

## Chapter 9 – GUI using AWT

### Question 1:

You construct a list by calling

```
new List(10, true).
```

Which statement or statements below are correct? (Assume that layout manager doesn't modify the list in any way.)

- ☐ A) The new List object has 10 rows.
- ☐ B) The new List object has 10 rows and it supports multiple selection.
- ☐ C) The new List object has 10 rows and it doesn't support multiple selection.
- ☐ D) The new List object doesn't support multiple selection.
- ☐ E) The new List object supports multiple selection.

### Question 2:

You have a checkbox in a panel. the panel is in an applet. The applet contains no other components. Using `setFont()`, you give the applet a 20-point font, and you give the panel a 10-point font. Which statement or statements below are correct?

- ☐ A) The checkbox uses a 20-point font.
- ☐ B) The checkbox uses a 10-point font.
- ☐ C) The checkbox uses the applet's font, because you can't set a font bigger than 6-points on a panel.
- ☐ D) The checkbox uses the applet's font, because you can't set a font on a panel.
- ☐ E) The checkbox doesn't use the panel's font.

Question 3:

True or False:

A menu can contain a Checkbox component.

- ☐ A) true
- ☐ B) false

Question 4:

True or False:

By setting the attributes of a component that was added to the container, the attributes that were set determine the look of the component. If the component's attributes weren't set then the container attributes would determine. Each component's attribute that wasn't set specifically gets the value of the matching container attribute.

- ☐ A) true
- ☐ B) false

Question 5:

You construct a text area by calling

```
new TextArea(8, 24 ).
```

Which statement or statements below are correct? (Assume that layout manager doesn't modify the text area in any way.)

- ☐ A) The new text area has 8 rows.
- ☐ B) The new text area height is 8 rows.
- ☐ C) The new text area width is 24 character.
- ☐ D) The new text area can holds a line of text which is longer than 24

characters

- ☐ E) The new text area can hold at the most a line of 24 characters

Question 6:

True or False:

The CheckboxGroup class extends the Component class.

- ☐ A) true  
☐ B) false

Question 7:

True or False:

A Menu can contain checkboxes.

- ☐ A) true  
☐ B) false

Question 8:

True or False:

If a text field is constructed using the constructor that receives a String reference, and later, it is assigned a String which is longer then the text field becomes wider.

- ☐ A) true
- ☐ B) false

Question 9:

True or False:

A panel can contain a menu bar.

- ☐ A) true
- ☐ B) false

Question 10:

True or False:

An applet can contain a menu bar.

☐ A) true

☐ B) false

Question 11:

True or False:

A panel can contain a menu.

☐ A) true

☐ B) false

Question 12:

True or False:

The components appear in the same way on every platform.

- ☐ A) true
- ☐ B) false

Question 13:

True or False:

With the `GridLayoutManager` the number of lines can be changed according to the number of the components that were added.

- ☐ A) true
- ☐ B) false

Question 14:

True or False:

With the GridLayoutManager all of the components that within him have the same width and height.

- ☐ A) true
- ☐ B) false

Question 15:

True or False:

Putting an horizontal scroll bar at East in a frame that use the BorderLayout manager won't be practical as putting it at South.

- ☐ A) true
- ☐ B) false



Question 16:

Given the code below,

```
1.  class Guky
2.  {
3.      public static void main(String args[])
4.      {
5.          Frame f = new Frame();
6.          Panel p = new Panel();
7.          Button b = new Button();
8.          b.setBounds(10, 10, 250, 180);
9.          p.add(b);
10.         f.add(p);
11.         f.setSize(400,300);
12.         f.setVisible(true);
13.     }
14. }
```

which of the following sentences (one or more) is true ?

- ☐ A) The button width will be 250
- ☐ B) The button height will be 180
- ☐ C) The button height will be 10
- ☐ D) The button width will be 10
- ☐ E) The button width and height will be determined by the relevant layout manager.

Question 17:

True or False:

When adding a component to container that a BorderLayout manage its components location, it is legal to omit the second parameter to the add() call. In this case, the BorderLayout manager will assume that the component should be placed in the center.

- ☐ A) True
- ☐ B) False

Question 18:

Which of the following sentences (one or more) is true ?

- ☐ A) The Label component doesn't respond to user input, and doesn't send out any events.
- ☐ B) The default alignment for labels is to the left.
- ☐ C) It is possible to change the text that the label component represent more than once.
- ☐ D) A list component is a collection of text items, arranged vertically. If a list component contains more items than it can display, then a vertical scroll bar is displayed.

Question 19:

True or False:

When a text area component is created using the constructor that receives the number of rows and the number of columns, the exact width depends on the Font that is used by the text area component.

- ☐ A) True
- ☐ B) False

Question 20:

Which of the following sentences (one or more) is true ?

- ☐ A) A Dialog is a pop-up window that accepts user input.
- ☐ B) The Dialog class is the superclass of the FileDialog class.
- ☐ C) The default layout manager of the Dialog class is border layout.

