iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** IV **Month of publication:** April 2020

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

To make Simple Dynamic and Entertaining Web Pages in JAVA using Applet

Vasimraj Siraj Tamboli[1], Ravindra Sundarlal Kakade[2],
[1]*lecturer in Computer Department, Pad. Dr. Vitthalrao Vilkhe patil Polytechnic Loni*

Abstract: An applet is a small Java program that can be used for display the graphics, animation and text. Applet code is embedded into a web page using HTML. It is independent so runs inside the web browser. Applets are used to make the dynamic and entertaining web pages.

I. DEFINITION OF APPLET

1) Applet is small java programs that used in Internet computing, and any web browser that supports java.
2) It can perform arithmetic operations, display graphic, play sounds , accept user input, create animation, and play interactive games

A. Difference between Applet and Application

Table 1. Difference between Applet and Application

Applet	Application
1) Applet do not use main () method for initialization and execution of the code.	1) Application use main() method for initialization and execution of the code.
2)Applet cannot run independently	2) Application can run independently.
3) It requires web browser to display output.	3) It does not require web browser.
4)Applet required HTML code	4) Application does not require HTML code

B. The Basic Applet Life Cycle

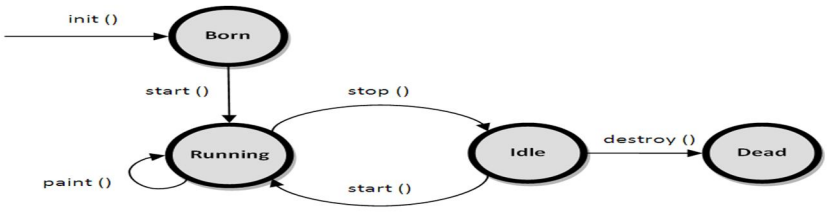


Fig.1 Applet Life Cycle

1) *Born or Initialization State*
a) Applet enters the initialization state when it is first loaded
b) This is achieved by calling the init() method of Applet class
c) The Applet is born. We may do the following if required
i) Create object needed by the Applet
ii) Setup initial value
iii) Load images or fonts
iv) Setup colours of Applet
d) The initialization occurs only once in the Applet Life cycle

```
public void init()
{
.....
 .....
}
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

2) Running State

a) Applet enters the running state when the system calls the start() method of Applet class.

b) This occurs automatically after the Applet is initialized

```
public void start()
{
.....
.....(action)
}
```

3) Idle or stopped State

a) An Applet becomes idle when it stopped from running

b) Stopping occurs automatically when we leave the page containing the currently running applet

c) We can also do so calling the stop () method

```
public void stop()
{
......
......
}
```

4) Dead State

a) An Applet is said to be dead when it is removed from memory

b) This occurs automatically by invoking the destroy () method. When we quit the browser

```
public void destroy ()
{
.....
....
(Action)
}
```

5) Display States

a) Applet moves the display state whenever it has to perform some output operations on the screen.

b) This happens immediately after the Applet enters into the running States

c) The paint() method is called to accomplishments this task

d) Almost every Applet will have a paint() method

```
public void paint(Graphics g)
{
.......
.......
}
```

C. Applet Tag

1) We have included a pair of <applet> and <\applet>tag.

2) The <applet> tag supplies the name of the applet to be loaded and tells the browser how much space the Applet required.

3) It contains certain attributes that must specify.

