(11) What are applets? Explain its architecture with an example program.

→ Applets are small applications that are accessed on an Internet server, transported over the Internet automatically installed, and run as part of document

Applet Architecture.

- an applet is a window based program
- First, applets are event driven, an applet remains until an event occurs.
- Second, the user initiates interaction with an applet.
- User interacts with applets the way he wants. These interactions are sent to the applet as event to which the applet must respond.

Example Program :-

```
import java.awt.*;
import java.applet.*;
/* <applet code = " Appletshel " width = 300 height=10
   </applet> */
public class Appletshel extends Applet {
    public void init(){
        // initialisation
    }
```

```
public void start () {
    // start or resume execution
}

public void paint (Graphics g) {
    // redisplay contents of window
}
}.
```

(12) How are parameters passed to applet. Give example

→ The Applet tag in HTML allows us to pass parameters to the applet.
→ To return a parameter, we getParameter () method
→ It returns the value of specified parameter in the form of string object.
→ For numeric and Boolean values, we will need to convert their string representation into their internal formats.

→ Example program

```
import java.awt.*;
import java.applet.*;
/* <applet code = "ParamDemo" width = 300 height = 80
   <param name = fontName value = Courier>
   <param name = fontsize value = 14>
```

```
<param name = leading value = 2)
<param name = antialiased value = true)
</applet> */

public class ParamDemo extends Applet {
        String fontName;
        int fontSize;
        float leading;
        boolean active;

    public void start () {
        String param;
        fontName = getParameter ("FontName");
        if (fontName == NULL)
            fontName = "Not found";
        param = getParameter ("fontSize");
    try {
        if (param != NULL)
            fontSize = Integer.parseInt(param);
        else
            fontSize = 0;
    }
    catch (NumberFormatException e) {
            fontSize = -1;
    }
    param = getParameter ("leading");
    try {
```

```
if (param != null)
    leading = Float.parseFloat(param);
else
    leading = 0;
}
catch (NumberFormatException e) {
    leading = -1;
}

param = getParameter("accountEnabled");
if (param != null)
    active = Boolean.parseBoolean(param);
}

public void paint(Graphics g) {
    g.drawString("FontName" + FontName, 0, 10);
    g.drawString("FontSize" + FontSize, 0, 26);
    g.drawString("leading" + leading, 0, 42);
    g.drawString("AccountActive" + active, 0, 58);
}
}
```

(13) ~~What~~ write an applet program to display a banner which scroll a message horizontally.

→
```
import java.awt.*;
import java.applet.*;
/*<appletcode = "SampleBanner" width=300, height=30>
```

```
</applet> */
public class SampleBanner extends Applet implements
Runnable {
  String msg = " A Simple Banner ";
  Thread t = null;
  int state;
  boolean stopFlag;
  public void init() {
    setBackground (Color.cyan);
    setForeground (Color.red);
  }

  public void start () {
    t = new Thread (this);
    stopFlag = false;
    t.start ();
  }
    public void run () {
      char ch;
    for ( ; ; ) {

try {
        repaint ();
        Thread.sleep (250);
        ch = msg.charAt (0);
      msg = msg.substring (msg.length ());
      msg += ch;
```

```
if( stop flag )
    break;
} catch ( InterruptedException e) { }
}
}

    public void stop () {
        stop flag = true;
        t = null;
    }

public void paint (Graphics g) {
    g.drawString (msg, 50, 30);
    }
}
```

(14) List and explain predefined streams in Java.

→ java.lang package defines a class called System which encapsulates several aspects of the run-time environment.

→ System.out refers to the standard output stream by default, this is the console.

→ System.in refers to standard input, which is the keyboard by default.

→ System.err refer to the standard error stream which also is the console by default.

→ These streams may be redirected to any compatible I/O devices.

→ System.in is an object of type InputStream.
→ System.out and System.err are objects of type PrintStream.

(16). Write a program to ~~list~~ list the contents of a specified directory.

```
→ import java.io.*;
public class OnlyExt implements FilenameFilter {
    String ext;
    public OnlyExt(String ext) {
        this.ext = "." + ext;
    }
    public boolean accept(File dir, String name)
    {
        return name.endsWith(ext); }
    }
// Directory of HTML files.
import java.io;
class DirList {
    public static void main(String args[]) {
        String dirname = "/java";
        File f = new File(dirname);
        FilenameFilter only = new OnlyExt("html");
        String s[] = f.list(only);
        for (int i=0; i<s.length; i++) {
            System.out.println(s[i]);
        } } }
```

15) Write program to copy one file to another using byte oriented stream class from java.io. package.

```
import java.io.*;
public class ByteCopyFile.
  {
    public static void main (String args[]) throws
                                        IO Exception
    {
        FileInputStream fin = null;
        FileOutputStream fout = null;
        try.
      {
          fin = new FileInputStream ("/home/student/
                                  1616101s7/week9/
                                  TextFile");
          fout = new FileOutput ("/home/student/1616101s4/
                                  week9/TextFileCopy");
          int filesize = fin.available();
// returns the remaining number of bytes in the file
   that can be read.
   S.O.P ("File size in bytes:" + filesize);
   int ch;
   while ((ch= fin.read()) != -1)
     {
       fout.write(ch);
     }
   S.O.P ("File copied successfully");
}
```

```
finally
{
    fin.close();
    fout.close();
}
}
}.
```

17. How are swings better than AWT ?

Swings	AWT.
(i) They are platform - independent.	They are platform dependent.
(ii) Swings component are light-weight.	AWT components are heavy weight.
(iii) Supports pluggable look and feel.	doesn't suppor pluggable look and feel.
(iv) provides more powerful components such as table, list, scrollpanes, etc.	provides less components than swings.
(v) It supports MVC.	It doesn't follow MVC.

18. Write a swing application to display a message in a pane.

→
```
import javax.swing.*;
import java.awt.*;
public class JavaExampleGridLayout extends JFrame
{
    Private JButton BtnOne = new JButton ("BtnOne");
    private JButton BtnTwo = new JButton ("BtnTwo");
    private JButton BtnThree = new JButton ("BtnThree");
    private JButton BtnFour = new JButton ("BtnFour");
    private JButton BtnFive = new JButton ("BtnFive");
    Private GridLayout layout = new GridLayout (3,2,5,5);
    public JavaExample (GridLayout ()
    {
        setSize (200,200);
        Set layout (layout);
        add (BtnOne);
        add (BtnTwo);
        add (BtnThree);
        add (BtnFour);
        add (BtnFive);
    } setVisible (true);

p.s.v.m (String [] args)
    {
        JavaExample GridLayout frame = new JavaExample
                                        GridLayout ();
    }
}.
```