Question 21:

True or False:

There are four kinds of elements that can populate a menu:

1. Menu items
2. Check-box menu items
3. Separators
4. Menus

☐ A) True

☐ B) False

Question 22:

Which of the following sentences (one or more) is true ?

☐ A) In creating a Font instance the first parameter to be sent is the name of the font (String). Font availability is platform dependent. It is possible to get a list of available font names by calling the `getFontList()` method from the current toolkit. For example:

```
String names[ ] = Toolkit.getDefaultToolkit().getFontList();
```

☐ B) In creating a Font instance the second parameter to be sent is the style parameter. The style parameter should be one of the following three options:

1. Font.PLAIN
2. Font.BOLD
3. Font.ITALIC

☐ C) In creating a Font instance the third parameter to be sent is the font size.

Question 23:

Given the following code,

```
g.drawRect(40,20,100,80);
```

Which of the following sentences (one or more) is true ?

- ☐ A) The rectangle that will be painted will have a width of 60
- ☐ B) The rectangle that will be painted will have a height of 60
- ☐ C) The rectangle that will be painted will have a width of 100
- ☐ D) The rectangle that will be painted will have a height of 80
- ☐ E) The upper left corner of the rectangle that will be painted will be in (40,20)

Question 24:

Given the following code,

1. `int xes[]={40, 100, 100, 40};`
2. `int yes[]={ 40, 40, 100, 100};`
3. `g.drawPolygon(xes, yes, 4);`

which of the following answers is true ?

- ☐ A) The output is a rectangle
- ☐ B) The output is a triangle
- ☐ C) The output is a broken line
- ☐ D) The code won't compile successfully
- ☐ E) The code will compile, but during execution an exception will be thrown

Question 25:

Given the code below as part of an applet code,

```
1.  public void paint(Graphics g)
2.  {
3.      g.drawString("ISRAEL",0,0);
4.  }
```

What will be the output ?

- ☐ A) The word ISRAEL will be shown in the upper left corner of the applet.
- ☐ B) The word ISRAEL won't be shown on the screen.
- ☐ C) The word ISRAEL will be shown in the upper right corner of the applet.
- ☐ D) The word ISRAEL will be shown in the center of the applet.

Question 26:

True or False:

When using the GridLayout Manager the number of the rows or the columns can be zero (in the constructor). By doing so the number of the rows\cols (the one that was set to zero) will be flexible.

- ☐ A) True
- ☐ B) False

Question 27:

Given the code below:

```
1.  public class Mac extends java.applet.Applet
2.  {
3.      public void init()
4.      {
5.          add(new java.awt.Button("press"));
6.      }

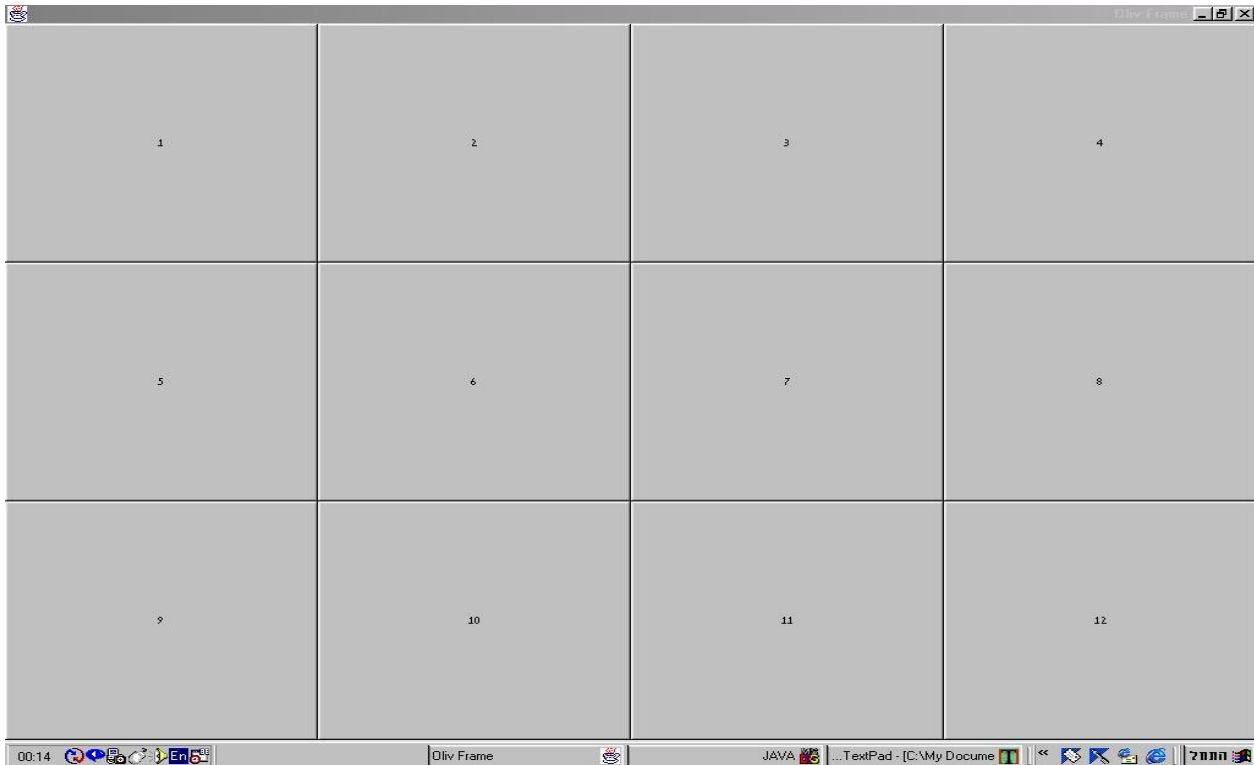
7.      public static void main(String args[])
8.      {
9.          java.awt.Frame f = new java.awt.Frame("zulu");
10.         java.awt.Panel peny = new Mac();
11.         ((java.applet.Applet)peny).init();
12.         f.add(peny);
13.         f.setSize(400,300);
14.         f.setVisible(true);
15.     }
16. }
```

