SECTION 1: JAVA PROGRAMMING LAB

Structure **Page Nos.**

1.0 INTRODUCTION

In this section of lab course you will have hands on experience on Java Programming. This section is based on course MCS–024: Object Oriented Technology and Java Programming. Before getting into problem solving using java you will learn that how java programs are compiled and executed (interpreted). What are the basic requirements for java programming? You will learn writing of both categories of java programs: Java Applications and Java Applets, in this section. A list of programming problems is given session wise at the end of this section.

1.1 OBJECTIVES

After completing this lab section, you will be able to:

- Use Java's basic data types in your programs
- Writ Java programs using Sequential, Conditional and iterative statements
- Handle arrays of fixed and variable size
- Creating Classes and Objects using Java
- Implementing Constructors and Constructor Overloading
- Solving problems using Inheritance and Polymorphism
- Create your own package
- Create your own interface
- Handling exceptions arising in programs
- Use concept of multithreading in programs writing
- Handle Strings
- Use GUI components in your programs
- Use Layout Manages
- Write simple java programs to implement Sockets
- Write simple java programs to connect databases

1.2 GENERAL GUIDELINES

- You should come prepared for you lab session to properly utilize the maximum time of Lab session.

- You should attempt all lab exercises/assignments given in the list (session wise).

- You may seek assistance in doing the lab exercises from the available lab instructor.

- For each program, you should add general comments such as your name date of programming etc.

- There should be proper comments for description of the problem, requirement of class, function etc. Proper comments are to be provided as and when necessary in the programming. This will develop a good practice in you for rest of your professional life

- Your program should be interactive and properly documented with real Input/ Output data.

- Proper management of file of Lab record is necessary. Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, program code along with comments and output for various inputs given.

- There are total 10 lab sessions (of 3 hours each) for this lab course.

1.3 JAVA PROGRAMMING

Future of computing will revolve around Internet. Java is a Programming language developed by Sun Microsystems to take case of requirements of Internet computing. Java is a platform independent programming language that is why it is a very popular language for cross– platform applications and programming on Word Wide Web.

Java is an Object Oriented Programming Language, which serve the purpose of Object Oriented Paradigm of Programming. An object oriented language use the concept of abstraction, encapsulation, inheritance, and polymorphism to provide flexibility, modularity, and reusability for developing software.

Following feature of java make it popular and very useful programming language:

Platform Independent: Java programs can run on any machine and operating system that support Java Virtual Machine.

Multithreaded: these capabilities of java provide capability to single program to perform several tasks simultaneously. Multithreading is very useful for developing applications like animation, GUI, and networks. Unlike other programming languages multithreading is integrated in Java. In other programming languages, you have to call operating systems specific procedures to perform the task of multithreading.

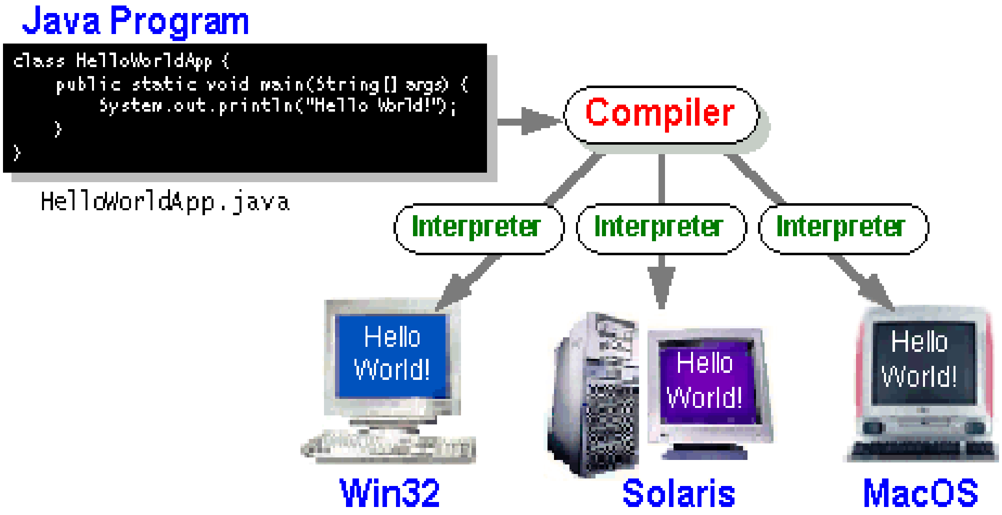
Distributed: Using java programs simultaneous processing can be done on multiple computer on Internet. Java provides strong networking features to write distributed programs.

