**1. SYNOPSIS**

This application "**My java platform"** is developed using JAVA. It is basically an **editor** with some additional options. This editor is designed in such a way to fulfill the complete needs of the java Integrated Development Environment or java development using **SUN JDK** under windows(all versions), Compilation and execution can be done in command prompt only. This application helps to do compilation and execution in editor itself. So that user can identify the **warning(s) and error(s)** in the editor itself. This editor reduces the user's time. It is easy to compile and run the programs. This editor is more flexible and user friendly.

This application is developed in java language using the following concepts **AWT, FRAME, SWING, and IO.**

**PROJECT DESCRIPTION**

This editor is used to write any java program easily. It is helpful to compile and run the Java program in the same window. This JPDK has two-text area. One is used for the code development. And the order for debugging. The following are the menus in the My java Platform -java editor:

* File
* Edit
* Search
* Format
* Run
* Window
* Help

**File Menu**

The file menu deals with file handling and it consists of the following menu items.

**New**

To open a new file. When this menu item is clicked a new file with the name “untitled” is opened.

**Open**

To open the existing files. When this menu item is clicked the open dialog box is displayed which provides facility of choosing any file and when a file is selected its content gets displayed in the Text area.

Save

To save the current file. If one wants to save the untitled file, the save dialog box gets displayed. If the file is an existing file then the Dialog box does not appear. The contents of the file are just updated.

## Save As

To save the existing file with a new name. When the save as menu item is clicked a save as dialog box is opened and we can save the file with the new name.

**Close**

To close the file that has been opened.

Quit

To exit from the application. When this menu item is clicked the exit operation takes place.

**Edit Menu**

The Edit menu deals with editing operations and it consists of the following menu items.

**Undo**

Repeat Last action.

**Clear**

Clears the screen by deleting the contents of the text area.

Cut

To copy the selected text to the buffer and delete from the existing area.

Copy

To copy the selected text to the buffer.

**Paste**

To paste the copied text or the text that was cut wherever needed.

Select all

To select all the text in the text area for editing operations.

**Search Menu**

This menu item is used to handle the search operations.

Find

To search a particular word throughout the file. When the user clicks this option, the Find Dialog box is displayed on the screen. The user gives the text to be found in the text field and enters o.k. If the word is present in the file it is highlighted else a message box gets displayed with the message “Not Found”. The find Dialog also provides the options match case, whole words by means of checkbox**.**

Find Next

Find Next search for the next location of the word previously searched using the find option.

**Format:**

The Format menu provides customized setting to the user. It has two sub menus

* Font
* Color

**Font Selection**:

When the user wants to set any font of his choice he selects this option. It provides him option for setting font, font style and size for the text. It also has a preview facility.

**Color**:

By clicking the color menu the user can select any color of his choices from the color palette. The color palette provides the color using the primary colors. This dialog box support both background as well as foreground colors using Radio buttons. User can select one or them.

**RUN MENU**

**Compile:**

Compiles the file that is currently opened in the text area and displays the errors in the error text area. If there are no errors then the message “Good!!.There is no error in your Program.” is displayed.

**Application:**

When the user selects the Application option then the editor searches for the class file of the current file to run. If the class file is available then the file is run successfully and output is obtained else failure is reported.

**Applet**

Invokes the applet viewer with the file that is currently open. This option is used run the Applet programs and Swing programs.

**Iexplorer**

To open the current html file in the web browser.

##### ServletRunner

This option is used to run the generic Servlets programs**.**

#### Oracle

This option is used to connect with Oracle application.

#### Ms-Access

#### This option is used to connect with Ms-Access application .

**Window Menu**

**Output window**

This option is used to enable the output window

**FullScreen**

This option is used to disable the output window and display the textarea with entire screen

**Help:**

This module supports high profile quick help. This help explains all the tags with their syntax and suitable examples to facilitate the user. When the user needs to see sample programs he has to click the example link.

# 2. System Configuration

###### 2.1Hardware Requirements

Processor : intel i4

Hard Disk : 10GB

RAM : 2 GB

Monitor : Samsung 15 inches

Mouse : Logitech Scroll Mouse

Keyboard : Samsung 101 keys

### 2.2Software Requirements

Java Development Toolkit 1.6 with

* Swings
* Java Compiler
* Java Interpreter
* Applet viewer.

**Operating System** : developed on windows 7

compatible with all OS

**Windows 7**

**Windows 7** is an operating system produced by Microsoft for use on personal computers, including home and business desktops, laptops, netbooks, tablet PCs, and media center PCs. It was released to manufacturing on July 22, 2009, and became generally available retail worldwide on October 22, 2009, less than three years after the release of its predecessor, Windows Vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time. Windows 7 is succeeded by Windows 8.

Unlike Windows Vista's many new features, Windows 7 was an incremental upgrade designed to work with Vista-compatible applications and hardware. Presentations given by Microsoft in 2008 focused on multi-touch support, an updated Windows shell with a new taskbar, referred to internally as the *Superbar*, a home networking system called HomeGroup, and performance improvements. Some standard applications that have been included with prior releases of Microsoft Windows, including Windows Calendar, Windows Mail, Windows Movie Maker, and Windows Photo Gallery, are not included in Windows 7; most are instead offered separately at no charge as part of the Windows Essentials suite.

**3. SOFTWARE SPECIFICATION**

**JAVA:**

**JAVA is a machine independent and operating system independent language. A program written in JAVA can be run on any machine or in any platform provided the interpreter of that particular platform.**

**ADVANTAGES:**

* Simple and powerful.
* safe.
* Robust.
* Interactive.
* Neutral.
* Interpreted and high performance.

**JAVA FEATURES:**

**JAVA IS SIMPLE**

Even though Java is object oriented, there are some management differences in C++ and JAVA. In Java there are no pointers, no multiple inheritance, no go to, an operator overloading. The Java language has an embedded auto garbage collection mechanism. The Garbage collector simplifies Java programming by automatically freeing memory area when it is not required. Java programs are therefore simple, comparative bug-free Memory.

