

# Applet

- An applet is a special kind of Java program that runs in a Java enabled browser.
- Applet is typically embedded inside a web page and runs in the browser.
- In other words, we can say that Applets are small Java applications that can be accessed on an Internet server, transported over Internet, and can be automatically installed and run as apart of a web document.

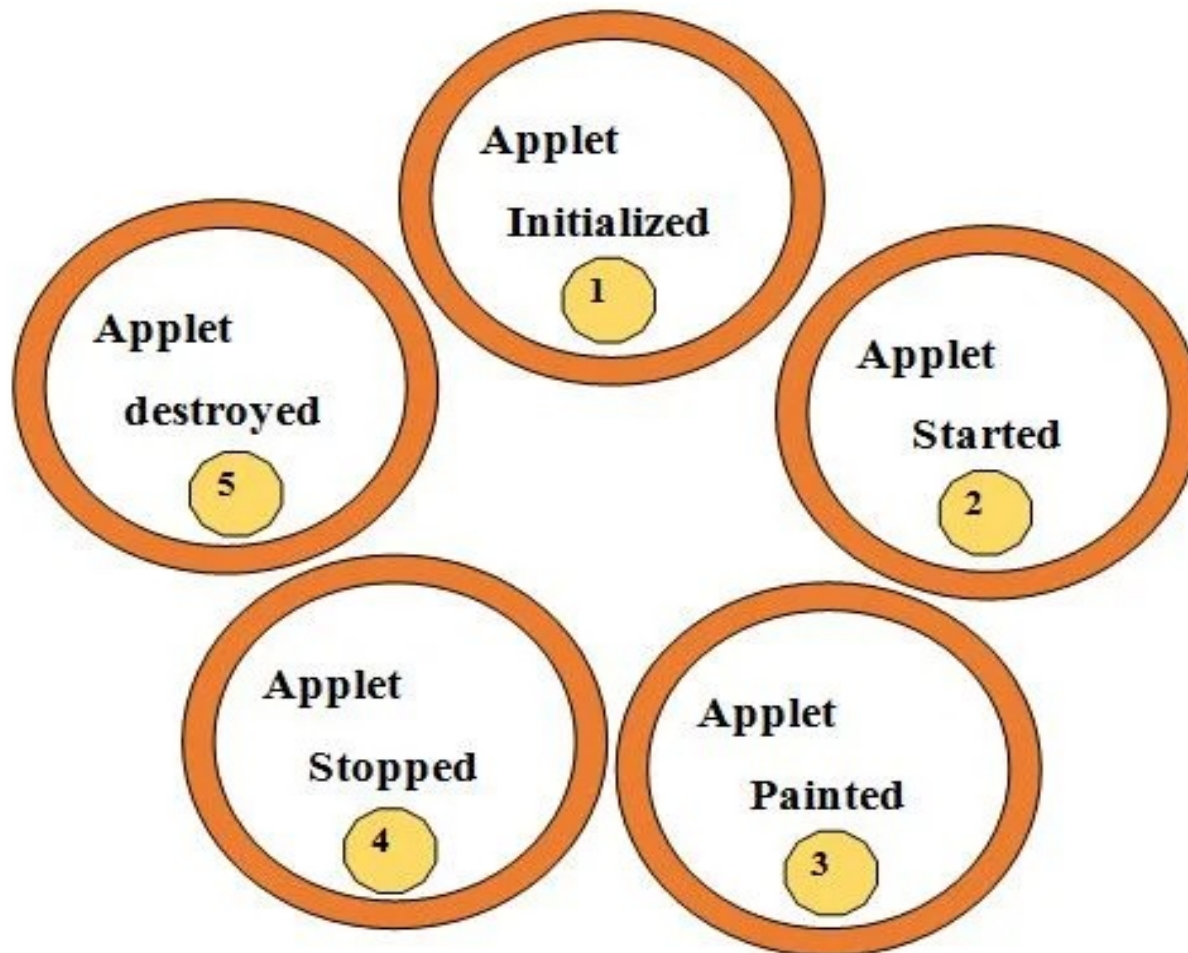
# Applet

- To create an applet, a class must extend `java.applet.Applet` class.
- An Applet class does not have any `main()` method. It is viewed using JVM. The JVM can use either a plug-in of the Web browser or a separate runtime environment to run an applet application.
- JVM creates an instance of the applet class and invokes `init()` method to initialize an Applet.