```
<Applet
    Code= “Appletname.class”      // Name of Applet
    Width= 400                    // width of Applet (in pixel)
    Height= 400                   // Height of Applet (in pixel)
</Applet>
```

D. Graphics Programing

1) Important feature of java is its ability to draw graphics.

2) We can draw java applets that draw lines, figure of different shapes , images, and text. In different fonts and style.

3) We can also incorporate different colours in display.

4) Every Applet has its own area of the screen known as canvas, where it creates it's display.
5) Java's co-ordinate system has the origin (0,0) in the upper left corner.
6) Positive X values are to the right and positive y values are to the bottom.
7) The values of coordinates X and y are in pixels.

E. Graphics Class
1) Graphics class includes methods for drawing many different types of shapes , from simple lines to polygons to text in a variety of fonts.
2) To draw a shape on the screen , we may call one of the methods available in the graphics class .
3) Graphics class contain most commonly used drawings methods.
4) All drawing methods have argument representing end points , corners, starting location of a shapes as values in the Applet co-ordinate systems .

II. LINES AND RECTANGLES

A. Linea Drawing
1) The simplest shape we can draw with the graphics class is a line.
2) The drawLine() method takes two pair of coordinates (x1,y1) and (X2,y2) as argument and draws a line between them.
a) *Syntax:* drawLine(x1,y1,X2,y2);
b) *Example:* drawLine (10,60,40,30);

B. Rectangle Drawing
1) We can draw a rectangle using the drawRect() method.
2) This method takes four arguments.
3) The first two represent the X and Y coordinates of the top left corner of the remaining two represent the width and height of the rectangle
a) *Syntax :* drawRect(X ,Y, W, H);
b) *Example:* drawRect(10,60,40,30);
Note: That here first two coordinate are starting coordinates third coordinates represent width of rectangle and fourth coordinates represent height of rectangle.

C. Filling Rectangle
1) We can draw a solid box by using the method fillRect().
2) This also takes four parameters.
3) First two are starting points, third is width and fourth coordinates represent height.
a) *Syntax:* fillRect(X ,Y, W, H);
b) *Example:* fillRect(10,60,40,30);

D. Drawing Rounded Rectangle:-
1) We can also draw Rounded rectangles.
2) This is also a rectangle with Rounded edges.
3) To draw Rounded Rectangle using the methods drawRoundRect() .
4) These method is similar to drawRect() except that they take two extra arguments representing the width and height of angle of corners.
5) These extra Parameters indicates how much of corners will be rounde ***drawRoundRect(10,100,80,50,10,10);***

E. Filling Rounded Rectangle
1) We can also fill rounded rectangle.
2) To fill rounded Rectangle using fillRoundRect() method.
3) These methods is similar to fillRect() except two extra arguments **fillRoundRect(20,110,60,30,5,5);**
Program for draw line , rectangle & roundedRect and fill it.

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

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

```
public class LineRect extends Applet
  {
public void paint(Graphics g)
  {
 g.drawLine(10,10,50,50);
 g.drawRect(10,60,40,130);
 g.fillRect(60,10,30,80);
 g.drawRoundRect(10,100,80,50,10,10);
 g.fillRoundRect(20,110,60,30,5,5);
 g.drawLine(100,10,230,140);
 g.drawLine(100,140,230,10);
 }
 }/*<Applet code=LineRect.class Width=400 Height=400></Applet>*/
```

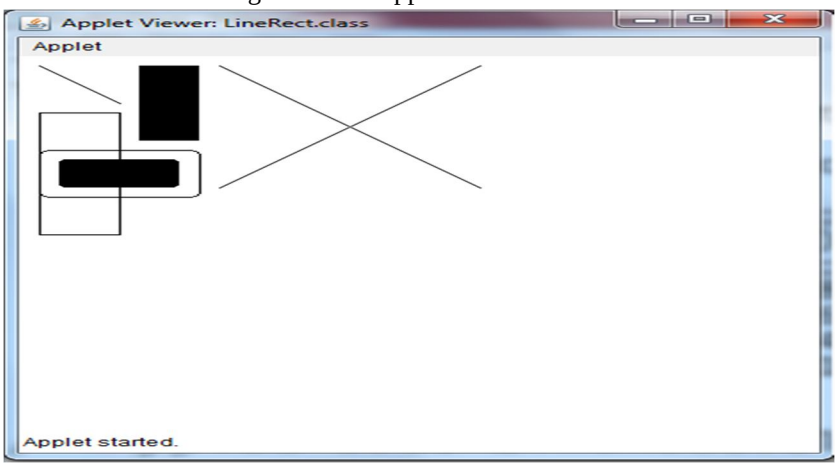


Fig.2 Program for draw line , rectangle & roundedRect and fill it

F. Circle and Ellipse

1) The graphics class does not have any Method for circle or ellipse
2) The drawOval() method can be used to draw a circle or an Ellipse
3) Ovals are just like rectangle with overly rounded corners
4) drawOval() methods takes four arguments. First two represent the top left corners of the imaginary rectangle and other two represent the width and height of the oval drawOval(20,20,200,120);
5) Like rectangle methods, the drawOval() method Draw outline of an oval, and the fillOval() method draws a solid Oval. fillOval(20,20,200,120);

G. Drawing Arcs

1) An arc is a part of an oval
2) We can think of an Oval as a series of arcs that are connected together in an orderly manner
3) The drawArc() method designed to draw arcs takes six arguments
4) The first four are the same as the arguments for drawOval () methods and last two represent the starting angle of the arc and the numbers of degree (Sweep Angle) around the arc drawArc(60,125,40,180,180);

H. Fill Arc

1) To fillArc() method is use to display solid Arc fillArc(60,125,80,40,180,180)

Program for drawing human face