**JAVA IS OBJECT - ORIENTED**

Java is an Object – Oriented language. It defines data as objects with methods that support the objects. Java does not support multiple inheritance. It support the objects. Java does not support multiple inheritance. It supports inheritance where one class can inherit from only one other class; on the other hand Java supports abstract classes using which, programmers can stimulate multiple inheritance. The abstract base class for all Java classes is a class called an object. All the code used for Java is divided into classes. Behaviors can be inherited from one class to the next.

**JAVA IS DISTRIBUTED**

Java like any other Client/Server application, Java shares the workload of a data processing and information – sharing. Java can access data across the Web, using URLs (Universal Resource Locators), as simply as it would access data on a local system. Java also has a library of routines using which support for Internet protocols like HTTP and FTP, can be easily built into the applet.

**JAVA HAS INTERPRETER AND COMPILER**

Java applet is written and compiled by byte-code. This Byte-code is binary and platform-independent. When this applet has to be executed, it is fetched to the local system; in other words it travels to the local system, where it is interpreted by the browser. Only twenty percent of Java program is interpreted, but a twenty-percent is very crucial. The reason behind Java’s security is it can run on multiple platforms and final steps of compilation are done locally.

**JAVA IS ROBUST**

Robust means reliable. A reliable language will allow you to write programs that do not crash when least expected and will also be bug free. Since Java is a strongly typed language, it does a comprehensive compile-time check on the program, which means that any bugs can be identified at compile-time. Java program do not have access to all of your computer’s memory, which means they cannot change a value that is not to be changed, unlike traditional applications that have complete access to all or any part of the system memory, and hence cannot cause your system to crash.

**JAVA IS SECURE**

A program traveling across the Internet onto your machine is not visible, could possibly carrying a virus along with it. This is a possibility in Java programs. But due to the strong type checking done by Java on the user’s local machine, any changes to the program are tagged as an error. The program will not be executed. This means that Java programs cannot be infected by a virus.

**JAVA IS ARCHITECTURE NEUTRAL**

Due to the byte-code compilation process, and interpretation by the browser, Java can work with variety of hardware and operating systems. The only requirement is that the system should have a Java-enabled Internet browser.

**JAVA IS PORTABLE**

The Java system was built to be portable. The Java compiler is written using the Java language and the run-time environment; the interpreter is Written in ANSI C. The Java interpreter can execute the byte-code directly on any machine onto which the interpreter can execute the byte-code directly on any machine onto which the interpreter has been installed.

**JAVA IS HIGH PERFORMENCED**

High performance does not mean that Java is faster than another language; Instead, Java is comparable in speed to other languages like C or C++. For a simple application, the difference in speed is hardly distinguishable.

#### JAVA IS MULTITHREADED

Multithreading is the ability of an application to perform multiple tasks at the same time; for example, if you were playing a combat and another thread could be handling the graphics on the screen. In a single-threaded application, only one task can be performed at a given point of time.

#### JAVA IS DYNAMIC

Unlike C or C++, Java is a language that has been designed to adapt to evolving environments. In case of C or C++, you can become dependent on a second party by using the class libraries provided by them .If the second party alters library , you will have to respond by including , recompiling , possibly redistributing , and your own software.

**SWING:**

Swing is a set of classes that provides more powerful and flexible components more than are possible with the AWT .In addition to familiar components, such as Buttons, Check boxes, and Labels. Swing supplies several additions, including Tabbed panes, Scroll panes, Trees and Tables. Even familiar components such as buttons have more capabilities in Swing. For example, a button may have both an image and a text string associated with it. Also, the image can be changed as the state of the Button changes.

Swing components are platform independent so the term lightweight is used. The Swing-related classes are contained in javax.swing and its subpackages, such as javax.swing.tree.

**JAVA INPUT AND OUTPUT**:

The user interacts with a Java program with the help of various input and output facilities provided by Java. All the input output operations are handled differently by Java. They are called as streams and each type of operation has its own class.

In Java an object from which a sequence of bytes can be read is called an Input Stream and an object from which a sequence of bytes can be written is called as Output Stream. The “ Input Stream “ and the “ Output Stream “ classes are defined and abstract classes from which all other Stream classes are derived.

Some of the input and output Streams are

* **File Input Stream**
* **File Output Stream**
* **Byte Array Input Stream**
* **Byte Array Output Stream**
* **Data Input Stream**
* **Data Output Stream**

**INTERNET OVERVIEW:**

INTERNET – The world’s largest computer network. It is precisely been called as network of networks. The various separate networks in the world are inter-inked to form one entity that is communicated through a protocol usually TCP/IP (Transmission Control Protocol/Internet Protocol) The Internet is known by many terms such as CYBERSPACE.

**JAVA AWT—POWERFUL TOOL:**

AWT stands for Abstract Windowing Toolkit. It has all libraries required for GUI programming Java AWT is a package that embeds all GUI classes that can be used for programming. For any GUI program this package is first imported. Some of the tools that are available in this package includes.

* Button
* Scroll Bars
* List Boxes
* Choice Boxes
* Windows

AWT also includes methods for handling events graphical operations.

**NETWORK CONCEPTS:**

A network is a collection of computers and other devices that can send data to and receive data from each other, more or less in real time. A network is normally connected by wires, and the bits of data are turned into electromagnetic waves that move through the wires. However, wireless networks that transmit data through infrared light or microwaves are beginning to appear; and many long-distance transmissions are mow carried over fiber-optic cables that send visible light through glass filaments. each machine on a network is called a node. Nodes that are fully functional computers are also called hosts.

The address that organization is allowed to choose for its computers are assigned to it by the organization's INTERNET SERVICE PROVIDER(ISP). ISP'S get their Internet Protocol addresses from one of three regional Internet Registries ,which are in turn assigned Internet Protocol addresses by the Internet Assigned Numbers Authority(IANA).

A protocal is a precise set of rules defining how computers communicate:

The format of addresses, how data is split into packets etc. There are many different protocols defining different aspects of network communication.

Transmission Control Protocol (TCP)

Most Internet applications use TCP t o implement the transport layer. TCP provides a reliable connection oriented continues stream protocol. The implications of these characteristics are;

Realiable When TCP segments are lost of corrupted ,the TCP implementation will detect and retransmit necessary segments.

Connection Oriented TCP sets up a connection with a remote system by transmitting control transmission often known as handshake, before beginning a communication. At the end of the connect, a similar closing handshake ends the transmission.

