

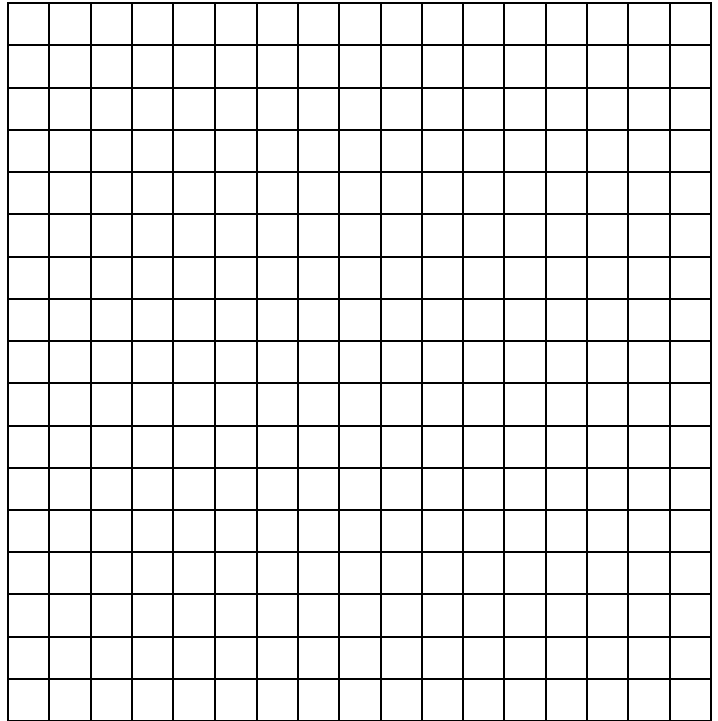
1. Vocabulary – Fill in the answer column. BE SPECIFIC!

Term	Question	Answer
MouseListener	What two Event Handler Methods correspond to this?	1. 2.
x= y.isMetaDown()	a) What data type is x? b) What type of thing is y? c) What does isMetaDown() check for?	a) b) c)
HTML tag	Give an example of one of these (show HTML code)	
super.paintComponent() and setBackground()	These are related, but do very different things. What does the first one DO?	
focusGained()	What is this used for? (In other words, what would happen if this was not used?)	
mouseExited()	What kind of activity will cause this method to be executed/called?	

2. On the graph provided, draw as accurately as possible the output of the following program.

```
g.drawOval(0, 50, 30, 30);
g.drawArc(20, 60, 15, 15, 170, 110);
int [] x = {30, 45, 45};
int [] y = {65, 75, 55};
g.drawPolygon(x, y, 3);
g.drawRect(45, 55, 75, 20);
g.drawString("2", 80, 70);
g.fillArc(110, 55, 20, 20, 270, 180);
for (int coor=105; coor<=115; coor+=5)
    g.drawLine(coor, 55, coor, 75);
```

You may assume that each square to the right is 10 pixels by 10 pixels. The entire grid is 170 pixels by 170 pixels.



3. Roshambo.java

In Rock-Paper-Scissors (also known as Ro-Sham-Bo), a user plays against the computer on a 300 by 300 JApplet.

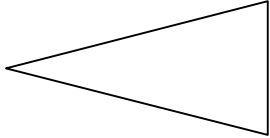
See the next page for detailed instructions.

Here's a sequence of screen shots so you can see how the game should run:



Your job is to complete the following java code –
don't worry about HTML code.

- On mouse click one, a red “1” should be printed to the screen, using a big font. This means you should use this line of code before you print to the JPanel: `g.setFont(“Courier”, Font.BOLD, 30);`
- On mouse click two, a red “2” should be printed to the screen (same font size as before).
- On mouse click three, a red “3” should be printed to the screen (same font size as before).
- Only after there have been three mouse clicks, should the computer use the keyboard to show the rock, paper, or scissors the user selects. For example, to select rock, the user should use ‘r’ or ‘R’ on the keyboard. Use the chart below to determine what should be used for paper and scissors. No other keyboard input should cause the computer to display or do anything else.
- As soon as the keyboard is used, a graphic should appear on the screen. For example, to show the rock, a 100-diameter white circle should be drawn. Use the chart below to determine what should be displayed for scissors and paper.