```
import java.awt.*;
import java.applet.*;
public class Face1 extends Applet
```

```
    {
  public void paint (Graphics g)
    {
g.drawOval(40,40,120,150);
g.drawOval(57,75,30,20);
g.drawOval(110,75,30,20);
g.fillOval(68,81,10,10);
g.fillOval(121,81,10,10);
g.drawOval(85,100,30,30);
g.fillArc(60,125,80,40,180,180);
g.drawOval(25,92,15,30);
g.drawOval(160,92,15,30);
g.drawOval(160,92,15,30);
}
}
/*<Applet code =Face1.class Width= 400 Height= 400> </Applet>*/
```

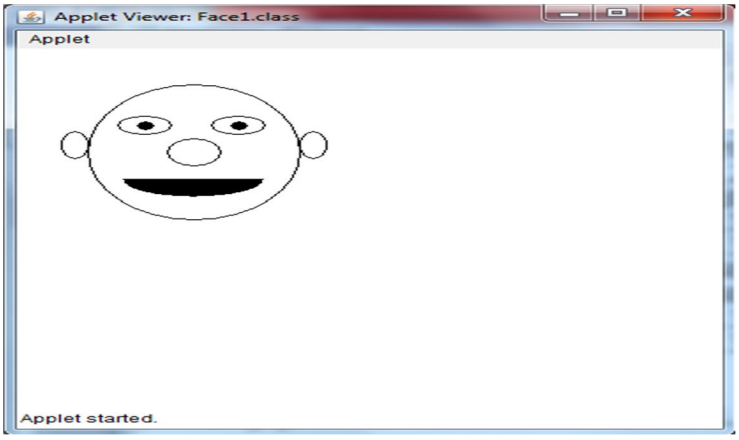


Fig.3 Output of program for drawing human face

I. Drawing Polygon

1) Polygons are shapes with many Sides.
2) A Polygons may be considered a set of lines connected together
3) The end of first line is the beginning of the second line , the end of second is beginning of the third and so on
4) We can draw a polygon within sides using the drawLine() method n times in succession.

Example draws a polygon of three sides using drawLine method