An important addressing schemes that TCP defines is t he port. ports separate various TCP communications streams that are running concurrently on the same system. For server applications ,which wait for TCP clients to initiate contact ,a specific port can be established from where communications will originate .These concepts come together in a programming abstract ion known as sockets.

TCP Client /Server based Applications

* The process of designing our protocol implementation involves the following steps:
* Client connects to server, waiting for a client to communicate.
* Server and the client exchange informations, depending upon the programming done, on each part.
* Terminate the connection and quit the program(if needed).

**The 'InetAddress' object**

This object is one that stores an IP address of a remote system.

**The Internet hosts are identified in one of the following ways:**

The port number of FTP is 21. When data is send between two computers, the network protocol use this numeric address for determining where to send the data. Host names are created for convenience. It has no public constructor methods , but does have a number of static methods that return instances of InetAddress.

**The 'SOCKET' class**

Sockets are programming abstraction that isolates the code from the low level implementations of the TCP/IP protocol stack. It enables to quickly develop the own custom client /server applications. A TCP Socket uses the TCP protocol the inheriting the behavior of that transport protocol. The following Information's are necessary to create a TCP Socket .

* Local system's IP address
* The TCP port number , use by the local application
* The remote system's IP address.
* The TCP port number to which the remote application response

These Sockets are often used in client/server applications. A centralized service waits for varies remote machines to request specific resources, handling each request as it arrives. For clients to know how to communicate with the server, standard application protocols a re assigned well known ports.

**The two classes of the Socket are**

1. **java.net.Socket**
2. **java.net.ServerSocket.**

##### The 'ServerSocket' class

This wonderful class is used to create server side applications.

It allows to bind a port and wait for clients to connect, setting up a complete Socket object at that time .

User Datagram protocol(UDP )

UDP is a low- overhead alternative to TCP for host-to-host communications. In contrast to TCP,UDP has the following features.

* Unreliable UDP has no mechanism for detecting errors, nor retranslating lost or corrupted information.
* Connectionless UDP does not negotiate a connection before transmitting data. Information is sent with the assumption that the recipient will be listing.
* Message - Oriented UDP enables applications to send self -contained messages within UDP datagrams, the unit of UDP transmission. The application must package all information within individual datagrams.

**4.SYSTEM DESIGN**

**4.1 SCREEN DESIGN**

My java platform is a user friendly editor,with a good GUI. The various options are arranged into the respective menus and submenus such that the java ide is quite simple to use , especially for the beginners.

The screen shot is shown here:

**5. SYSTEM TESTING**

**TESTING STRATEGY :**

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is

achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

**The project of Java program development Toolkit was tested and it is found to be work accurately. The physical design is found to work and system testing in implementation arrives at the conclusion that the system is prefect in all aspects.**

**Each menu and menu item was tested individually with its test. Tested menus and menu items are found to link perfectly. The system was tested in the actual operating environment and outputs generated are in tune with the requirements**

**6. IMPLEMENTATION**

**6.1 SAMPLE CODE**

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.datatransfer.\*;

import java.util.\*;

class fra extends Frame implements ActionListener,ItemListener

{

TextArea ta1,ta2;

CheckboxMenuItem m61,m62;

TextField t1,t2,t3;

Toolkit tk;

Dialog d1;

String cs,sf;

Choice c,c1,c2,c3;

Scrollbar s1,s2,s3;

int sso=0;

static int r,g,b,r1,g1,b1,ip;

fra()

{

super("JPDK 1.0 - Untitled");

setSize(800,600);

setLayout(null);

setBackground(Color.lightGray);

menusetup();

ctrlsetup();

setVisible(true);

}

void ctrlsetup()

{

tk=getToolkit();

ta1=new TextArea(10,10);

ta1.setBackground(Color.gray);

ta2=new TextArea(10,10);

ta2.setBackground(Color.gray);

ta1.setBounds(5,43,790,360);

ta2.setBounds(5,425,790,145);

ta2.setEditable(false);

add(ta1);

add(ta2);

}

void menusetup()

{

try{

MenuBar mb=new MenuBar();

Menu m1=new Menu("File");

Menu m2=new Menu("Edit");

Menu m3=new Menu("Search");

Menu m4=new Menu("Format");

Menu m5=new Menu("Run");

Menu m6=new Menu("Window");

Menu m7=new Menu("Help");

MenuItem m11=new MenuItem("New");

MenuItem m14=new MenuItem("Open");

MenuItem m12=new MenuItem("Save");

MenuItem m13=new MenuItem("Save As");

MenuItem m15=new MenuItem("Close");

MenuItem m16=new MenuItem("Quit");

MenuItem m21=new MenuItem("Undo");

MenuItem m22=new MenuItem("Cut");

MenuItem m23=new MenuItem("Copy");

MenuItem m24=new MenuItem("Paste");

MenuItem m25=new MenuItem("Clear");

MenuItem m26=new MenuItem("Select All");

MenuItem m31=new MenuItem("Find");

MenuItem m32=new MenuItem("Find Next");

MenuItem m41=new MenuItem("Font");

Menu m42=new Menu("Colors");

MenuItem m421=new MenuItem("ForeColor");

MenuItem m422=new MenuItem("BackColor");

m42.add(m421);

m42.add(m422);

MenuItem m51=new MenuItem("Compile");

MenuItem m52=new MenuItem("Application");

MenuItem m53=new MenuItem("Applet");

MenuItem m54=new MenuItem("Servlet Runner");

MenuItem m55=new MenuItem("IExplorer");

MenuItem m56=new MenuItem("Ms-Access");

MenuItem m57=new MenuItem("Oracle8.0");

m61=new CheckboxMenuItem("Full Screen");

m62=new CheckboxMenuItem("Output Window",true);

m61.addItemListener(this);

m62.addItemListener(this);

MenuItem m71=new MenuItem("Help");

