

## Introduction

Layout means the arrangement of components within the container. In other way we can say that placing the components at a particular position within the container. The task of laying out the controls is done automatically by the Layout Manager.

## Layout Manager

The layout manager automatically positions all the components within the container. If we do not use layout manager then also the components are positioned by the default layout manager. It is possible to layout the controls by hand but it becomes very difficult because of the following two reasons.

- It is very tedious to handle a large number of controls within the container.
- Oftenly the width and height information of a component is not given when we need to arrange them.

Java provide us with various layout manager to position the controls. The properties like size, shape and arrangement varies from one layout manager to other layout manager. When the size of the applet or the application window changes the size, shape and arrangement of the components also changes in response i.e. the layout managers adapt to the dimensions of appletviewer or the application window.

The layout manager is associated with every Container object. Each layout manager is an object of the class that implements the `LayoutManager` interface.

Following are the interfaces defining functionalities of Layout Managers.

Sr. No.	Interface & Description
1	<a href="#">LayoutManager</a> The <code>LayoutManager</code> interface declares those methods which need to be implemented by the class whose object will act as a layout manager.
2	<a href="#">LayoutManager2</a> The <code>LayoutManager2</code> is the sub-interface of the <code>LayoutManager</code> . This interface is for those classes that know how to layout containers based on layout constraint object.

## AWT Layout Manager Classes:

Following is the list of commonly used controls while designed GUI using AWT.

Sr. No.	LayoutManager & Description
1	<a href="#">BorderLayout</a> The <code>borderlayout</code> arranges the components to fit in the five regions: east, west, north, south and center.
2	<a href="#">CardLayout</a>

The CardLayout object treats each component in the container as a card. Only one card is visible at a time.

### 3 [FlowLayout](#)

The FlowLayout is the default layout. It layouts the components in a directional flow.

### 4 [GridLayout](#)

The GridLayout manages the components in form of a rectangular grid.

### 5 [GridBagLayout](#)

This is the most flexible layout manager class. The object of GridBagLayout aligns the component vertically, horizontally or along their baseline without requiring the components of same size.

### 6 [GroupLayout](#)

The GroupLayout hierarchically groups components in order to position them in a Container.

### 7 [SpringLayout](#)

A SpringLayout positions the children of its associated container according to a set of constraints.