```
        import java.awt.*;
        import java.applet.*;
        public class Poly extends Applet
        {
        public void paint (Graphics g)
        {
        g.drawLine(10,20,170,40);
        g.drawLine(170,40,80,140);
        g.drawLine(80,140,10,20);
        }
        }
/*<Applet code=Poly.class Width=400, Height=400></Applet>*/
```

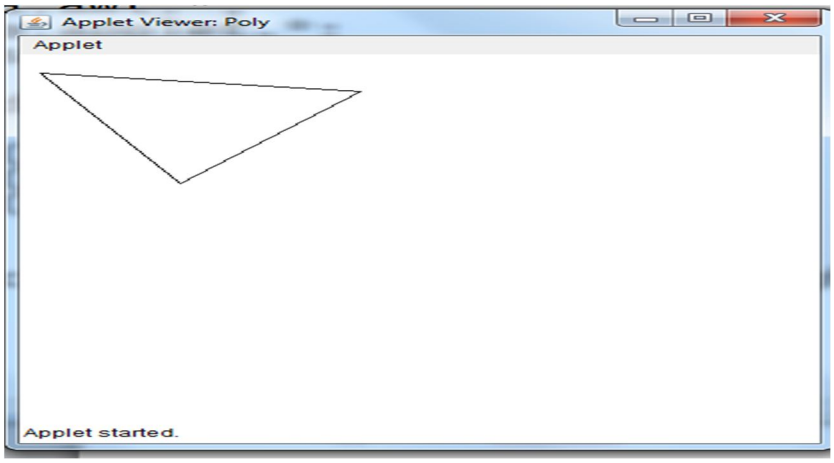


Fig.4. Output of Program draws a polygon of three sides using drawLine method

Note that the end point of the third line is the same as the starting point of the polygon

a) We can draw Polygon more conveniently using the drawPolygon() method of Graphics class

b) This method takes three arguments

i) An array of integers containing X coordinates

ii) An array of integers containing y coordinates

iii) An integers for the total number of points

c) It is obvious that X and Y arrays should be of the same size and we must repeat the first point at the end of the arrays for closing the polygon

Example draws a polygon of three sides using drawPoly method.

```
import java.awt.*;
import java.applet.*;
public class Poly1 extends Applet
{
public void paint (Graphics g)
{
int XPoints[]= {10,170,80,10};
int YPoints[]={20,40,140,20};
int nPoints= XPoints.length;
g.drawPolygon(XPoints,YPoints,nPoints);
}
}/*<Applet code= Poly1.class Width= 400 Height =400> </Applet>*/
```

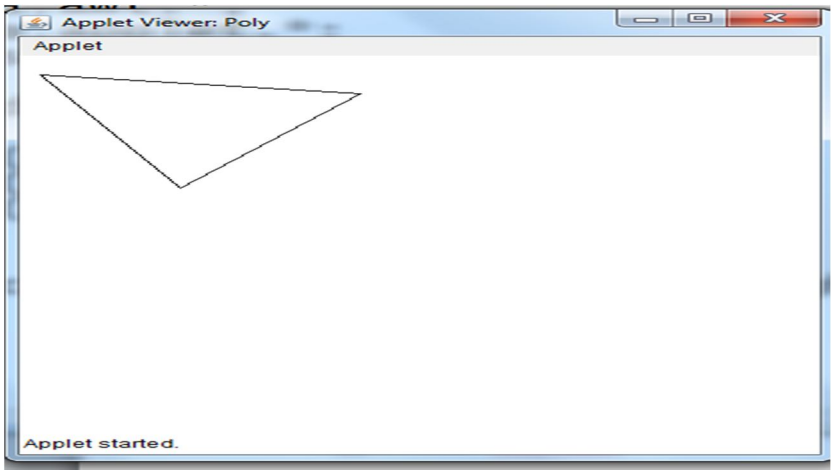


Fig.5.Output draws a polygon of three sides using drawPoly method.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

J. Example for Drawing and Filling Polygons