MenuItem m72=new MenuItem("About JPDK 1.0");

m1.add(m11);

m1.add(m14);

m1.add(m12);

m1.add(m13);

m1.add(m15);

m1.addSeparator();

m1.add(m16);

m2.add(m21);

m2.addSeparator();

m2.add(m22);

m2.add(m23);

m2.add(m24);

m2.add(m25);

m2.addSeparator();

m2.add(m26);

m3.add(m31);

m3.add(m32);

m4.add(m41);

m4.add(m42);

m5.add(m51);

m5.addSeparator();

m5.add(m52);

m5.add(m53);

m5.add(m54);

m5.add(m55);

m5.addSeparator();

m5.add(m56);

m5.add(m57);

m6.add(m61);

m6.add(m62);

m7.add(m71);

m7.addSeparator();

m7.add(m72);

mb.add(m1);

mb.add(m2);

mb.add(m3);

mb.add(m4);

mb.add(m5);

mb.add(m6);

mb.add(m7);

setMenuBar(mb);

m1.addActionListener(this);

m2.addActionListener(this);

m3.addActionListener(this);

m4.addActionListener(this);

m5.addActionListener(this);

m6.addActionListener(this);

m7.addActionListener(this);

m42.addActionListener(this);

}

catch(Exception e){}

}

public void itemStateChanged(ItemEvent e)

{

if(m61.getState())

{

ta1.setBounds(5,43,790,545);

m62.setState(false);

validate();

}

else

{

ta1.setBounds(5,43,790,360);

validate();

}

if(m62.getState())

{

ta2.setVisible(true);

validate();

}

else

{

ta2.setVisible(false);

validate();

}

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

if (s=="New")

{

ta1.setText("");

d1=new Dialog(this,"New ... ",true);

d1.setLocation(300,300);

d1.setSize(250,90);

d1.setLayout(new FlowLayout());

d1.add(new Label("Select the File Type"));

c=new Choice();

c.add("Application");

c.add("Applet");

c.add("Frame");

c.add("Bean");

c.add("Swing");

c.add("JDBC");

c.add("Servlet");

c.add("HTML");

d1.add(c);

Button b1=new Button("Start");

Button b2=new Button("Cancel");

d1.add(b1);

d1.add(b2);

b1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent v)

{

String xx=c.getSelectedItem();

String title=getTitle().substring(11,getTitle().length()-5);

if(s=="Quit")

System.exit(0);

if (s=="Undo")

{

ta1.setText(cs);

}

if(s=="Cut")

{

String s1=ta1.getSelectedText();

String b=ta1.getText();

cs=ta1.getText();

b=ta1.getText().substring(0,ta1.getCaretPosition());

b=b+ta1.getText().substring(ta1.getCaretPosition()+ta1.getSelectedText().length(),ta1.getText().length());

ta1.setText(b);

StringSelection ss=new StringSelection(s1);

Clipboard cl=tk.getSystemClipboard();

cl.setContents(ss,ss);

}

if (s=="Copy")

{

String s1=ta1.getSelectedText();

StringSelection ss=new StringSelection(s1);

Clipboard cl=tk.getSystemClipboard();

cl.setContents(ss,ss);

}

if (s=="Paste")

{

String dat="";

Clipboard cl=tk.getSystemClipboard();

Transferable con=cl.getContents(this);

try{

dat=(String) con.getTransferData(DataFlavor.stringFlavor);

}

catch(Exception ex){}

StringBuffer tt=new StringBuffer(ta1.getText());

tt.insert(ta1.getCaretPosition(),dat);

ta1.setText(tt.toString());

}

if(s=="Clear")

{

String s1=ta1.getSelectedText();

String b=ta1.getText();

cs =ta1.getText();

b=ta1.getText().substring(0,ta1.getCaretPosition());

b=b+ta1.getText().substring(ta1.getCaretPosition()+ta1.getSelectedText().length(),ta1.getText().length());

ta1.setText(b);

}

if (s=="Select All")

{

ta1.select(0,ta1.getText().length());

}

if (s=="Compile")

{

try{

String r = getTitle().substring(11,getTitle().length());

Runtime rt =Runtime.getRuntime();

Process p =rt.exec("javac "+r);

p.waitFor();

int i =p.exitValue();

InputStream ip =p.getErrorStream();

int a =ip.available();

ta2.setText("");

ta2.setText("Compiling ...." +"\n\n");

if(a==0)

ta2.setText(ta2.getText()+"Good !!. There is no error in your Program\n");

else

{

int c;

String ts=ta2.getText();

while((c =ip.read()) != -1)

{

ts=ts+(char) c;

}

ta2.setText(ts);

}

}catch(Exception ex){}

}

if(s =="Application")

{

String r = getTitle().substring(11,getTitle().length()-5);

try

{

Runtime t =Runtime.getRuntime();

Process p=t.exec("java "+r);

p.waitFor();

int i =p.exitValue();

InputStream ip =p.getInputStream();

int a =ip.available();

ta2.setText("Output of " + r + ".java ...... \n\n");

String ts=ta2.getText();

if(a!=0)

{

int c;

while((c =ip.read()) != -1)

{

ts=ts+(char) c;

}

ta2.setText(ts);

}

else

ta2.setText("Program Successfully Executed");

}catch(Exception eq){}

}

if(s =="Applet")

{

String r = getTitle().substring(11,getTitle().length());

try

{

Runtime t =Runtime.getRuntime();

Process p=t.exec("appletviewer "+r);

p.waitFor();

int i =p.exitValue();

InputStream ip =p.getInputStream();

int a =ip.available();

ta2.setText("Applet Executed");

String ts=ta2.getText();

int c;

while((c =ip.read()) != -1)

{

ts=ts+(char) c;

}

ta2.setText(ts);

}catch(Exception eq){}

}

if(s =="IExplorer")

{

try{

Runtime t =Runtime.getRuntime();

Process p=t.exec("start iexplore");

p.waitFor();

int i =p.exitValue();

}catch(Exception ip){}

}

public class x1 extends fra

{

public static void main(String ar[])