- ☐ A) It will compile successfully
- ☐ B) It will not compile successfully
- ☐ C) The code can function as an application
- ☐ D) The code can function as an applet



Question 28:

Given the GUI screen below:



and the code below:

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

```
import java.awt.event.*;
```

```
public class OlivFrame extends Frame
```

```
{
```

```
    public static void main(String args[])
```

```
{
```

```

        OlivFrame frame = new OlivFrame("Oliv Frame");
        frame.go();
    }

    public OlivFrame(String str)
    {
        super(str);
    }

    public void go()
    {
        setLayout(_____);
        Button buttons[] = new Button[12];
        for(int i=0; i<buttons.length; i++)
        {
            buttons[i] = new Button(""+(i+1));
            add(buttons[i]);
        }
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent event)
            {
                System.exit(0);
            }
        }
    }
};

```

```
        setSize(500,500);  
        setVisible(true);  
    }  
}
```

The missing code can be (choose one or more):

- ☐ A) new GridLayout(3,4)
- ☐ B) new GridLayout(3,3)
- ☐ C) new GridLayout(3,2)
- ☐ D) new GridLayout(3,1)
- ☐ E) new GridLayout(3,5)
- ☐ F) new GridLayout(2,4)

Question 29:

True or False:

Given the GUI screen below:



The code below might create it.

```
import java.awt.*;  
import java.awt.event.*;
```

```

public class OlivFrame extends Frame
{
    public static void main(String args[])
    {
        OlivFrame frame = new OlivFrame("Oliv Frame");
        frame.go();
    }

    public OlivFrame(String str)
    {
        super(str);
    }

    public void go()
    {
        setLayout(new FlowLayout());
        Button buttons[] = new Button[12];
        for(int i=0; i<buttons.length; i++)
        {
            buttons[i] = new Button(""+(i+1));
            add(buttons[i]);
        }
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent event)

```

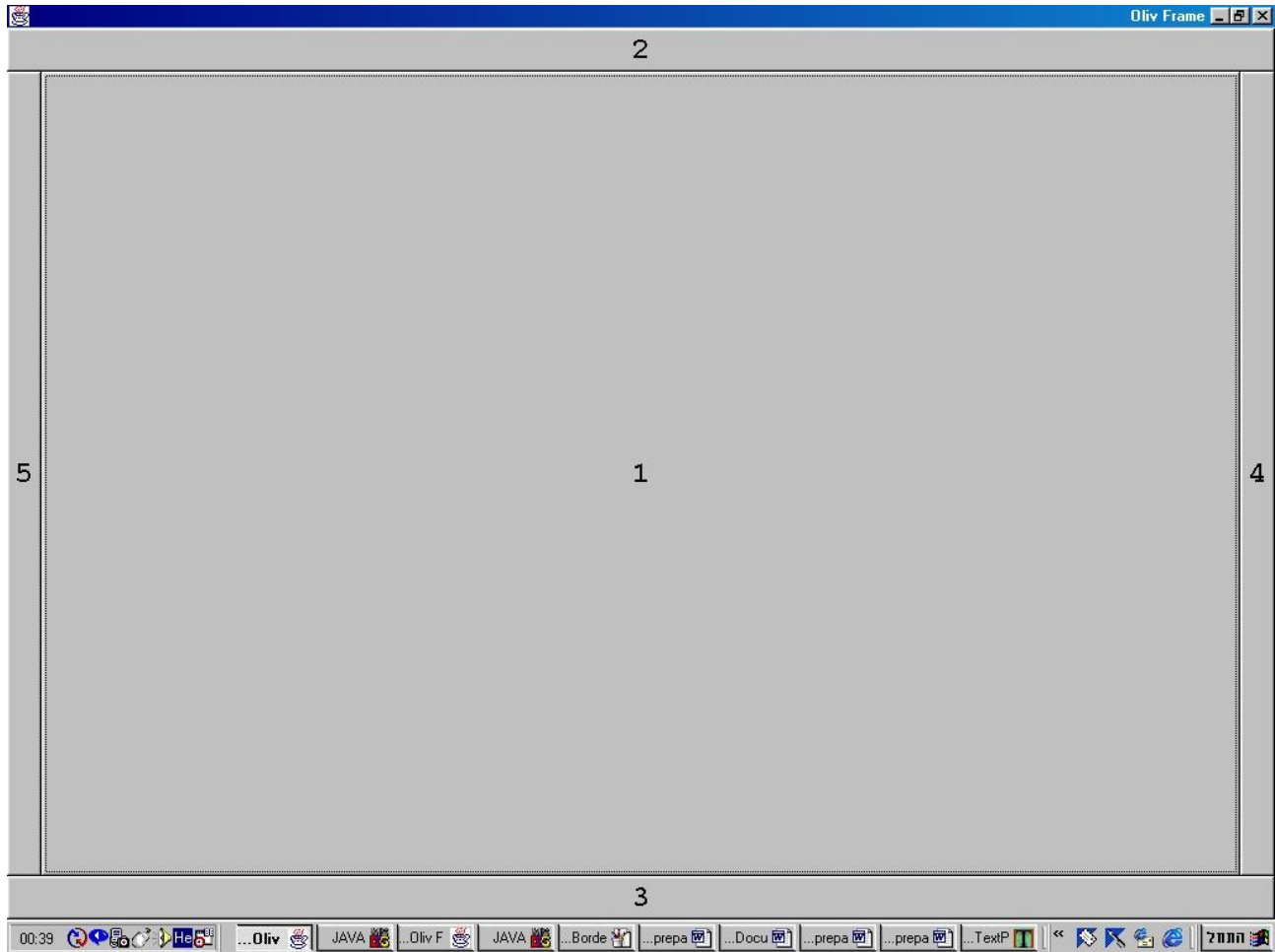
```
        {  
            System.exit(0);  
        }  
    }  
    );  
    setSize(500,500);  
    setVisible(true);  
}  
}
```

