        int n = 3;
        g.drawPolygon(x, y, n);
        g.drawRect(10, 150,100, 80);
        g.drawOval(10, 250, 100, 80);
        g.fillOval(10, 350, 100, 80);
        }
}

/*
<applet code = DrawGraphics.class height = 500 width = 400>
</applet>*/
```

5.2.7. Setting color of an Applet

- Background and foreground color of an applet can be set by using followings methods

  void setBackground(Color.newColor)

  void setForeground (Color.newColor)

  where newColor specifies the new color. The class color defines the constant for specific color listed below.

Color.black	Color.white	Color.pink	Color.yellow
Color.lightGray	Color.gray	Color.darkGray	Color.red
Color.green	Color.magenda	Color.orange	Color.cyan

Example

```
setBackground(Color.red);

setForeground (Color.yellow);
```

- The following methods are used to retrieve the current background and foreground color.

  Color getBackground( )

Color getForeground( )

5.2.8. Font class

A font determines look of the text when it is painted. Font is used while painting text on a graphics context & is a property of AWT component.

The Font class defines these variables:

Variable	Meaning
String name	Name of the font
float pointSize	Size of the font in points
int size	Size of the font in point
int style	Font style

5.2.8.1. Use of font class [W-14, S-15]

- The Font class states fonts, which are used to render text in a visible way.
- It is used to set or retrieve the screen font.

Syntax to create an object of Font class. [W-14]

To select a new font, you must first construct a Font object that describes that font. Font constructor has this general form:

Font(String fontName, int fontStyle, int pointSize)

fontName specifies the name of the desired font. The name can be specified using either the logical or face name.

All Java environments will support the following fonts:

Dialog, DialogInput, Sans Serif, Serif, Monospaced, and Symbol. Dialog is the font used by once system‟s dialog boxes.

Dialog is also the default if you don‟t explicitly set a font. You can also use any other fonts supported by particular environment, but be careful—these other fonts may not be universally available.

The style of the font is specified by fontStyle. It may consist of one or more of these three constants:

Font.PLAIN, Font.BOLD, and Font.ITALIC. To combine styles, OR them together.

For example,

Font.BOLD | Font.ITALIC specifies a bold, italics style.

The size, in points, of the font is specified by pointSize.

To use a font that you have created, you must select it using setFont( ), which is defined by Component.

It has this general form:

void setFont(Font fontObj)

Here, fontObj is the object that contains the desired font

5.2.8.2. Methods of font class

Q. Describe any three methods of font class with their syntax and example of each. [W-14, S-15]

Sr. No	Methods	Description
1	static Font decode(String *str*)	Returns a font given its name.
2	boolean equals(Object *FontObj*) :	Returns **true** if the invoking object contains the same font as that specified by *FontObj*.Otherwise, it returns **false**.
3	String toString( )	Returns the string equivalent of the invoking font.
4	String getFamily( )	Returns the name of the font family to which the invoking font belongs.
5	static Font getFont(String *property*)	Returns the font associated with the system property specified by *property*. **null** is returned if *property* does not exist.
6	static Font getFont(String *property*,Font *defaultFont*)	Returns the font associated with the System property specified by *property*. The font specified by *defaultFont* is returned if *property* does not exist.
7	String getFontName( )	Returns the face name of the invoking font.
8	String getName( )	Returns the logical name of the invoking font.
9	int getSize( )	Returns the size, in points, of the invoking font.
10	int getStyle( )	Returns the style values of the invoking font.
11	int hashCode( )	Returns the hash code associated with the invoking object.
12	boolean isBold( )	Returns **true** if the font includes the **BOLD** style value. Otherwise, **false** is returned.
13	boolean isItalic( )	Returns **true** if the font includes the **ITALIC** style value. Otherwise, **false** is returned.
14	boolean isPlain( )	Returns **true** if the font includes the **PLAIN** style value. Otherwise, **false** is returned.

Example:-

//program using equals method

```
import java.awt.*;
import java.applet.*;

public class ss extends Applet
        {
        public void paint(Graphics g)
                {
                Font a = new Font ("TimesRoman", Font.PLAIN, 10);
                Font b = new Font ("TimesRoman", Font.PLAIN, 10);
```

```
                // displays true since the objects have equivalent settings
                g.drawString(""+a.equals(b),30,60);
                }
        }

/*<applet code=”ss.class” height=200 width=200>
</applet>*/
```

// program using getFontName,getFamily(),getSize(),getStyle(),.getName()