Secure: Design of java is having multiple layers of security that ensures proper access of private data and have control over access of disk files.

You will cover major topics of Java Programming in this lab section during problem solving including programming structures, methods objects, inheritance, exception handling, multithreading, AWT, I/O, and applets. Because you know C programming language, it will be easier to you to learn java programming. It is very important to keep in mind the object-oriented features during problem solving.

Java Is also known as a Modern high-level language due to its characteristics: Simple, Architecture neutral, Object oriented, Portable, Distributed, High performance, Interpreted, Multithreaded, Robust, Dynamic, and Secure.

The Java programming language is unusual in the sense that a java program is both compiled and interpreted as shown in Figure 1.1.



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Figure 1.1: Execution of a Sample Java Program

Now let us see how you will run your java programs

1.4 PATH AND CLASSPATH SETTING

To run java programs you need JDK (Java Development Kit) installed on your machine. Latest version of JDK you can download from **java.sun.com** for free.

It is suggested, set PATH and CLASSPATH properly before you try to compile and run your java programs.

Because of the problem related to the ***path*** settings on your PC, your program may not compile and run.

PATH Setting

If whenever you type on your dos prompt

```
javac TestProg.java
```

it does not compile your TestProg program; but instead it spits out some strange message about a "**Bad Command**", .

What's wrong? Can you guess?It is not a matter of burry , this message does not mean that your machine is broken or that you didn't install JDK correctly. The machine says "**Bad command**" because it doesn't *know* what the commands **javac** or **java** mean!

You need to tell DOS that these commands live in the Java's JDK directory and that whenever it doesn't recognize a command like **javac** that it should also check the Java's JDK directory for a possible interpretation of this command.

This problem can be solved in two ways on by using DOS prompt , and another by using **My Computer** on your window screen.

Type path at your DOS prompt ,You will get

```
PATH=C:WINNT\...;C:\ProgramFiles\.....
```

where the dots represent all kinds of directory names. This is a list of all the places DOS looks when it's trying to find out what a particular command means. If your JDK directory is not on this list, your DOS won't understand the JDK's commands. So you need to add it to the list. You can do this by typing

```
set path=c:\jdkwhateverItIs\bin;%path%
```

where **whateverItIs** is the version of JDK . This adds your JDKdirectory to the rest of the existing path. If you have put your JDK directory somewhere else, alter the directory name in the command above appropriately to match the actual location. If there is any **spaces** in the command above (except between **set** and **path**) path will not be set properly.

Now try to run the java commands again.

If it *still* doesn't work, move your ***.java** files into the JDK directory -- and you should be able to work there (or in the **bin** subdirectory of the JDK directory).

Here is the problem again because you'll need to reset your path every time you turn your machine back on and want to work on the Java programs again. But there is a way to fix it permanently. You need to alter (edit) your **autoexec.bat** file, which is in your main **C: directory**. It is suggested to copy this file first to a back up version, then do required modifications told to you here.

```
C: copy autoexec.bat autoexecBACK1.bat
```

Because this is the file that has all the important settings for making your PC run the way you want it. One of the settings it makes is the **PATH**.

Now edit the **autoexec.bat** file and find the line that sets the path. Add your version of the "**set path=C:/jdk.2....**" command (the one you used earlier) *after* the line that sets the rest of the path in the **autoexec.bat** file. Now save the file. Then the next time you open a DOS window (it maybe be required to restart your machine), the **PATH** should be set correctly. If you made an error in editing **autoexec.bat**, things might not work quite right now . In this case, just copy your back up version of **autoexecBACK1.bat** back into **autoexec.bat** and try the whole process again.

CLASSPATH Setting

If you find some message during compilation or execution of Java program that a class is not found .You know where the class file is ? May be in your current working directory. To tell the path of class files you need to set your **CLASSPATH**.

At the DOS prompt, type

classpath

and DOS will respond with

CLASSPATH=C:stuff.....

Now type

set classpath=.;%classpath%

Don't leave any spaces except after the word **set**. This should add the current directory (the directory having your class files) to your classpath. Then **java** should know to look in the current directory for the ***.class** files that your **javac** compiles are creating.