```
import java.awt.*;
import java.applet.*;
public class Poly2 extends Applet
{
int x1[]={20,120,220,20};
int y1[]={20,120,20,20};
int n1=4;
int x2[]= {120,220,220,120};
int y2[]={120,20,220,120};
int n2=4;
public void paint (Graphics g)
{
g.drawPolygon(x1,y1,n1);
g.fillPolygon(x2,y2,n2);
}
}/*<Applet code= Poly2.class Width = 400 Height = 400> </Applet>*/
```

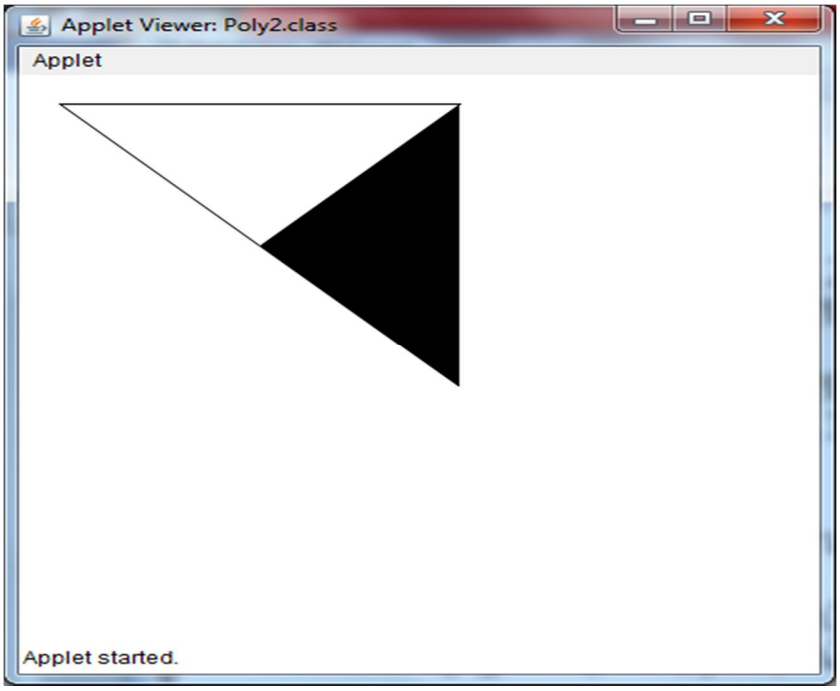


Fig.6.Output of Program Example for drawing and filling polygons

K. Working with Color

1) Java supports color in a portable, device independent fashion
2) The AWT color system allows You to specify any color you want
3) Color defines several constant (examples color.black) to specify a number of common colors
4) You can also create your own color, using one of the color constructors.
5) The most commonly used forms are.
a) Color(int red,int green,int blue)
b) Color(int rgb value)
c) Color(float red, float Green, float blue);

i) The first constructor takes three integers that specify the color as a mix of red, green, blue This values must between 0 and 225.
Colour C1 = new Color(225,100,100);

ii) The second color constructors takes a single integer that contains the mix of red, green , blue, packed into an integer

The integers is organised with red in bits 16 to 23, green in bits 8 to 15 and Blue in bits 0 to 7

Example:-

Int NewRed= (Oxff000000 l (oxco<<16) (oxoo<<8)/ oxx);

Color darkRed= new Color (newRed);

iii) The third constructod , takes three float values between 0.0 and 1.0 that specify the relative mix of red, green , blue

6) Once you have created a color , You cash use it to set the foreground and/or background Color by using following method

L. Background Color

1) To set background color to applet windows use setBackground () method

2) This method is define by components

a) Syntax: setBackground (color new color);

b) Example: setBackground (Color.pink);

M. Foreground Color

1) To set the foreground Color (the color in which text is shown)use setforeground()

2) This method is defined by components

a) Syntax: setforeground (Color.newColor);

b) Example : setforeground (color.red);

Note- A good place to set foreground and Background color is in the init().

N. Settings the current Graphics color:-

1) setColor(Color.newcolor)

a) To set the color to draw and fill in the Current foreground Color

i) Example: setColor(color.Red);

2) getColor()

b) You can obtain the current color by calling getColor ()

i) Example: getColor()

O. Example of Color