{

fra f=new fra();

f.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

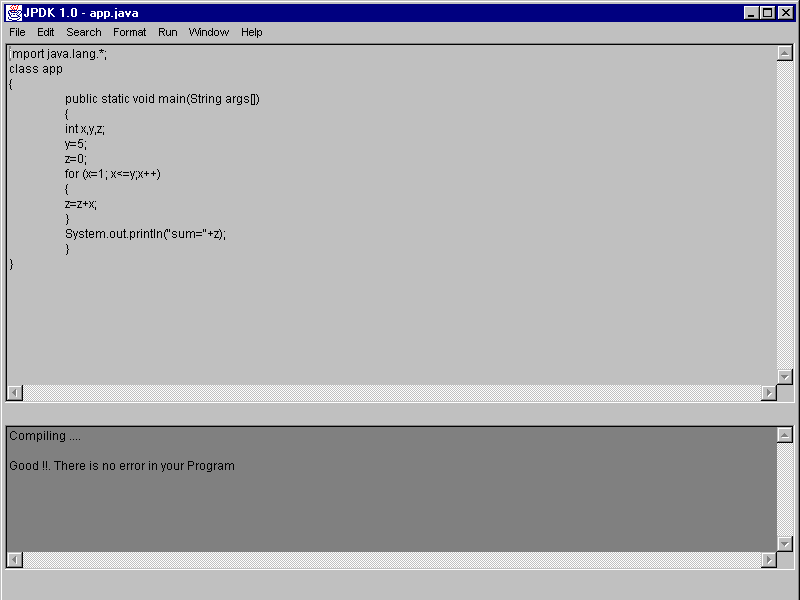
});

}

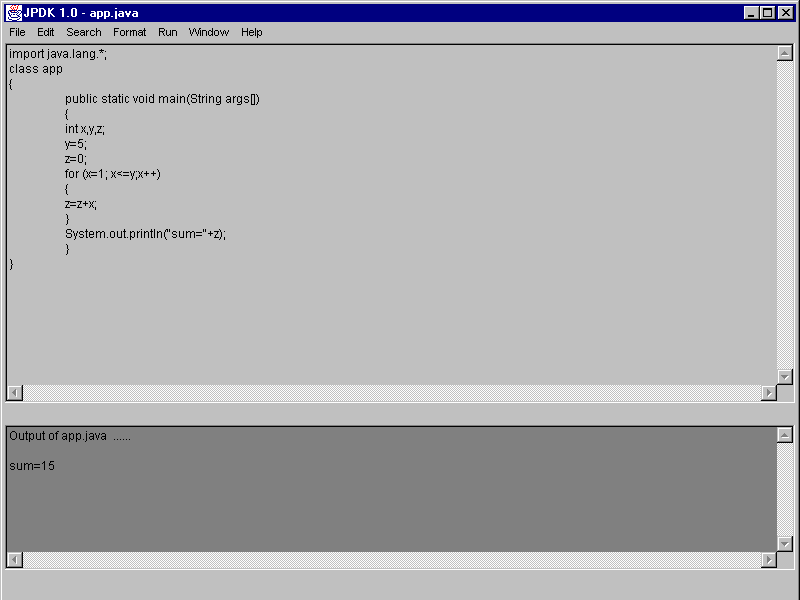
}

**6.2 OUTPUT**

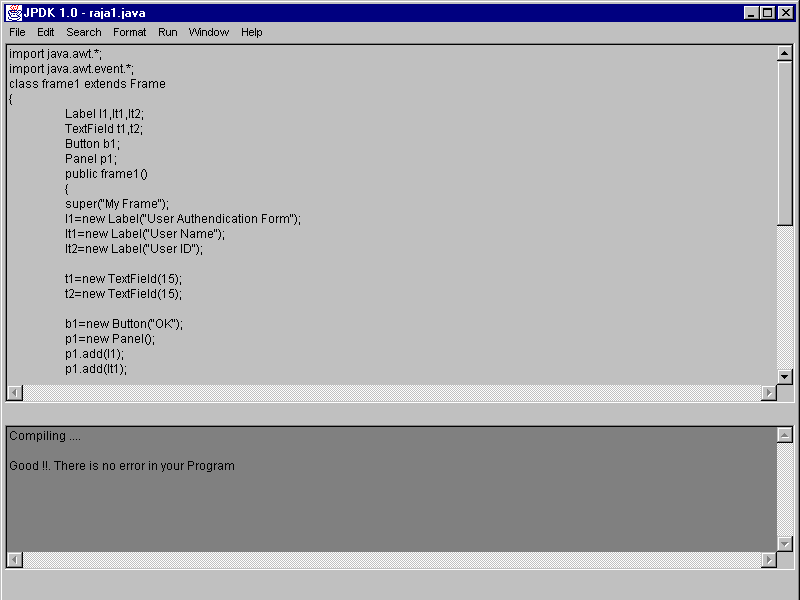
Application Program:



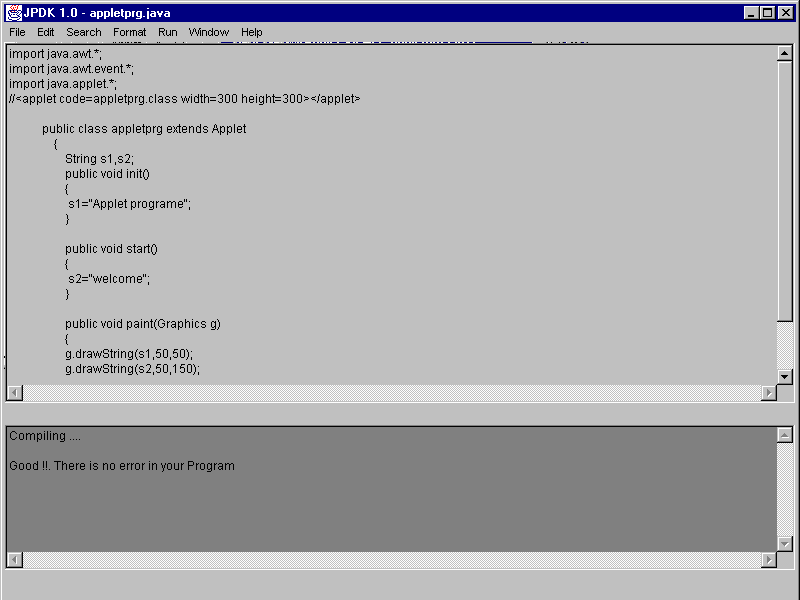
Output:

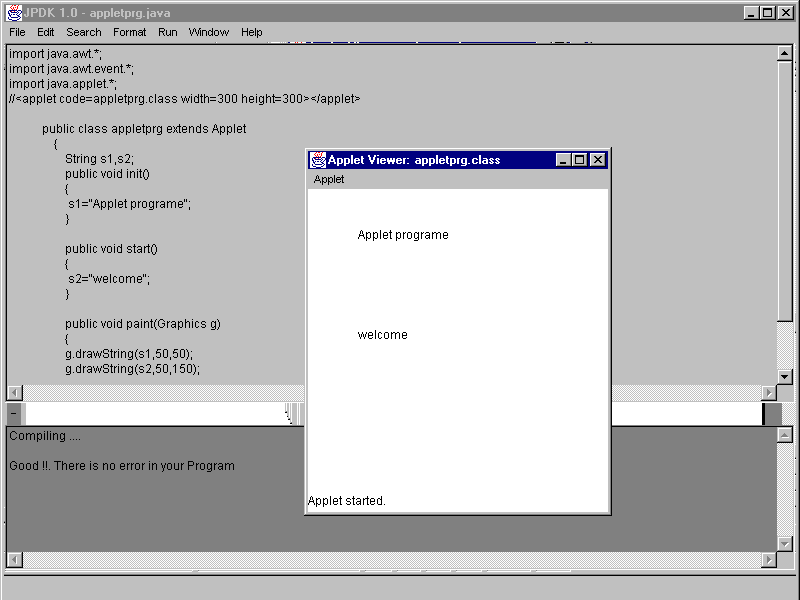


Frame Program:

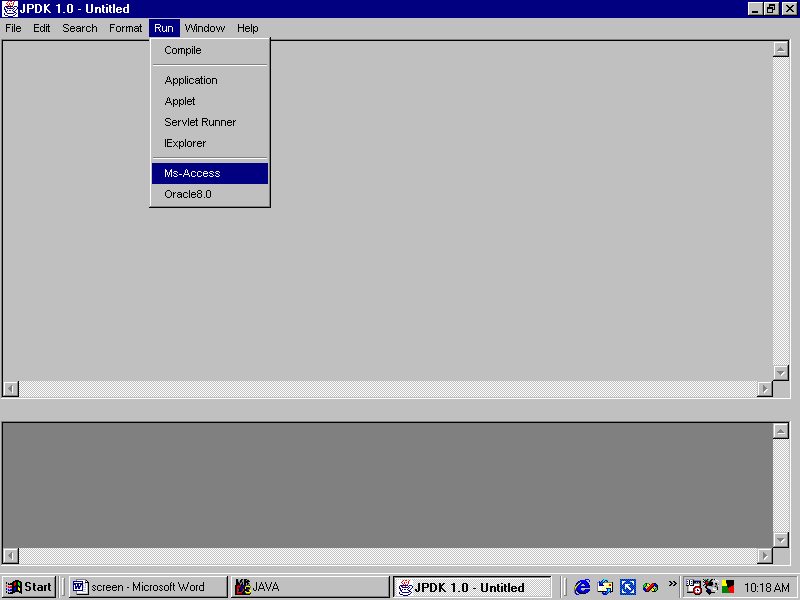


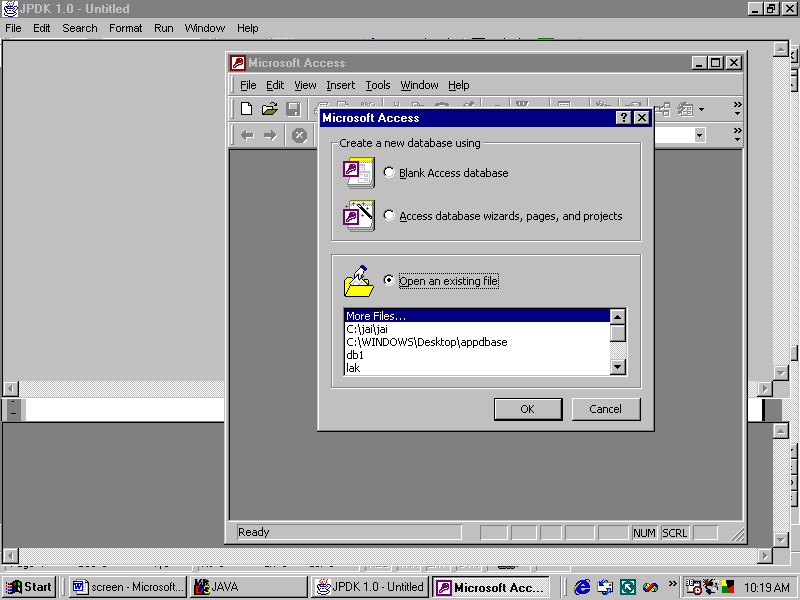
Applet Program :