- ( ) A) True
- ( ) B) False

Question 30:

True or False:

Given the GUI screen below:



The code below might create it:

```
import java.awt.*;  
import java.awt.event.*;
```

```
public class OlivFrame extends Frame
{
    public static void main(String args[])
    {
        OlivFrame frame = new OlivFrame("Oliv Frame");
        frame.go();
    }

    public OlivFrame(String str)
    {
        super(str);
    }

    public void go()
    {
        setLayout(new BorderLayout());
        Button buttons[] = new Button[5];
        String position[] = {"Center", "North", "South", "East", "West"};
        for(int i=0; i<buttons.length; i++)
        {
            buttons[i] = new Button(""+(i+1));
            buttons[i].setFont(new Font("Courier", Font.BOLD, 24));
            add(buttons[i], position[i]);
        }
    }
}
```



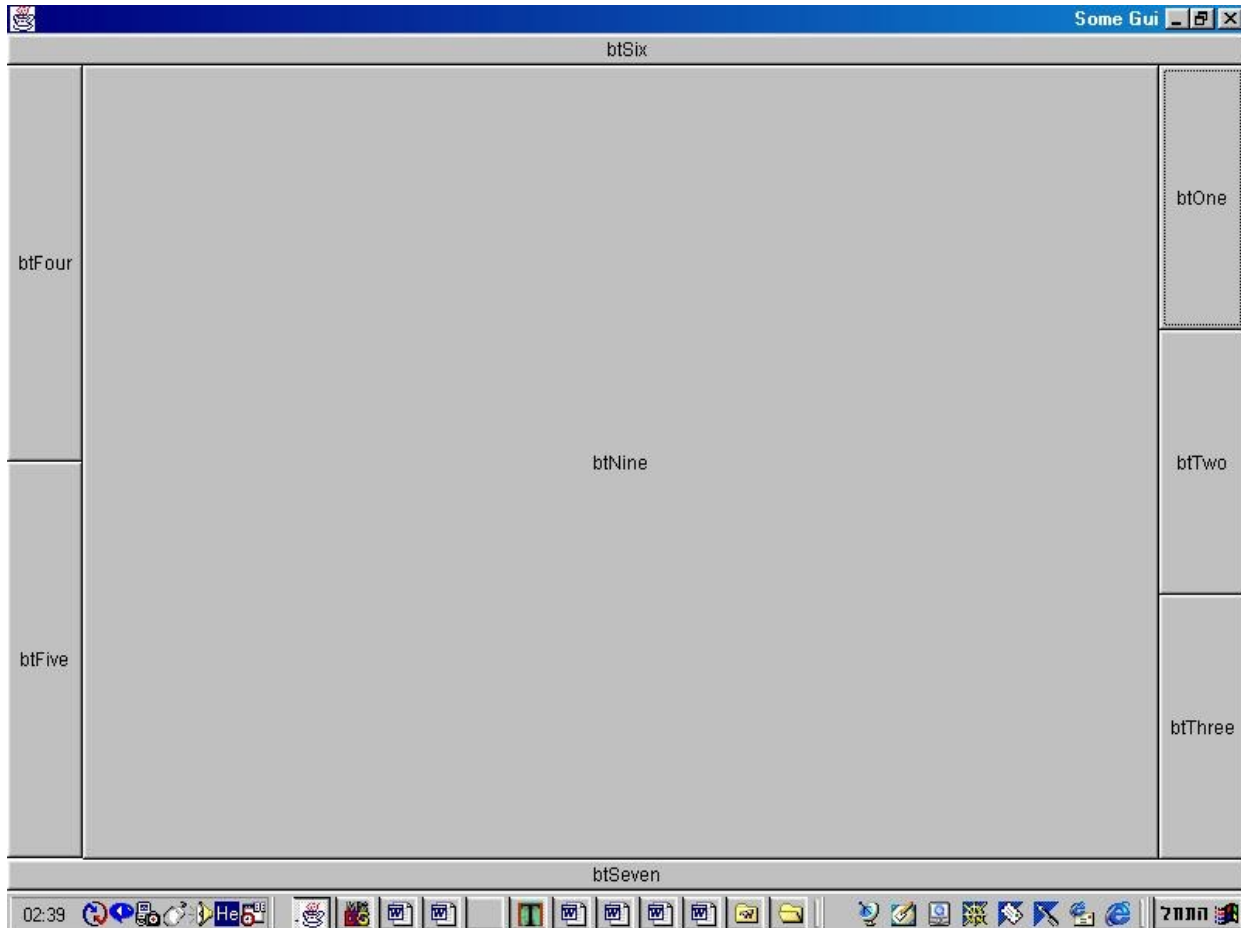
```
addWindowListener(new WindowAdapter()  
{  
    public void windowClosing(WindowEvent event)  
    {  
        System.exit(0);  
    }  
});  
setSize(500,500);  
setVisible(true);  
}  
}
```