In XP/2000/ME machines PATH and CLASSPATH can be set using "My Computer." The process is given as follows:

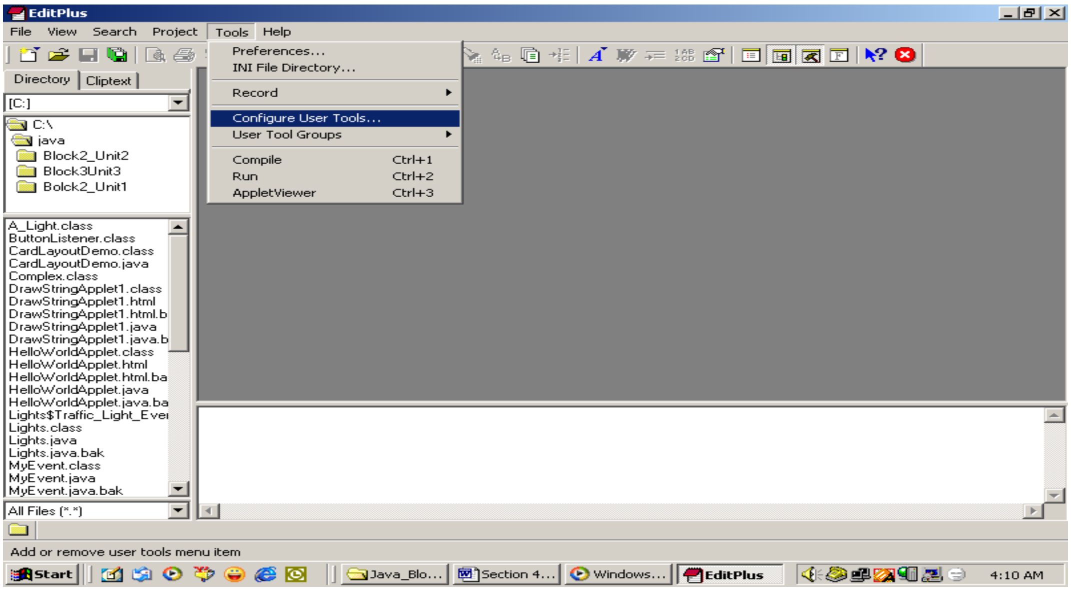
1. Right click on "My Computer" and Click on Properties.
2. Click on the Advanced tab.
3. Click on the "Environment Variables" button near the bottom.
4. A dialog box comes up with two boxes: In the bottom box, look for "Path" and highlight that line. Then click on "Edit" .A small dialog box appears. In the second text field for the value, at the END of the line, add a semicolon then the path to where your **java.exe** file is. For example path is like:C:\jdk1.3….\bin
5. Click "OK."
6. Now, in the top box, click on the "New" button.
7. In the first field, enter "classpath" as one word. For the value in the second field, enter a single period.

This is all you have to do for setting PATH and CLASSPATH.
Now you can compile and run your applications and applets without any problem of PATH and CLASSPATH.PATH and CLASSPATH setting need to be done only in the case of, if you are willing to run programs on DOS prompt. If you write Java programs in some specific editors like EditPlus, JBbuilderfor etc. then these seting discussed above are editor dependent.

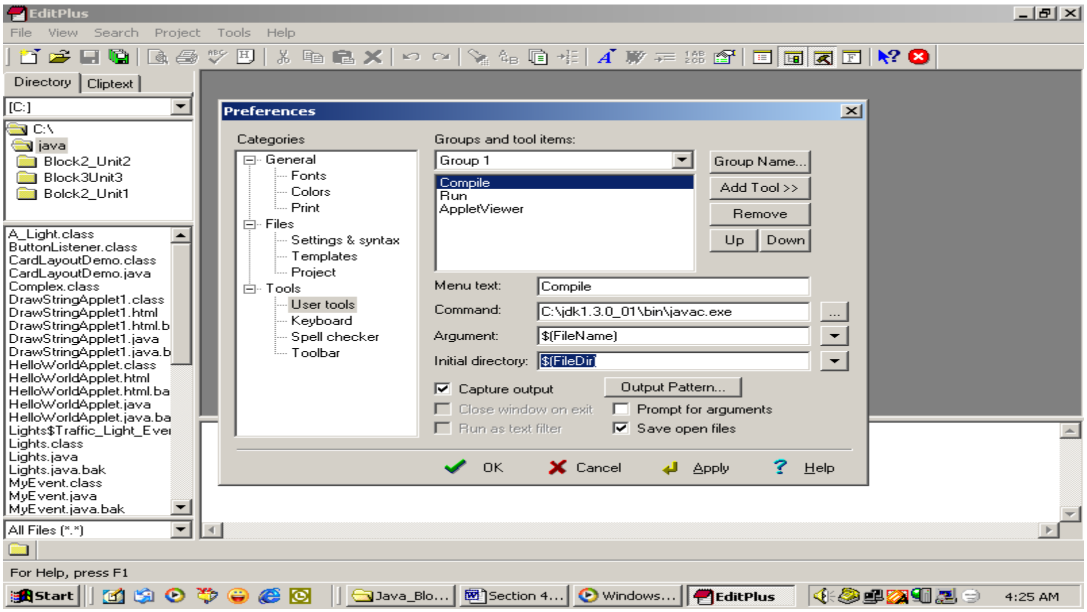
Also you may use NetBeans IDE for writing your programs . It lets you quickly and easily develop Java desktop, mobile, and web applications. NetBeans IDE is free and open source with a large community of users and developers around the world. You may install NetBeans IDE on any operating systems that support Java. For more details you may visit: **www.netbeans.org**