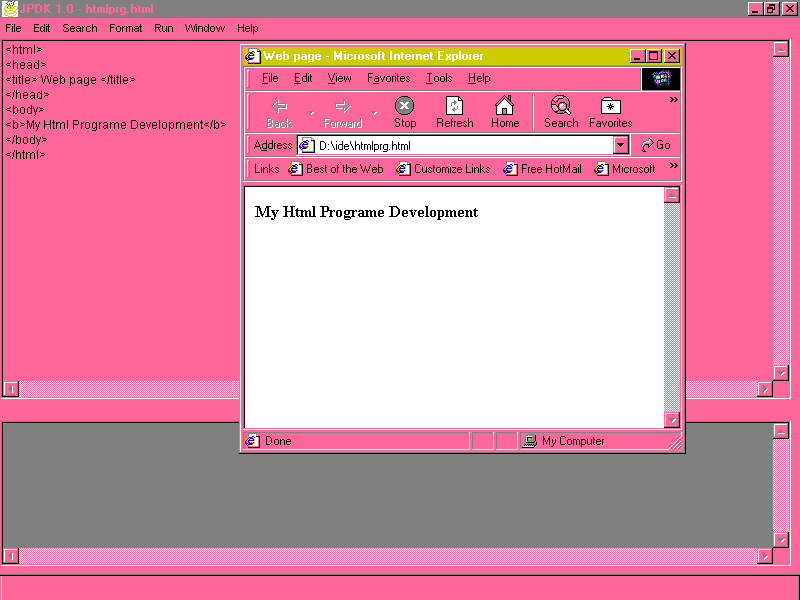


Access Interface

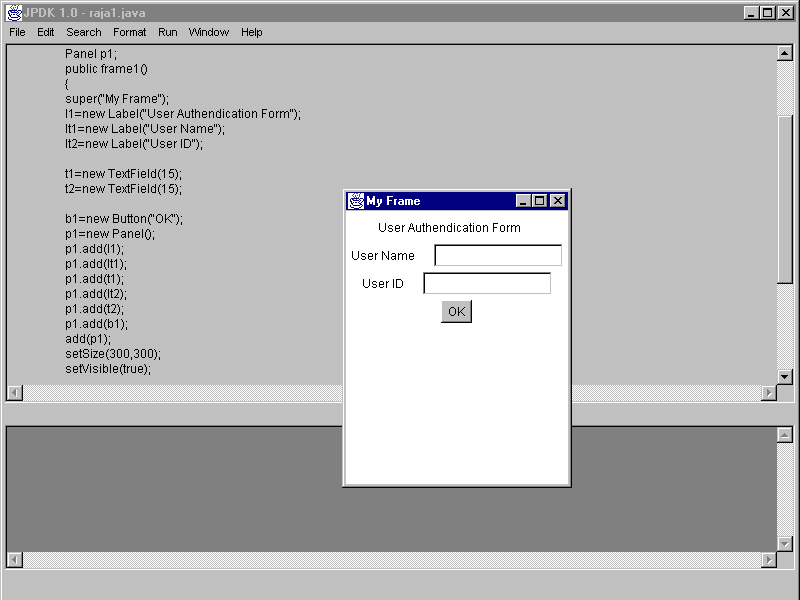




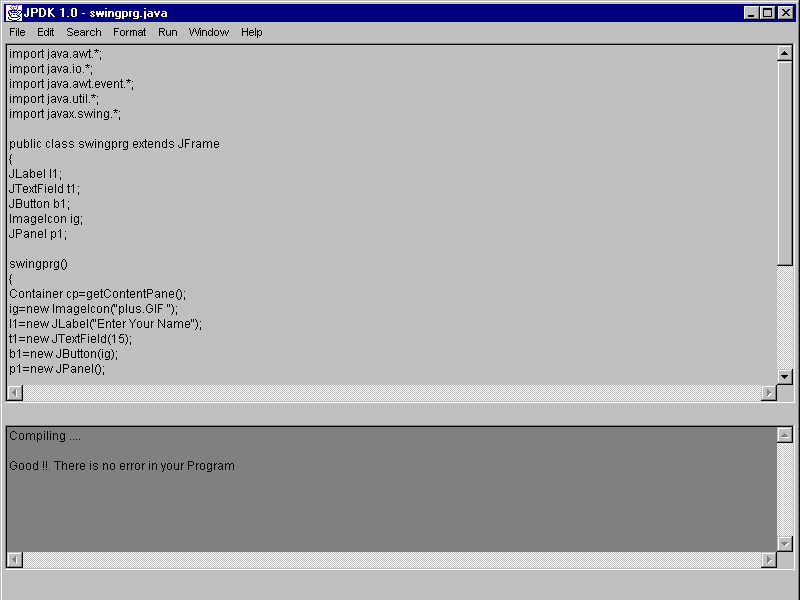
HTML Program Designer and Interface with EXPLORER