- ( ) A) True
- ( ) B) False

Question 31:

True or False:

Given the GUI screen below:



The code below might create it:

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

```
public class SomeGui extends Frame
```

```
{  
  
    public SomeGui(String str)  
    {  
        super(str);  
    }  
  
    public static void main(String args[])  
    {  
        SomeGui sg = new SomeGui("Some Gui");  
        sg.go();  
    }  
  
    public void go()  
    {  
        Panel panel1 = new Panel();  
        Panel panel2 = new Panel();  
        panel1.setLayout(new GridLayout(3,1));  
        panel2.setLayout(new GridLayout(2,1));  
        panel1.add(new Button("btOne"));  
        panel1.add(new Button("btTwo"));  
        panel1.add(new Button("btThree"));  
        panel2.add(new Button("btFour"));  
        panel2.add(new Button("btFive"));  
        add(panel1, "East");  
        add(panel2, "West");  
    }  
}
```

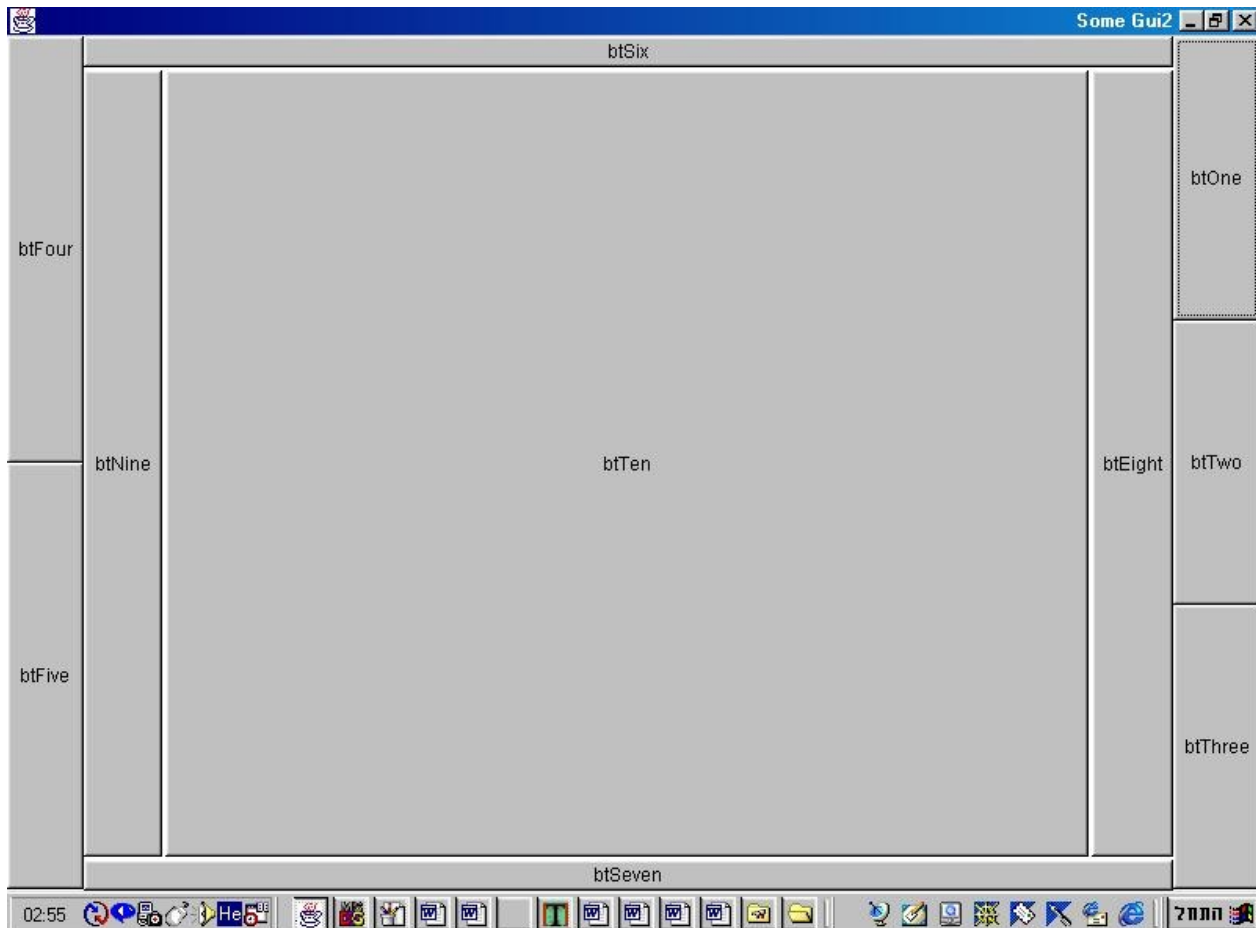
```
        add(new Button("btSix"), "North");  
        add(new Button("btSeven"), "South");  
        add(new Button("btEight"), "Center");  
        add(new Button("btNine"));  
  
        setSize(600, 400);  
  
        setVisible(true);  
    }  
}
```