Now we will learn how EditPlus can be set for Java Programming. EditPlus is distributed as Shareware. You can freely download and try it for 30 days from **http://www.editplus.com/download.html**.

1. Open EditPlus
2. Select Tools–>Configure User Tools .You will find a dialog Window in which you have to select a Group Name
3. Select Group1 as Group Name you can select any..
4. Click on Add Tool >> button and select Program
   Setting for Compilation of Java Programs :
5. In front of Menu text : write Compile



6. In front of Command: browse and select C:\jdk1.3.0_01\bin\javac.exe or the Directory where javac.exe file is located.
7. In front of Argument: select $(FileName)
8. In front of Initial directory: select $(FileDir)
9. Select Capture output box.
10. Select Save open files box.



Groups and tool items: Group 1

Compile
Run
AppletViewer

Menu text: Compile

Command: C:\jdk1.3.0_01\bin\javac.exe

Argument: $(FileName)

Initial directory: $(FileDir)

☑ Capture output

□ Close window on exit

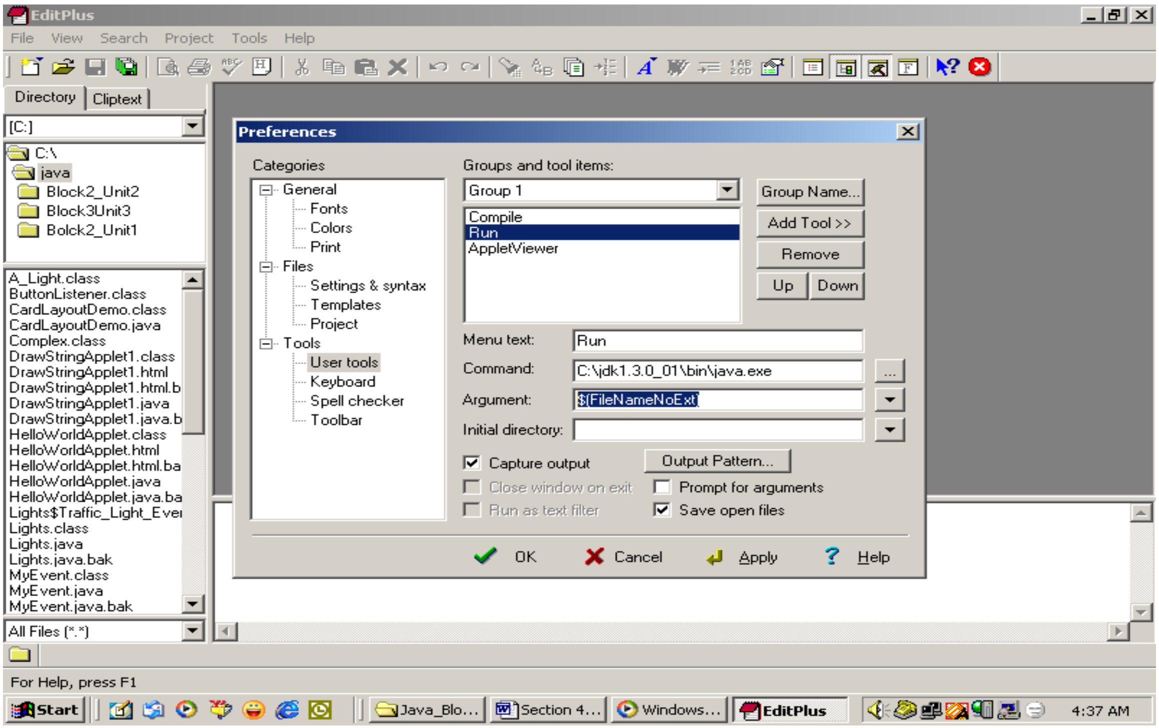
□ Prompt for arguments

□ Run as text filter

☑ Save open files

Setting for Running Java Application Programs:

11. In front of Menu text : write Run
12. In front of Command: browse and select C:\jdk1.3.0_01\bin\java.exe or the Directory where java.exe file is located.
13. In front of Argument: select $(FileNameNoExt)
14. In front of Initial directory: Leave black
15. Select Capture output box.
16. Select Save open files box.



Groups and tool items: Group 1

Compile
Run
AppletViewer

Menu text: Run

Command: C:\jdk1.3.0_01\bin\java.exe

Argument: $(FileNameNoExt)

Initial directory:

☑ Capture output

□ Close window on exit

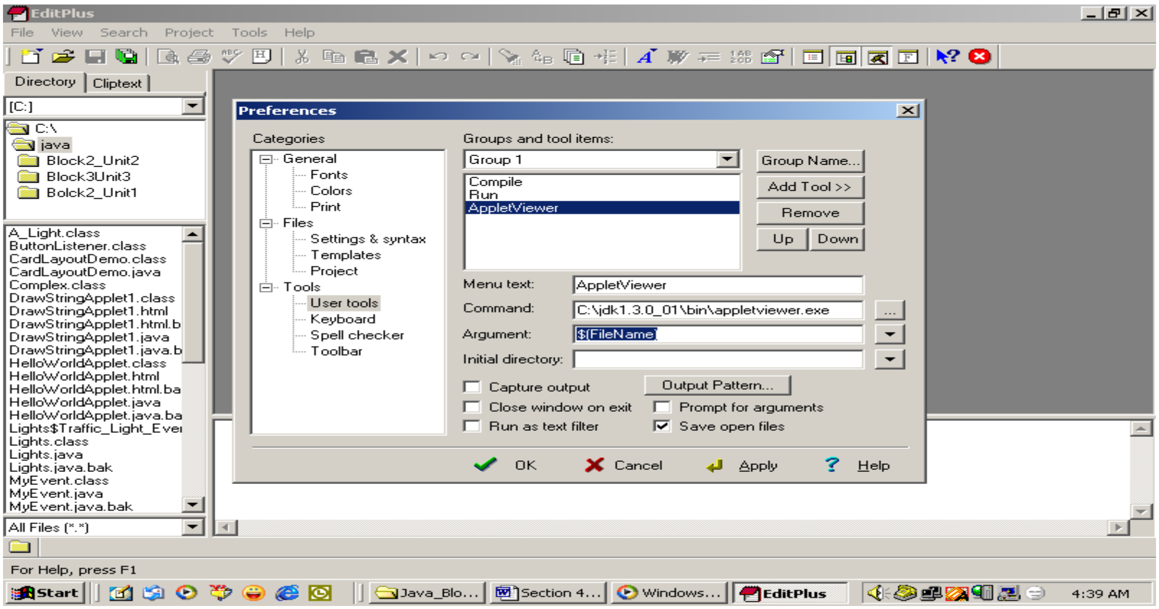
□ Prompt for arguments

□ Run as text filter

☑ Save open files

Setting for running java Applet Programs:

17. In front of Menu text : write AppletViewer
18. In front of Command: browse and select C:\jdk1.3.0_01\bin\appletviewer.exe or the Directory where appletviewer.exe file is located.
19. In front of Argument: select $(FileName)
20. In front of Initial directory: Leave black
22. Select Save open files box.