```
import java.awt.*;
import java.applet.*;

public class font1 extends Applet
        {
        Font f, f1;
        String s, msg;
        String  fname;
        String  ffamily;
        int size;
        int style;
        public void init()
                {
                f= new Font("times new roman",Font.ITALIC,20);
                setFont(f);
                msg="is interesting";
                s="java programming";
                fname=f.getFontName();
                ffamily=f.getFamily();
                size=f.getSize();
                style=f.getStyle();
                String f1=f.getName();
                }
        public void paint(Graphics g)
                {
                g.drawString("font name"+fname,60,44);
                g.drawString("font family"+ffamily,60,77);
                g.drawString("font size "+size,60,99);
                g.drawString("fontstyle "+style,60,150);
                g.drawString("fontname "+f1,60,190);
                }
        }

/*<applet code=font1.class height=300 width=300>
</applet>*/
```

Q. Write method to set font of a text and describe its parameters. [S-16]

The AWT supports multiple type fonts emerged from the domain of traditional type setting to become an important part of computer-generated documents and

displays. The AWT provides flexibility by abstracting font-manipulation operations and allowing for dynamic selection of fonts.

Fonts have a family name, a logical font name, and a face name. The family name is the general name of the font, such as Courier. The logical name specifies a category of font, such as Monospaced. The face name specifies a specific font, such as Courier Italic To select a new font, you must first construct a Font object that describes that font.

One Font constructor has this general form:

Font(String fontName, intfontStyle, intpointSize)

To use a font that you have created, you must select it using setFont( ), which is defined by Component.

It has this general form:

void setFont(Font fontObj)

Example

```
import java.applet.*;
import  java.awt.*;
import  java.awt.event.*;
public class SampleFonts extends Applet
        {
        int next = 0;
        Font  f;
        String msg;

        public void init( )
                {
                f = new Font("Dialog", Font.PLAIN, 12);
                msg = "Dialog";
                setFont(f);
        public void paint(Graphics g)
                {
                g.drawString(msg, 4, 20);
                }
        }
```

Q. State purpose of get Available Font Family Name ( ) method of graphics environment class.

Purpose of getAvailableFontFamilyName() method:

It returns an array of String containing the names of all font families in this Graphics Environment localized for the specified locale

Syntax:

public abstract String[ ] getAvailableFontFamilyNames(Locale 1)

Parameters:

l - a Locale object that represents a particular geographical, political, or cultural region. Specifying null is equivalent to specifying Locale.getDefault().

Or

String[ ] getAvailableFontFamilyNames( )

It will return an array of strings that contains the names of the available font families

Important Questions:-

4 Marks Questions:-

1) Write syntax and example of 1) drawString ( ) 2) drawRect ( ) ; 3) drawOval ( ) 4) drawArc ( ).
2) Describe following states of applet life cycle : a) Initialization state. b) Running state. c) Display state
3) State the use of font class. Describe any three methods of font class with their syntax and example of each.
4) Differentiate applet and application with any four points.
5) State syntax and explain it with parameters for : i)drawRect ( ) ii) drawOral ( )
6) Design an Applet program which displays a rectangle filled with red color and message as "Hello Third year Students" in blue color.
7) Describe applet life cycle with suitable diagram.
8) Differentiate between applet and application (any 4 points).
9) Write a program to design an applet to display three circles filled with three different colors on screen.
10) Explain all attributes available in < applet > tag.

6 & 8 Marks Questions:-

1) Explain <PARAM> Tag of applet with suitable example.
2) State the use of font class. Describe any three methods of font class with their syntax and example of each.
3) Write a simple applet program which display three concentric circle.
4) Write method to set font of a text and describe its parameters.
5) Explain <applet> tag with its major attributes only. State purpose of get Available Font Family Name ( ) method of graphics environment class.
6) Design an applet which displays three circles one below the other and fill them red, green and yellow color respectively.
7) Write the syntax and example for each of following graphics methods : 1) drawPoly ( ) 2) drawRect ( ) 3) drawOval ( ) 4) fillOval ( )
8) State the use of Font class. Write syntax to create an object of Font class.
9) Describe any 3 methods of Font class with their syntax and example of each.
10) Write syntax and example of following Graphics class methods : (i) drawOval( ) (ii) drawPolygon( ) (iii) drawArc( ) (iv) drawRect( )
11) Differentiate between applet and application and also write a simple applet which display message 'Welcome to Java'.
12) How can parameters be passed to an applet ? Write an applet to accept user name in the form of parameter and print 'Hello < username >'.