- ( ) A) True
- ( ) B) False

Question 32:

True or False:

Given the GUI screen below:



The code below might create it:

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

```
public class SomeGui2 extends Frame
```

```
{  
  
    public SomeGui2(String str)  
    {  
        super(str);  
    }  
  
    public static void main(String args[])  
    {  
        SomeGui2 sg = new SomeGui2("Some Gui2");  
        sg.go();  
    }  
  
    public void go()  
    {  
        Button btOne = new Button("btOne");  
        Button btTwo = new Button("btTwo");  
        Button btThree = new Button("btThree");  
        Button btFour = new Button("btFour");  
        Button btFive = new Button("btFive");  
        Button btSix = new Button("btSix");  
        Button btSeven = new Button("btSeven");  
        Button btEight = new Button("btEight");  
        Button btNine = new Button("btNine");  
        Button btTen = new Button("btTen");  
        Panel panel1 = new Panel();  
    }  
}
```

```
Panel panel2 = new Panel();  
Panel panel3 = new Panel();  
panel1.setLayout(new GridLayout(3,1));  
panel2.setLayout(new GridLayout(2,1));  
panel3.setLayout(new BorderLayout(2,2));  
panel1.add(btOne);  
panel1.add(btTwo);  
panel1.add(btThree);  
panel2.add(btFour);  
panel2.add(btFive);  
panel3.add(btSix, "North");  
panel3.add(btSeven, "South");  
panel3.add(btEight, "East");  
panel3.add(btNine, "West");  
panel3.add(btTen, "Center");  
add(panel1, "East");  
add(panel2, "West");  
add(panel3, "Center");  
setSize(600, 400);  
setVisible(true);  
}  
}
```

- ( ) A) True
- ( ) B) False

Question 33:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.  public class SimpleTF
5.  {
6.      Frame frame;
7.      TextField tf1, tf2, tf3;
8.
9.      SimpleTF()
10.     {
11.     }
12.
13.     public void go()
14.     {
15.         frame = new Frame();
16.         frame.setLayout(new FlowLayout());
17.         tf1 = new TextField(5);
18.         tf2 = new TextField(5);
19.         frame.add(tf1);
20.         frame.add(tf2);
21.         frame.setSize(400,400);
```



```
22.         frame.setVisible(true);
23.         frame.addWindowListener(new WindowAdapter()
24.             {
25.                 public void windowClosing(WindowEvent event)
26.                 {
27.                     System.exit(0);
28.                 }
29.             }
30.         );
31.         tf3 = new TextField(5);
32.         frame.add(tf3);
33.     }
34.
35.     public static void main(String args[])
36.     {
37.         SimpleTF simpleTF = new SimpleTF();
38.         simpleTF.go();
39.     }
40. }
```

Assuming that the user doesn't change the dimension of the frame, how many text fields will be displayed?

( ) A) 2

( ) B) 3

- ( ) C) The code doesn't compile successfully
- ( ) D) 1
- ( ) E) 0

Question 34:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.  public class SimpleTF
5.  {
6.      Frame frame;
7.      TextField tf1, tf2, tf3;
8.
9.      SimpleTF()
10.     {
11.     }
12.
13.     public void go()
14.     {
15.         frame = new Frame();
16.         frame.setLayout(new FlowLayout());
```

```
17.         tf1 = new TextField(5);
18.         tf2 = new TextField(5);
19.         tf3 = new TextField(5);
20.         frame.add(tf1);
21.         frame.add(tf2);
22.         frame.add(tf3);
23.         frame.setSize(400,400);
24.         frame.setVisible(true);
25.         frame.addWindowListener(new WindowAdapter()
26.             {
27.                 public void windowClosing(WindowEvent event)
28.                 {
29.                     System.exit(0);
30.                 }
31.             }
32.         );
33.     }
34.
35.     public static void main(String args[])
36.     {
37.         SimpleTF simpleTF = new SimpleTF();
38.         simpleTF.go();
39.     }
40. }
```