Groups and tool items: Group 1

Compile
Run
AppletViewer

Menu text: AppletViewer

Command: C:\jdk1.3.0_01\bin\appletviewer.exe

Argument: $(FileName)

Initial directory:

□ Capture output

□ Close window on exit

□ Prompt for arguments

□ Run as text filter

☑ Save open files

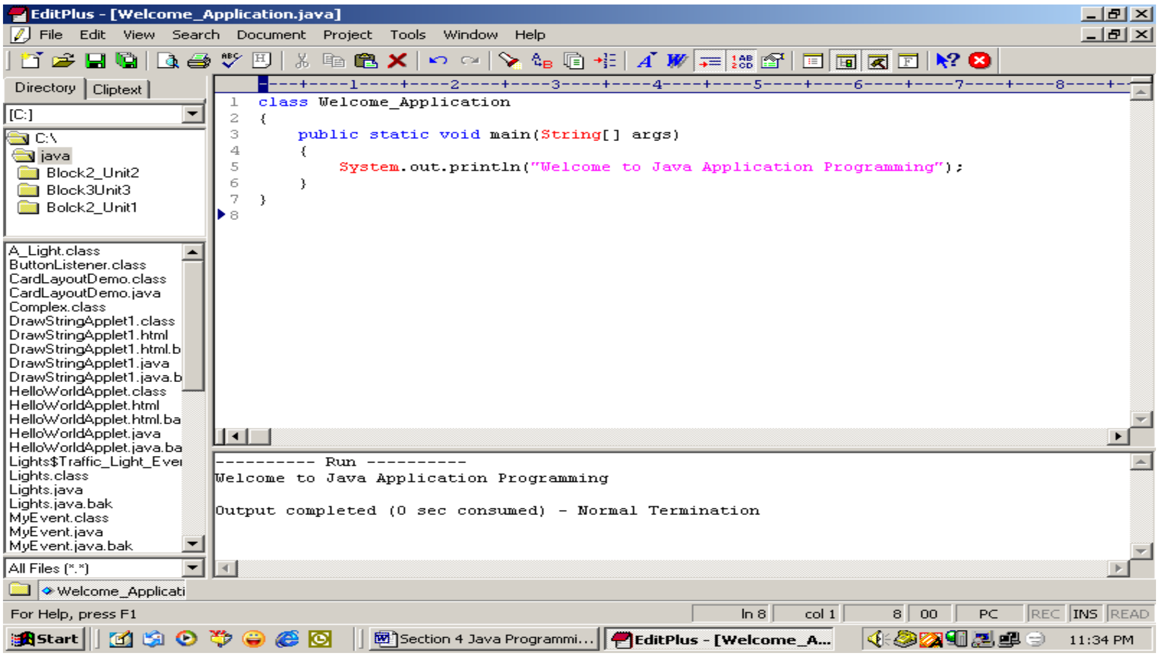
Now you will find that three more items Compile, Run, and AppletViewer are added in Tools menu of EditPlus.For Compile ctrl+1, for Run ctrl+2, and for AppletViewer ctrl+3 can be used.

EditPlus is set to be used for Java Programming. Let us take one application and one applet Program running using EditPlus.

1.5 EXAMPLE PROGRAMS

Welcome Application Program:

1. Write your program in EditPlus
2. Compile using ctrl+1 button
3. Run using ctrl+2 button


EditPlus - [Welcome_Application.java]
File Edit View Search Document Project Tools Window Help
Directory Cliptext
For Help, press F1

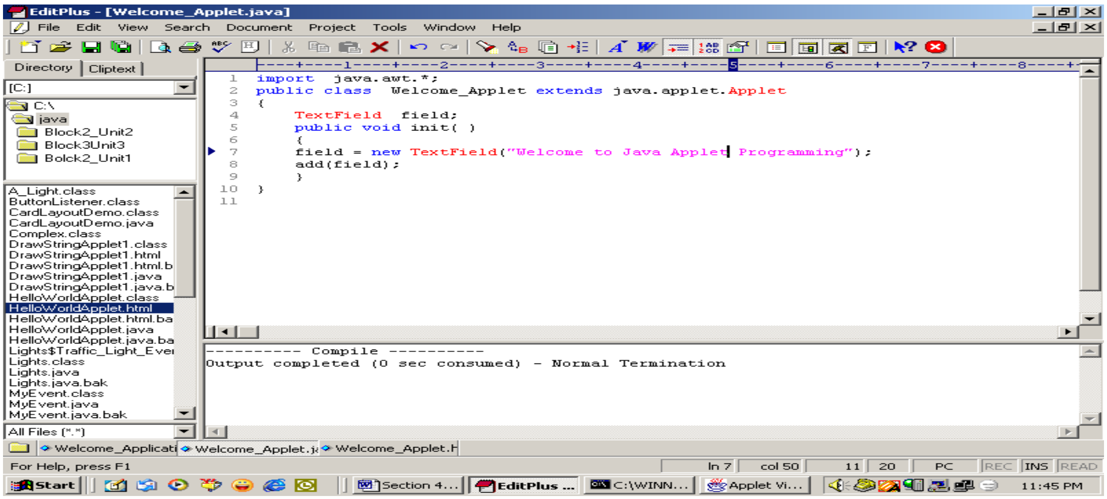
```
class Welcome_Application
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java Application Programming");
    }
}
```

```
---------- Run ----------
Welcome to Java Application Programming

Output completed (0 sec consumed) - Normal Termination
```