# Applet

- Lifecycle of Java Applet
- Following are the stages in Applet
  - Applet is initialized
  - Applet is started
  - Applet is painted
  - Applet is stopped
  - Applet is destroyed

# Applet



# Applet

- **Applet class**
- Applet class provides all necessary support for applet execution, such as initializing and destroying of applet. It also provide methods that load and display images and methods that load and play audio clips.

# Applet

- **An Applet Skeleton**
- Most applets override these four methods. These four methods forms Applet lifecycle.
- `init()` : `init()` is the first method to be called. This is where variable are initialized. This method is called only once during the runtime of applet.
- `start()` : `start()` method is called after `init()`. This method is called to restart an applet after it has been stopped.
- `stop()` : `stop()` method is called to suspend thread that does not need to run when applet is not visible.
- `destroy()` : `destroy()` method is called when your applet needs to be removed completely from memory.

# Applet

---

```
import java.awt.*;
import java.applet.*;
public class AppletTest extends Applet
{
    public void init()
    {
        //initialization
    }
    public void start ()
    {
        //start or resume execution
    }
    public void stop()
    {
        //suspend execution
    }
    public void destroy()
    {
        //perform shutdown activity
    }
    public void paint (Graphics g)
    {
        //display the content of window
    }
}
```

# Applet

- Every Applet application must import two packages - `java.awt` and `java.applet`.
- `java.awt.*` imports the Abstract Window Toolkit (AWT) classes. Applets interact with the user (either directly or indirectly) through the AWT. The AWT contains support for a window-based, graphical user interface.
- `java.applet.*` imports the applet package, which contains the class `Applet`. Every applet that you create must be a subclass of `Applet` class.



# Applet

- The class in the program must be declared as public, because it will be accessed by code that is outside the program.
- Every Applet application must declare a paint() method. This method is defined by AWT class and must be overridden by the applet. The paint() method is called each time when an applet needs to redisplay its output.
- Another important thing to notice about applet application is that, execution of an applet does not begin at main() method. In fact an applet application does not have any main() method.

# Applet

- How to run an Applet Program
- An Applet program is compiled in the same way as you have been compiling your console programs. However there are two ways to run an applet.
  - Executing the Applet within Java-compatible web browser.
  - Using an Applet viewer, such as the standard tool, applet viewer. An applet viewer executes your applet in a window

# Applet

- For executing an Applet in an web browser, create short HTML file in the same directory. Inside body tag of the file, include the following code. (applet tag loads the Applet class)
- `< applet code = "MyApplet.class" width=400 height=400  
>`
- `< /applet >`