Assuming that the user doesn't change the dimension of the frame, how many text fields will be displayed?

- ☐ A) 2
- ☐ B) 3
- ☐ C) The code doesn't compile successfully
- ☐ D) 1
- ☐ E) 0

Question 35:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.  public class SimpleTF
5.  {
6.      Frame frame;
7.      TextField tf1, tf2, tf3;
8.
9.      SimpleTF()
10.     {
11.     }
```

```
12.
13.     public void go()
14.     {
15.         frame = new Frame();
16.         frame.setLayout(new FlowLayout());
17.         tf1 = new TextField(5);
18.         tf2 = new TextField(5);
19.         tf3 = new TextField(5);
20.         frame.add(tf1);
21.         frame.add(tf2);
22.         frame.setSize(400,400);
23.         frame.setVisible(true);
24.         frame.addWindowListener(new WindowAdapter()
25.             {
26.                 public void windowClosing(WindowEvent event)
27.                 {
28.                     System.exit(0);
29.                 }
30.             }
31.         );
32.         frame.add(tf3);
33.         frame.validate();
34.     }
35.
36. public static void main(String args[])
```

```
37. {  
38.     SimpleTF simpleTF = new SimpleTF();  
39.     simpleTF.go();  
40. }  
41. }
```

Assuming that the user doesn't change the dimension of the frame, how many text fields will be displayed?

- ☐ A) 2
- ☐ B) 3
- ☐ C) The code doesn't compile successfully
- ☐ D) 1
- ☐ E) 0

Question 36:

Given the following code:

```
1. import java.awt.*;  
2. import java.awt.event.*;  
3.  
4. public class SimpleTF  
5. {
```

```
6.      Frame frame;
7.      TextField tf1, tf2, tf3;
8.
9.      SimpleTF()
10.     {
11.     }
12.
13.     public void go()
14.     {
15.         frame = new Frame();
16.         frame.setLayout(new FlowLayout());
17.         tf1 = new TextField(5);
18.         tf2 = new TextField(5);
19.         tf3 = new TextField(5);
20.         frame.add(tf1);
21.         frame.add(tf2);
22.         frame.setSize(400,400);
23.         frame.setVisible(true);
24.         frame.addWindowListener(new WindowAdapter()
25.             {
26.                 public void windowClosing(WindowEvent event)
27.                 {
28.                     System.exit(0);
29.                 }
30.             }
```

```
31.                                     );
32.         frame.add(tf3);
33.         frame.repaint();
34.     }
35.
36.     public static void main(String args[])
37.     {
38.         SimpleTF simpleTF = new SimpleTF();
39.         simpleTF.go();
40.     }
41. }
```

Assuming that the user doesn't change the dimension of the frame, how many text fields will be displayed?

- ☐ A) 2
- ☐ B) 3
- ☐ C) The code doesn't compile successfully
- ☐ D) 1
- ☐ E) 0



Question 37:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.
5.  public class ComponentsOrder
6.  {
7.      private Frame frame;
8.      private Button buttons[];
9.      private String str[] = {"AAA","AAABBB","AAABBBCCC"};
10.
11.     public void go()
12.     {
13.         frame = new Frame();
14.         buttons = new Button[3];
15.         frame.setLayout(null);
16.         for(int index=0; index<3; index++)
17.         {
18.             buttons[index] = new Button(str[index]);
19.             buttons[index].setBounds(50,50,100,100);
20.             frame.add(buttons[index]);
21.         }
```

```
22.         frame.addWindowListener(new WindowAdapter()
23.         {
24.             public void windowClosing(WindowEvent event)
25.             {
26.                 System.exit(0);
27.             }
28.
29.         }
30.                                     );
31.         frame.setSize(400,400);
32.         frame.setVisible(true);
33.     }
34.
35.     public static void main(String args[])
36.     {
37.         ComponentsOrder co = new ComponentsOrder();
38.         co.go();
39.     }
40. }
```

The output will be a button which its label is

- ( ) A) AAA
- ( ) B) BBB
- ( ) C) AAABBB

- ( ) D) AAABBBCCC
- ( ) E) AAAAAABBBAAABBBCCC

Question 38:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.
5.  public class ComponentsOrder
6.  {
7.      private Frame frame;
8.      private Button buttons[];
9.      private String str[] = {"AAA","AAABBB","AAABBBCCC"};
10.
11.     public void go()
12.     {
13.         frame = new Frame();
14.         buttons = new Button[3];
15.         frame.setLayout(null);
16.         for(int index=0; index<3; index++)
17.         {
```