```
import java.awt.*;
import java.applet.*;
public class ColorExample extends Applet
{
public void init()
{
setBackground(Color.RED);
setForeground(Color.BLUE);
}
public void paint( Graphics g)
{
g.drawLine(0,0,100,100);
g.drawLine(0,100,100,0);
g.setColor(Color.GREEN);
g.drawOval(10,10,50,50);
g.fillOval(70,90,140,100);
g.setColor(Color.CYAN);
g.fillRect(100,10,60,50);
g.drawRoundRect(140,10,60,50,15,15);
}
} /*<Applet Code= ColorExample.class Width= 400 Height = 400> </Applet>*/
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

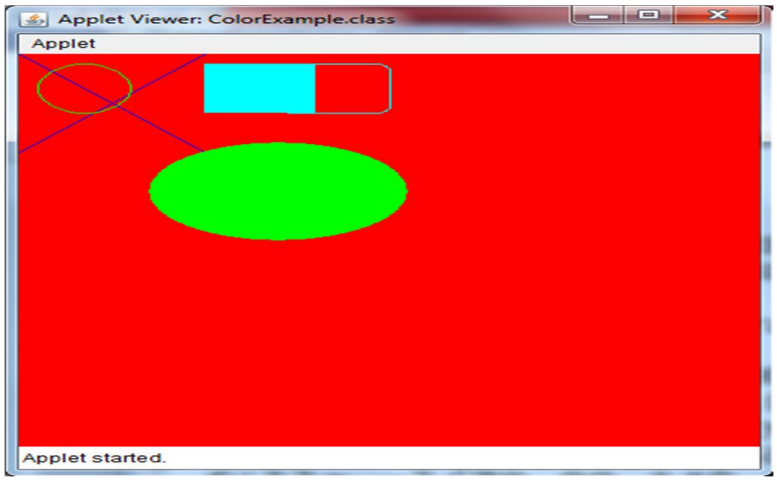


Fig.7.Output of Program Example of color.

P. Working with Fonts

1) The AWT support multiple type fonts
2) Font have emerged from the domain of traditional typesetting to becomes an important part of computer generated documents and display
3) Font have a family name , a logical font name and face name
4) The family name is general name of the font such as courier
5) The logical name specifics a category of font such as Monospaced.
6) The face name specifies a specifies font such as *Courier Italic*
7) Fonts are encapsulated by the Font Class

Table 2. The font class Defines these variables

Variable	Meaning
1) String name	Name of the font
2) Float point size	Size of the font in points
3) Int size	Size of the font in points
4) Int style	Font style