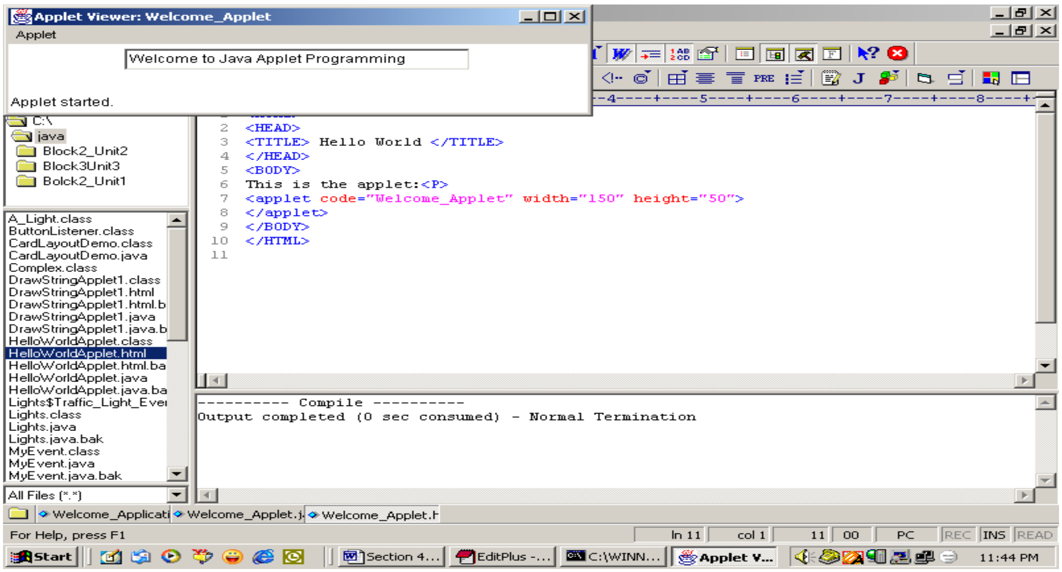
Welcome Application Program:

1. Write your program in EditPlus
2. Compile using ctrl+1 button
3. Run using ctrl+3 button but take care you are applying ctr+3 on appropriate HTML file .


EditPlus - [Welcome_Applet.java]
File Edit View Search Document Project Tools Window Help
Directory Cliptext
For Help, press F1

```
import  java.awt.*;
public class  Welcome_Applet extends java.applet.Applet
{
    TextField  field;
    public void init( )
    {
    field = new TextField("Welcome to Java Applet Programming");
    add(field);
    }
}
```

```
---------- Compile ----------
Output completed (0 sec consumed) - Normal Termination
```



Welcome to Java Applet Programming

Applet started.

```
<HEAD>
<TITLE> Hello World </TITLE>
</HEAD>
<BODY>
This is the applet:<P>
<applet code="Welcome_Applet" width="150" height="50">
</applet>
</BODY>
</HTML>
```

```
---------- Compile ----------
Output completed (0 sec consumed) - Normal Termination
```

List of assignments provided in section 1.6 for all the ten lab sessions assigned for Java Programming. By doing these assignments, you will be learning the concepts and programming skills you are gaining in MCS-024 course.

1.6 LIST OF LAB ASSIGNMENTS

SESSION 1

Exercise 1: Write a program in Java to implement the formula

Area = Height × Width

Exercise 2: Write a program in Java to find the result of following expression (Assume a = 10, b = 5)

i) (a < < 2) + (b > > 2)
ii) (a) | | (b > 0)
iii) (a + b ∗100) / 10
iv) a & b

Exercise 3: Write a program in Java to explain the use of break and continue statements.

SESSION 2

Exercise 4: Write a program in Java to find the average of marks obtained by a study in five papers.

	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
Marks	50	70	65	80	56

Exercise 5: Write a program in Java for find A×B where A is a matrix of 3×3 and B is a matrix of 3×4.