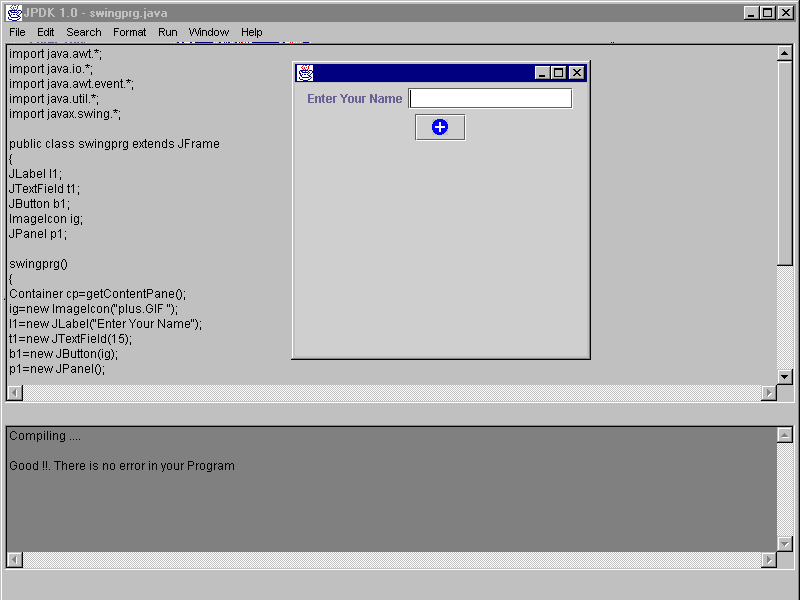


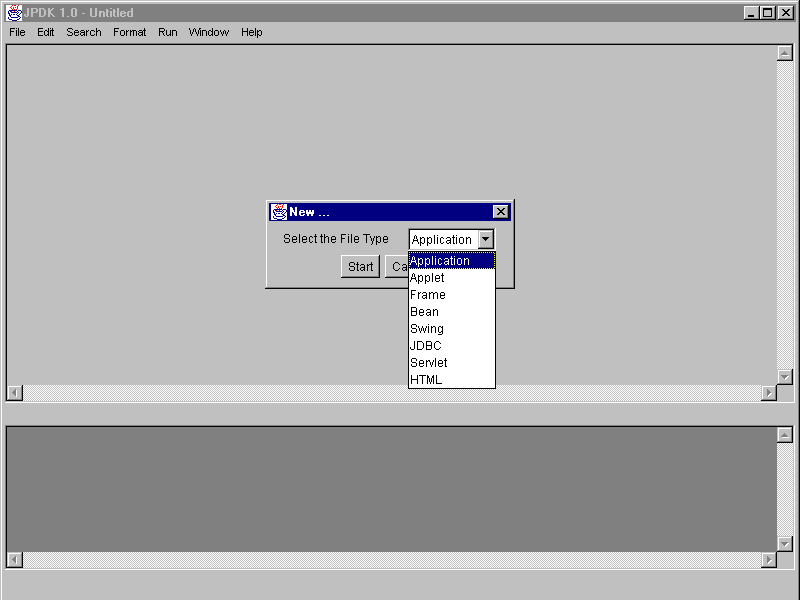
Output

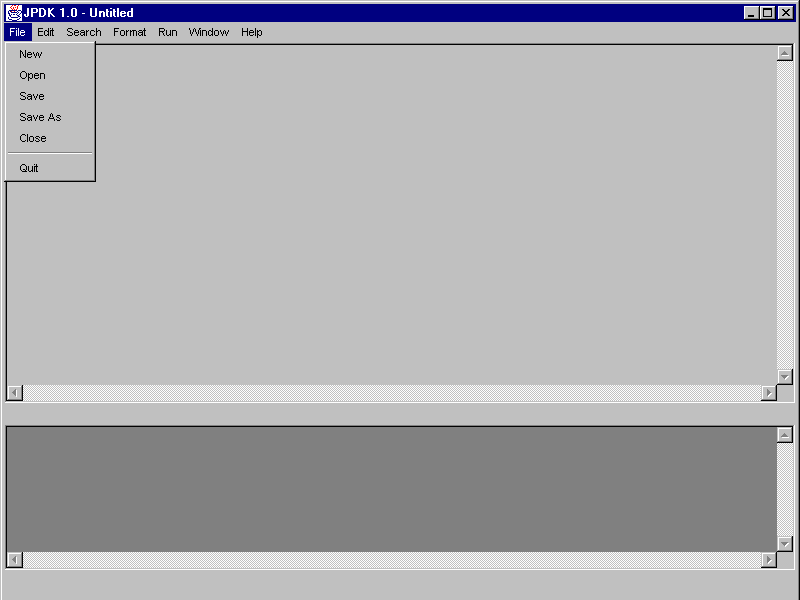


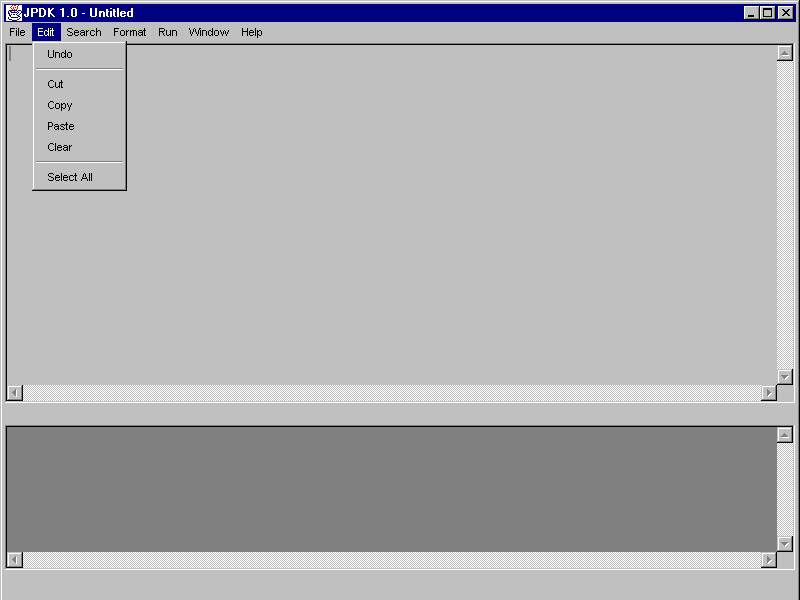
Swing Program

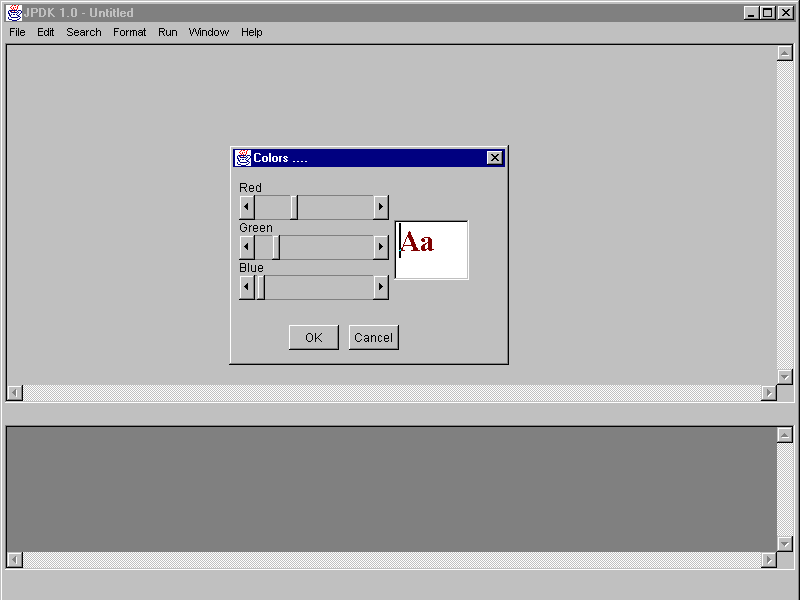












**7. CONCLUSION**

This My java platform editor developed is very evolutionary. This application is very flexible and more convenient for the users. This editor is very easy to use and edit the text and easily set different colors and fonts. And also it is easy to access the Oracle and MS-Access application. Using JPDK is easy to open the document in Explorer and run the generic servlets programs. This application supports more facilities like run and compile the programs within it.

As for future enhancement it can be made to support more facilities like running the Http Servlets programs within itself.

**8. BIBLIOGRAPHY**

1. **The Complete Reference-Java2**

**author - Patrick Naughton and Schildt**

**publisher - Tata Mc.Graw-Hill Edition**

1. **JAVA Servlet Programming**

**author - Jason Hunter with William Grawford**

**publisher - O’REILLY -Shroff Publishers & distributors PVT.LTD.**

1. **Java 2 Platform**

**author- Jamie Jaworski**

**pubisher – Techmedia Publication, First edition.**

1. **HTML 4.01 Weekend Crash Course**

**author - Greg Perry**

**publisher - IDG Books India(P) LTD.**

**5. Java Black Book**

WEBSITE:

1. [www.sun.com](http://www.sun.com)
2. [www.oreilly.com](http://www.oreilly.com)
3. [www.lucener.com](http://www.lucener.com)