Table 3.Font Method

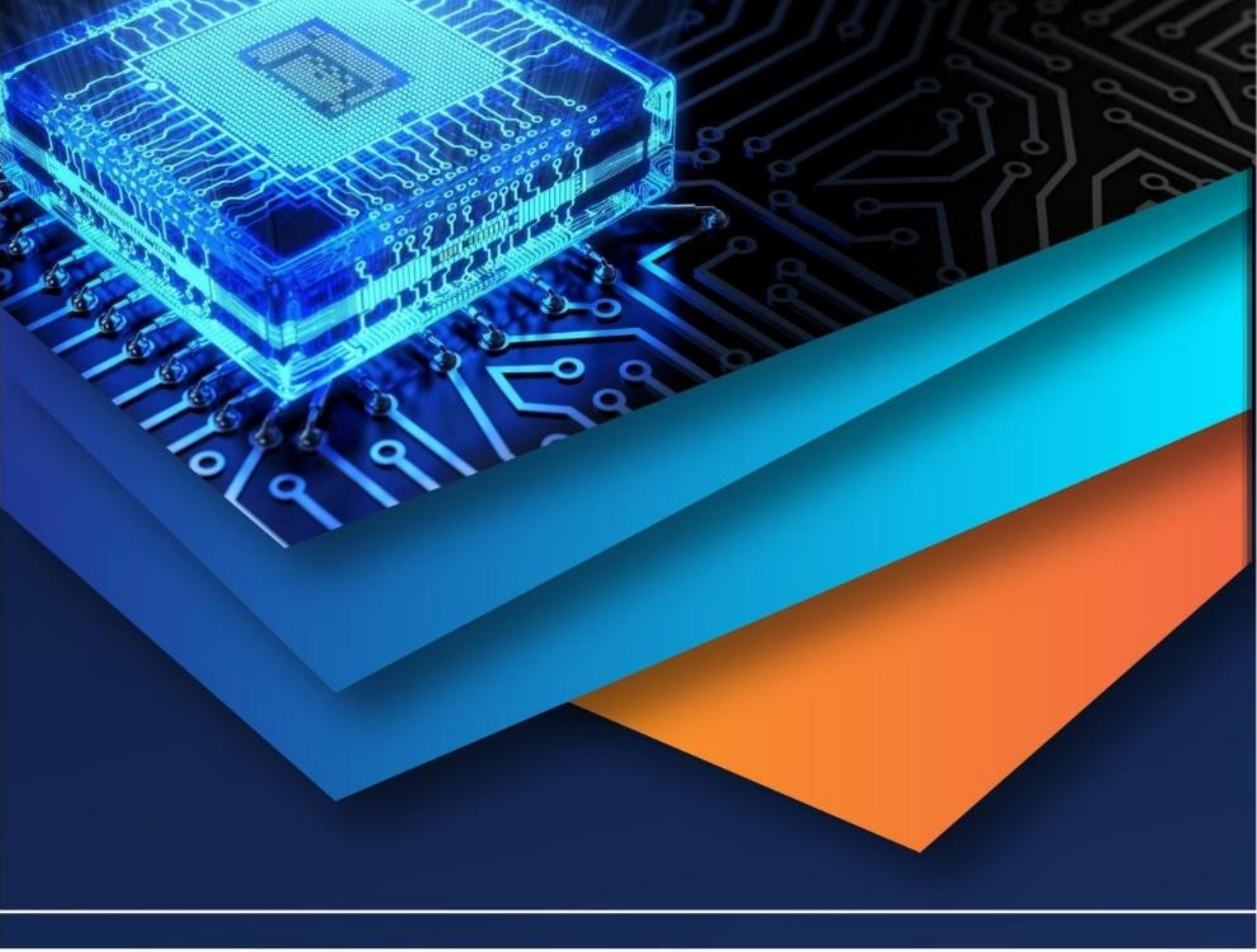
Methods	Description
1) String getFamily()	Return the name of the font family to which the invoking font belongs.
2) Static font getFont(String property)	Return the font associated with the system property specified by properties does not exist
3) Int getsize()	Returns the size , in points of the invoking font
4) String getFontName()	Returns the face name of the invoking font.
5) Int getStyle	Return the style values of the invoking font
6) getAvailableFontFamilyNames()	To know which fonts are available on your Machine
7) getAllFonts()	It remains an array of Font

III. CONCLUSION

This report presents an introduction to Java and how Java is used to build graphics and what are the methods that can be used to drawing the graphics graphics. This was an introduction to the main aim of paper is that design and development a simple program used to draw the shapes like Circle, Line, Rectangle, Square, and Oval, Rounded Rectangle Set Background Color & set Foreground Color.

REFERENCES

[1] "JAVA Primer" by E.Balgurusamy.
[2] 2D Graphics (The Java™ Tutorials) - Oracle Documentation, http://docs.oracle.com/javase/tutorial/2d/overview/index.html, Copyright © 1995, 2013 Oracle
[3] http://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html , 2013
[4] http://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html, 2013
[5] Oracle Technology, http://www.oracle.com/technetwork/java/painting-140037.html#callback



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY