학번 :

이름 :

JAVA Language

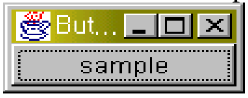
Final Exam, Fall 2012

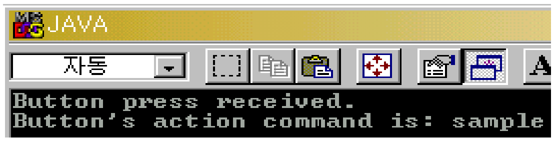
1. Fill the blank by reffering below figures.(7)

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

public class ButtonDemo implements ActionListener {
    private Frame f;
    public ButtonDemo () {
        Fill this blank.
        WindowDestroyer listener = new WindowDestroyer();
        f.addWindowListener(listener);
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        System.out.println("Button press received.");
        System.out.println("Button's action command is: " + e.getActionCommand());
    }
    public static void main(String args[]) {
        ButtonDemo bd = new ButtonDemo();
    }
}
```

Frame “Button Example”





```
Button press received.
Button's action command is: sample
```

2. Describe the six steps of “XOR Mode” to move a figure in the java program. (5)

3. Briefly explain about the following questions about “Multithreading”.(8)
   (a) What is the difference of “Multiple process” and “Multiple thread” ?
   (b) Describe about Thread Priority.
   (c) Draw the diagram of 4 thread states and provide method names in the arc for state change.
   (d) Briefly explain run(), start(), and sleep() method.

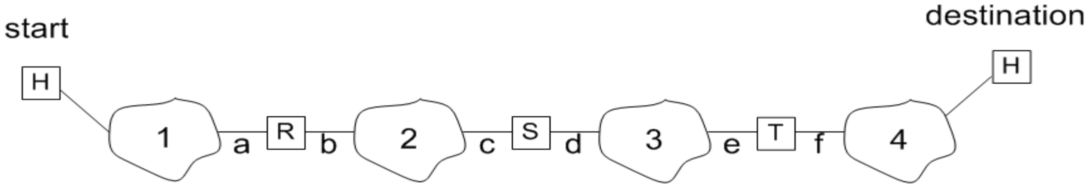
4. Write down the two differences between vector and array in JAVA.(4)

5. Answer the questions.
   (a) This sentence is a definition of Internet, fill the blanks and complete this definitio. (5)

Internet: A collection of (a.) switching networks connected by (b.) using (c.) protocols to form a single (d.) network.

   (b) Draw the c. in Question (a) network architectural model and explain the features of each layer. (5)

6. This network use the table driven routing method, fill the each routing table. (10)



( )

( )

( )

7. Fill the blanks. (5)

Converting Applications to Applets

Here are the specific steps for converting an application to an applet.

1. Make an HTML page with an APPLET tag.
2. Eliminate the (a.) method in the application. Usually (a.) contains code to make a new frame object. With applet objects, that is automatically taken care of by the browser, since it makes an object of the class specified in the APPLET tag. Also, (a.) usually sets the frame size. For applets, this is done with the (b.) and (c.) fields of the APPLET tag in the actual HTML file.
3. Derive the class from (d. **)**, not from Frame.
4. Replace the constructor with a method called (e.). When the browser creates an object of the applet class, it calls the (e.) method.
5. setSize(), setTitle(), and FlowLayout() methods are not valid in applet.
6. Menu cannot be used in Applet.

8. Replace the following applet & HTML code into JAVA code. (10)

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

public class apYeJe extends Applet {
        public void init() {
            setBackground(Color.red);
            }

public void paint(Graphics g) {
            g.setFont(new Font("바탕체", Font.BOLD, 30));
            g.drawString( "자바  애플릿   JAVA Applet", 10,100);
            }
  }
```

```
<APPLET CODE="apYeJe.class"    WIDTH=400 HEIGHT=300 IGNORE="">
</APPLET>
```

9. Fill the blanks by reffering execution result.(10)

```
class PanMae    {
    private String pmyung;
    private int sryang;
    private int danga;
    public PanMae daum;

    public PanMae(String pm, int sr, int dg) {
        pmyung = pm;
        sryang = sr;
        danga    = dg;
    }

    public void pyoSi() {
        System.out.println(pmyung + '\t' + sryang + '\t' + danga);
    }
}

class dsLinkedlist1 {
    public static void main(String[] args) {
        PanMae pm1 =    new PanMae("1 SaGwa", 5, 1000);
        PanMae pm2 =    new PanMae("2 Bae",      3, 2000);
        PanMae pm3 =    new PanMae("3 PoDo",    7,    500);

                                        (1)

        PanMae p = pm1;

        while (p != null) {
            p.pyoSi();
            p = p.daum;
        }

        PanMae pm4 =    new PanMae("4 Gam",    9,    200);

                                        (2)

        System.out.println("\n<Node SaIbHoo>\n");
        p = pm1;
        while (p != null) {
            p.pyoSi();
            p = p.daum;
        }
    }
}
```

Execution Result

```
1 SaGwa    5      1000
3 PoDo      7      500
2 Bae         3      2000

<Node SaIbHoo>

1 SaGwa      5    1000
3 PoDo        7    500
4 Gam         9    200
2 Bae          3    2000
```

10. Fill the blanks by referring to below table. (16)

- Print “Button Pressed” in command window, when you press the button.
- Change background color sequencially(repeat red-green-blue), when mouse pointer moves in window.

```
import java.awt.*;
①

public class Exam extends  ②  implements  ③  {
        private int index;
        Colors colors[] = {Color.red,
                        Color.green, Color.blue};
        public Exam (String str) {
                super(str);
                Button b = new Button(“bt”);
                SetLayout(new FlowLayout());
                add(b);
                ④;
                b.⑤;
                index = 0;
                setSize(100, 100);
                setVisible(true);
        }

        public void paint(Graphics g) {
                setBackground(  ⑥  );
        }
        public void  ⑦  {
                System.out.println(“Button pressed”);
        }
        ⑧
        public static void main(String args[]) {
                Exam exm = new Exam("Exam");
        }
}
```

Category	Interface Name	Methods
Action	ActionListener	actionPerformed(ActionEvent)
Item	ItemListener	itemStateChanged(ItemEvent)
Mouse motion	MouseMotionListener	mouseDragged(MouseEvent)
		mouseMoved(MouseEvent)
Mouse button	MouseListener	mousePressed(MouseEvent)
		mouseReleased(MouseEvent)
		mouseEntered(MouseEvent)
		mouseExited(MouseEvent)
		mouseClicked(MouseEvent)
Key	KeyListener	keyPressed(KeyEvent)
		keyReleased(KeyEvent)
		keyTyped(KeyEvent)

11. Fill the blanks. (15)

Server (IP: 127.0.0.1, Port:1111)

```
        import java.io.*;
        import java.net.*;
        public class Server{
                public static void main (String args[]) throws Exception           {
                        ServerSocket serverSocket = null;
                        (1) = null;
                        String reply = "Hello! I'm Server";
                        OutputStream outs;
                        DataOutputStream douts;

                        (2) (1111);

                        while (true)
                        {
                                newSocket = serverSocket.accept();

                                System.out.println("Client is connected.");
                                outs = (3)
                                douts = new DataOutputStream ( (4) );
                                douts.writeUTF(reply);

                                douts.close();
                                (5)
                        }
                }
}
```

Client

```
        import java.net.*;
        import java.io.*;

        public class Client {
                public static void main(String args[]) throws IOException {
                        (6) newSocket;
                        InputStream is;
                        DataInputStream dis;
                        String receive = null;
                        (7) ("127.0.0.1", 1111);

                         is = (8)
                        dis = new DataInputStream ( (9) );
                        receive = new String (dis.readUTF());
                        System.out.println (receive);
                        dis.close();
                        (10)
                }
}
```