User wants to show...	Keyboard Item Selected...	Graphic displayed...
Rock	‘r’ or ‘R’	100 pixel diameter white circle
Paper	Up arrow	100 pixel high white square
Scissors	‘%’ (shift and ‘5’)	100 pixel long, 50 pixel high white triangle like this: 

- Immediately after the user’s selection is shown, use `Math.random()` to determine which the computer picks, calculating 1, 2, or 3. The computer’s rock, paper, and scissors are the same size and shape as those shown in the chart above, and are also white. If 1 is the random number, then draw a white rock. If 2, draw white paper. If 3, draw white scissors. Draw the second shape as shown in the example output below.
- Print a sentence to the screen to indicate who won, in red (same font size as before). The sentence should be “User Wins!” “It’s A Tie!” or “Machine Wins!” Rock wins over scissors. Scissors win over paper. Paper wins over rock. A tie is when both the user and computer pick the same thing.

```
import // assume all imports are given to save space here for java.awt. java.awt.event; javax.swing
```

```
public class Roshambo
{
    public static void main ( ) // This method is complete
    {
        assume main calls a method that makes a frame that is 300 by 300, does all of the usual, makes an instance of
        RoshamboPanel, adds it to the frame and makes the frame visible.
    }
}

class RoshamboPanel extends JPanel implements KeyListener, MouseListener
{
    private boolean drawfigures;
    private int count, user, computer;
```

```

public RoshamboPanel ( ) // This method is complete
{
    drawfigures = false;
    count = computer = user = 0;
    setBackground(Color.BLACK);
    addKeyListener(this);
    addMouseListener(this);
}

public void paintComponent(Graphics g) // This method is complete
{
    super.paintComponent(g);
    Font myFont = new Font ( "Courier", Font.BOLD, 30 );
    g.setFont ( myFont );
    g.setColor ( Color.RED );
    g.drawString ( "ROSHAMBO", 80, 35 );
    if ( count != 0 )
        g.drawString ( "" + count, 145, 90 );
    else if (drawfigures)
    {
        drawUserFigure(g);
        drawMachineFigure(g);
        determineWinner(g);
    }
}

public void drawUserFigure ( Graphics g )
{
}

}

```

```
public void drawMachineFigure ( Graphics g )
```

```
{
```

```
}
```

```
public void determineWinner ( Graphics g ) // This method is complete
```

```
{    if ((user == 1 && computer == 2) || (user == 2 && computer == 3) || (user == 3 && computer == 1))
```

```
        g.drawString ( "MACHINE WINS!", 40, 280 );
```

```
    else if ((user == 2 && computer == 1) || (user == 3 && computer == 2) || (user == 1 && computer == 3))
```

```
        g.drawString ( "USER WINS!", 60, 280 );
```

```
    else
```

```
        g.drawString ( "IT'S A TIE!", 60, 280 );
```

```
}
```

```
public void keyTyped ( KeyEvent evt )
```

```
{
```

```
}
```

```
public void keyPressed ( KeyEvent evt )  
{
```

```
}
```

```
public void keyReleased ( KeyEvent evt ) {} // This method is complete
```

```
public void mouseClicked ( MouseEvent evt ) {}
```

```
public void mousePressed ( MouseEvent evt )
```

```
{
```

```
    // make it so the panel will respond to the key events
```

```
}
```

```
public void mouseReleased ( MouseEvent evt ) {} // This method is complete
```

```
public void mouseEntered ( MouseEvent evt ) {} // This method is complete
```

```
public void mouseExited ( MouseEvent evt ) {} // This method is complete
```

```
}
```