Mouse

```
import java.awt.*:
import java.awt.event.*:
import java.applet.*:
public class muse extends Applet implements MouseListener. MouseMotionListener
{  int p.q:
   public void init( ){  addMouseListener(this):addMouseMotionListener(this):   }
   public void mouseClicked(MouseEvent k) { p=k.getX( ):q=k.getY( ):}
   public void mouseEntered(MouseEvent k) {}
   public void mouseExited(MouseEvent m) { repaint( ): }
   public void mousePressed(MouseEvent n) {}
   public void mouseReleased(MouseEvent k){}
   public void mouseDragged(MouseEvent p){}
   public void mouseMoved(MouseEvent e){}
   public void paint(Graphics f){ f.drawOval(p-5,q-5,10,10); }
}
```

When mouse is moved out of applet a circle is displayed at location of last mouse click

1. Circle when mouse enters the window at the place where the mouse exited (radius 10).
2. When mouse is clicked a circle (radius 70) is displayed. The center is the place of click.
3. Circle when mouse released at the place where the mouse was pressed. Radius 10.
4. Line when mouse is released. The end points are places of mouse press and release.
5. Line is drawn from the place of mouse pressed to mouse dragged.
6. Circle when mouse is released at the place midway between mouse press and mouse released (radius 10).
7. Circle when mouse is released. The place is in opposite direction (same distance) of mouse pressed. If mouse is pressed at (150.78) and is released at (200,67) then the circle will appear at (100,89). [press (100,30) release (110,35) Circle (90,25)] radius 10
8. Circle when mouse is exited from the window. The place is midway between mouse press and mouse release (radius 10). [The last occurrence of mouse press and mouse release is taken]
9. Rectangle when mouse is released. The end points are places of mouse press and release. Assume that location of mouse release is east south of the place of mouse press.
10. Do above problem, when no such assumption is made.
11. Circle when mouse is released. Centre is the place of mouse pressed. The radius is the distance between mouse pressed and mouse released. [int g=(int)(Math.sqrt(b))]
12. Circle when mouse clicked. (A) The location is the place where the mouse was previously clicked. (B) The location where mouse was clicked two steps back. (r=10)
13. At the place of mouse click display a string "ram".
14. Read a string. Display it at the place of mouse click. [Hint: read string in init]
15. Read a string. Display its 0th latter at the place of first mouse click, 1st letter at the place of 2nd mouse click and so on.
16. Initially two circles appear. When mouse is clicked inside any circle then radius of that circle increases by 5. The radius of other circle is unchanged. This is done repeatedely. Initially circles have centre (100,100) and (200,200) respectively. The radius is 30 each.
17. Initially 4 circles appear. When mouse is clicked inside any circle then it disappears. Assume that mouse will be clicked at most 4 times. Assume radius of each circle is 30 each. Their centres are (100,100). (100,200). (200,100) and (200,200) respectively.

```
int p.q.k:
public void mouseClicked(MouseEvent g) {  p=g.getX( ):q=g.getY( ):k=1;repaint( ):}
public void mousePressed(MouseEvent n) {  p=n.getX( ):q=n.getY( ):k=2:repaint( ): }
public void paint(Graphics f)
{   if (k==1) f.drawOval(p.q.10.10). if (k==2) f.drawOval(p.q.100.100):   }
```

When mouse is clicked a small circle is drawn. When it is pressed a big circle is drawn

18. Circle when mouse pressed. Rectangle when mouse released. Location any where

8.5 A circle appears as soon as the mouse is clicked at the place of click (Radius 10). When the mouse exit, the window the circle moves left (5unit). When mouse enters radius 20.