Exercise 6: Write a program in Java with class Rectangle with the data fields width, length, area and color .The length, width and area are of double type and color is of string type .The methods are set_ length () , set_width (), set_ color(), and find_ area (). Create two object of Rectangle and compare their area and color. If area and color both are same for the objects then display "Matching Rectangles" otherwise display "Non matching Rectangle".

SESSION 3

Exercise 7: Create a class Account with two overloaded constructors. First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, addresses, type of account and current balance. Account class is having methods Deposit (), Withdraw (), and Get_Balance().Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them.

Exercise 8: Write a program in Java to create a stack class with push() and pop () methods. Create two objects of stack with 10 data item in both. Compare the top elements of both stack and print the comparison result.

Exercise 9: Write Java program to show that private member of a super class cannot be accessed from derived classes.

SESSION 4

Exercise 10: Write a program in Java to create a Player class. Intermit classes Cricket _Player, Football _Player and Hockey_ Player from Player class.

Exercise 11: Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker is has a name and a salary rate. Write method ComPay(int hours) to computethe week pay of every worker.A DailyWorker is paid on the basis of number of days s/he work.The SalariedWorker g ets paid the wage for 40 hours a week no matter what actual hours is. Test this program to calculate the pay of workers. You are expected to use concept of polymorphism to write this program.

Exercise 12: Consider trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning call. The charges depend on the duration and the type of the call. Writ a program-using concept of polymorphism in Java to calculate the chases.

SESSION 5

Exercise13: Write a program to make a package Balance in which has Account class with Display_Balance method in it. Import Balance package in another program to access Display_Balance method of Account class.

Exercise 14: Write a program in Java to show the usefulness of Interfaces as a place to keep constants of the program.

Exercise 15: Write a program in java which implement interface Student which has two methods Display_Grade and Atrendance for PG_Students and UG_Students(PG_Students and UG_Students are two different classes for Post Graduate and Under Graduate students respectively).

SESSION 6

Exercise 16: Write a program in Java to display name and roll number of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsExeption, so that any such problem doesn't cause illegal termination of program.

Exercise 17: Write a java program to facilitate user to handle any chance of divide by zero exception.

Exercise 18: On singles track two vehicles are running for as vehicles are going in same direction there is no problem. If the vehicles are running in same direction there is a chance of collision. To avoid collisions write a java program using exception handling. You are free to make necessary assumptions.

Exercise 19: Write a java program to create five threads with different priorities. Send two threads of highest priority in sleep state. Check the aliveness of the threads and mark which dread is long listing.

SESSION 7

Exercise 20: Write a java program using thread synchronization in multithreading (You can take some objects visible on screen for real time effect).

Exercise 21: Writ a program in java to create a String object. Initialize this object with your name. Find the length of your name using appropriate String method. Find whether character 'a' is in your name or not, if yes find the number of time 'a' it appear in your name. Print locations of occurrences of 'a' .Try same for different String objects.

Exercise 22: Write program in Java for String handling which perform followings

i) Checks the capacity of StringBuffer objects
ii) Reverse the contents of a string given on console and convert the resultant string in upper case.
iii) Read a string from console and append it to the resultant string of ii.

SESSION 8

Exercise 23: Write a program in java to read a statement from console, convert it in to uppercase and again print on console.

Exercise 24: Write a program in java, which takes the name of a file from user, read the contents of the file and display it on the console.

Exercise 25: Write a java program to copy a file into another file.

SESSION 9

Exercise 26: Write a Java Applet program which read your name and address in different text fields and when a button named find is pressed the sum of the length of characters in name and address is displayed in another text field. Use appropriate colors, layout to make good look of your applet.

Exercise 27: Write a Java Applet program, which provide a text area with horizontal and vertical scrollbars. Type some lines of text in text area and use scrollbars for movements in text area. Read a word in a text field and find whether the word is in the content of text area or not.

Exercise 28: Write a Java program to find the numeric address of the following web sites

i) www.ignou.ac.in
ii) www.indiatimes .com
iii) www.rediff.com
iv) www.apple.com

in addition to this find the Internet Address of your local host.

SESSION 10

Exercise 29: Write a program to test Socket functionality for appropriate hostname and port number.

Exercise 30: Write a Java program to connect to a database created in MS–ACCESS/SQL–SERVER/ORACLE using JDBC concept. Perform basic operations of Selection , Insertion and Deletion on the database.