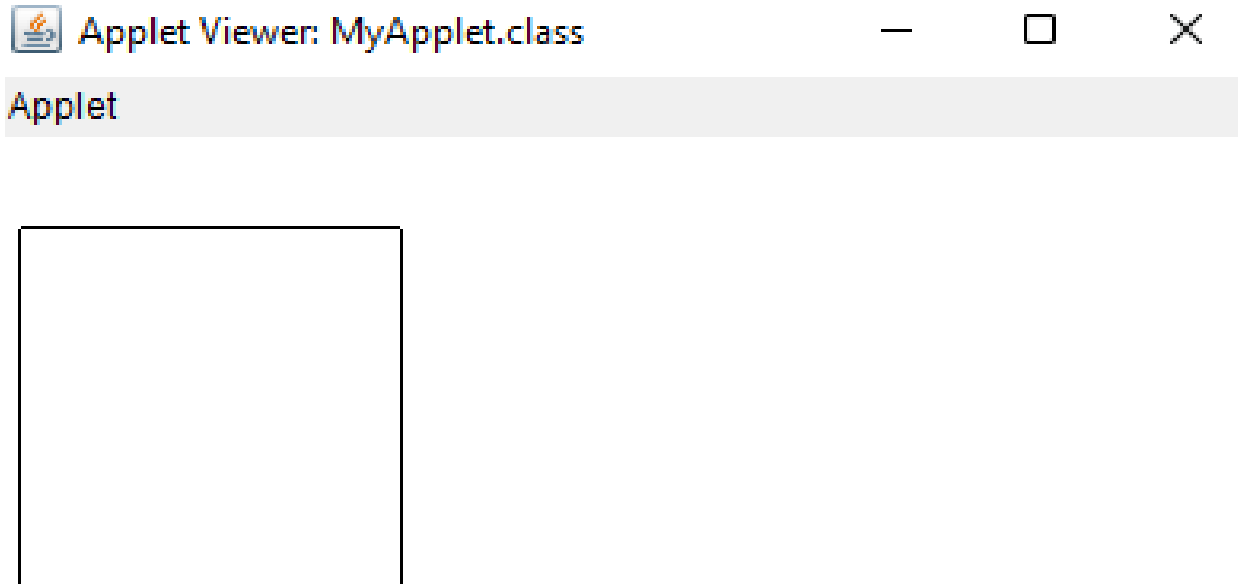
# Applet

- Running Applet using Applet Viewer
- To execute an Applet with an applet viewer, write short HTML file as discussed above. If you name it as run.htm, then the following command will run your applet program.
- `f:/>appletviewer run.htm`

# Applet

```
import java.applet.*;
import java.awt.*;
public class MyApplet extends Applet
{
    int height, width;
    public void init()
    {
        height = getSize().height;
        width = getSize().width;
    }
    public void paint(Graphics g)
    {
        g.drawRect(10, 30, 120, 120);
    }
}
```

# Applet

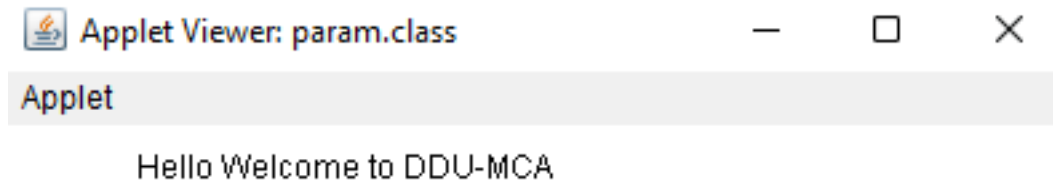


Applet started.

# Applet

```
import java.applet.*;
import java.awt.*;
public class param extends Applet
{
    String str;
    public void init()
    {
        str = "Welcome to DDU-MCA";
        str = "Hello " + str;
    }
    public void paint(Graphics g)
    {
        g.drawString(str, 50, 20);
    }
}
```

# Applet



Applet started.

---



# Applet

- Graphics in Applet
- `java.awt.Graphics` class provides many methods for graphics programming.

# Applet

1. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
2. **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
3. **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
4. **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
5. **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.
6. **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).

# Applet

7. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
8. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
9. **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc.
10. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
11. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

# Applet

---

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

public class GraphicsDemo extends Applet{

    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.drawString("Welcome",50, 50);
        g.drawLine(20,30,20,300);
        g.drawRect(70,100,30,30);
        g.fillRect(170,100,30,30);
        g.drawOval(70,200,30,30);

        g.setColor(Color.pink);
        g.fillOval(170,200,30,30);
        g.drawArc(90,150,30,30,30,270);
        g.fillArc(270,150,30,30,0,180);

    }
}
```

# Applet

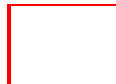


Applet Viewer: GraphicsDemo.class



Applet

Welcome



Applet started.

# Applet

- Displaying Image in Applet
- The `java.awt.Graphics` class provide a method `drawImage()` to display the image.
- Syntax of `drawImage()` method:
  - `public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):` is used draw the specified image.

# Applet

- How to get the object of Image:
- The java.applet.Applet class provides getImage() method that returns the object of Image. Syntax:
  - `public Image getImage(URL u, String image){}`
- Other required method of Applet class to display image:
  - `public URL getDocumentBase():` is used to return the URL of the document in which applet is embedded.

# Applet

---

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

public class DisplayImage extends Applet {

    Image picture;

    public void init() {
        picture = getImage(getDocumentBase(), "A.jpg");
    }

    public void paint(Graphics g) {
        g.drawImage(picture, 30, 30, this);
    }

}
```



# Applet



Applet started.

# Applet

- The 4th argument of drawImage() method of is ImageObserver object. The Component class implements ImageObserver interface. So current class object would also be treated as ImageObserver because Applet class indirectly extends the Component class.