```
18.         buttons[index] = new Button(str[index]);
19.         buttons[index].setBounds(50,50,100,100);
20.         frame.add(buttons[index],0);
21.     }
22.     frame.addWindowListener(new WindowAdapter()
23.     {
24.         public void windowClosing(WindowEvent event)
25.         {
26.             System.exit(0);
27.         }
28.
29.     }
30.                                     );
31.     frame.setSize(400,400);
32.     frame.setVisible(true);
33. }
34.
35. public static void main(String args[])
36. {
37.     ComponentsOrder co = new ComponentsOrder();
38.     co.go();
39. }
40. }
```

The output will be a button which its label is

- ( ) A) AAA
- ( ) B) BBB
- ( ) C) AAABBB
- ( ) D) AAABBBCCC
- ( ) E) AAAAAABBBAAABBBCCC

Question 39:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.
5.  public class ComponentsOrder
6.  {
7.      private Frame frame;
8.      private Button buttons[];
9.      private String str[] = {"AAA","AAABBB","AAABBBCCC"};
10.
11.     public void go()
12.     {
13.         frame = new Frame();
```

```
14.         buttons = new Button[3];
15.         frame.setLayout(null);
16.         for(int index=0; index<3; index++)
17.         {
18.             buttons[index] = new Button(str[index]);
19.             buttons[index].setBounds(50,50,100,100);
20.         }
21.         frame.addWindowListener(new WindowAdapter()
22.         {
23.             public void windowClosing(WindowEvent event)
24.             {
25.                 System.exit(0);
26.             }
27.
28.         }
29.         );
30.         frame.add(buttons[1],0);
31.         frame.add(buttons[2],2);
32.         frame.add(buttons[0],4);
33.         frame.setSize(400,400);
34.         frame.setVisible(true);
35.     }
36.
37.     public static void main(String args[])
38.     {
```

```
39.         ComponentsOrder co = new ComponentsOrder();
40.         co.go();
41.     }
42. }
```

The output will be a button which its label is

- ( ) A) AAA
- ( ) B) BBB
- ( ) C) AAABBB
- ( ) D) AAABBBCCC
- ( ) E) An IllegalArgumentException will be thrown.

Question 40:

Given the following code:

```
1.  import java.awt.*;
2.  import java.awt.event.*;
3.
4.
5.  public class ComponentsOrder
6.  {
7.      private Frame frame;
```

```
8.     private Button buttons[];
9.     private String str[] = {"AAA","AAABBB","AAABBBCCC"};
10.
11.    public void go()
12.    {
13.        frame = new Frame();
14.        buttons = new Button[3];
15.        frame.setLayout(null);
16.        for(int index=0; index<3; index++)
17.        {
18.            buttons[index] = new Button(str[index]);
19.            buttons[index].setBounds(50,50,100,100);
20.        }
21.        frame.addWindowListener(new WindowAdapter()
22.        {
23.            public void windowClosing(WindowEvent event)
24.            {
25.                System.exit(0);
26.            }
27.
28.        }
29.        );
30.        frame.add(buttons[1],0);
31.        frame.add(buttons[2],0);
32.        frame.add(buttons[0],0);
```



```
33.         frame.setSize(400,400);
34.         frame.setVisible(true);
35.     }
36.
37.     public static void main(String args[])
38.     {
39.         ComponentsOrder co = new ComponentsOrder();
40.         co.go();
41.     }
42. }
```

The output will be a button which its label is

- ( ) A) AAA
- ( ) B) BBB
- ( ) C) AAABBB
- ( ) D) AAABBBCCC
- ( ) E) An IllegalArgumentException will be thrown.

Question 41:

True or False:

AWT uses the validate method to cause a container to layout its subcomponents again after components it contains have been added to or modified.

- ☐ A) True
- ☐ B) False

Question 42:

True or False:

The paint() method is invoked whenever a component is made visible or resized and (sometimes) when the state of the component changes (clicking on a button, for instance, can result in a different screen representation, that represent a pressed button).

- ☐ A) True
- ☐ B) False

Question 43:

Which of the following is true:

- ☐ A) When a component is made visible or resized then the system causes the AWT-Event thread invokes the paint() method on that component.
- ☐ B) When the application calls the repaint() method on a certain component, the AWT-Event thread invokes a method called update(). The update() method clears the component background and then calls the paint() method.
- ☐ C) The background clearing that the update() method is the most common cause of flickering. Overriding the update() method is one of the common solutions to the flickering problem.
- ☐ D) When overriding the update() method, in order to solve the flickering problem, the best thing to do is declaring two new methods: paintForeground(), a method that paints the foreground, and paintBackground(), a method that paints the background. Then, within the update() method, we should place a call to the paintForeground() method and within the paint() method we should place a call to the paintForeground() method and a call to the paintBackground() method right after it.