

SECTION 1 DATA AND FILE STRUCTURES LAB MANUAL

Structure	Page No.
1.0 Introduction	5
1.1 Objectives	6
1.2 Arrays	6
1.3 Structures	7
1.4 Linked Lists	7
1.5 Stacks	8
1.6 Queues	8
1.7 Trees	9
1.8 Advanced Trees	10
1.9 Graphs	10
1.10 Searching and Sorting	10
1.11 Advanced Data Structures	11
1.12 Summary	11
1.13 Further Readings	11

1.0 INTRODUCTION

A data structure is one of the most important courses that is to be mastered by any software professional. They are the building blocks of a program. The life of a house depends on the strength of its pillars. Similarly, the life of a program depends on the strength of the data structures used in the program. It is very important to write an algorithm before writing a program. This will enable you to focus more on the semantics of the program rather than syntax. Also, it will lead you to write efficient programs.

In this section, a brief introduction to the data structures that are discussed in the corresponding course (MCS-021) is given. Each introduction is followed by a list of programming problems. Wherever, a particular program for a given programming problem was already given in the course material, it is suggested that you should not copy it. Rather, you should develop your own logic to write the program.

To successfully complete this section, the learner must have already:

- Thoroughly studied the course material of MCS-011,
- Thoroughly studied the Section-1(C Programming Manual) of MCSL-017, and
- Executed most of the programming problems of the Lab assignments.

Also, to successfully complete this section, the learner should adhere to the following:

- Before attending the lab session, the learner must have already written algorithms and programs in his/her lab record. This activity should be treated as home work that is to be done before attending the lab session.
- The learner must have already thoroughly studied the corresponding units of the course material (MCS-021) before attempting to write algorithms and programs for the programming problems given in a particular lab session.
- Ensure that you include comments in your program. This is a practice, which will enable others to understand your program and enable you to understand the program written by you after a long time.
- Ensure that the lab record includes **algorithms, programs, I/O and complexity (both time and space)** of each program.

1.1 OBJECTIVES

After following the instructions in this section, you should be able to

- Write programs using basic data structures such as Arrays etc. as well as advanced data structures such as trees etc.
-

1.2 ARRAYS

An Array is a collection of element(s) of the same data type. They are referred through an index along with a name.

Consider the following examples:

```
int a[5];
```

The above is a declaration of an array which is a collection of 5 elements of integer data type. Since the collection consists of 5 elements, the index starts from 0(zero) and ends at 4(four). Of course, it is very important to remember that, in ‘C’ language, the index always starts at 0 and ends at (length of array-1). The *length of array* is nothing but the number of maximum elements that can reside in an array. In our example, the length is 5. The first element is referred as a[0], the second element is referred to as a[1] . The third, fourth and fifth elements are referred to as a[2], a[3] and a[4].

The example given above is a single dimensional array. We can declare arrays of any dimension.

Consider the following array:

```
int b[5][6];
```

This is a two-dimensional array. It can hold a maximum of 30(5X6) elements. The first element is referred to as b[0][0], the second as b[0][1] and so on. The last element is referred to as b[4][5].

It is always possible to declare arrays of elements of any type. It is possible to declare an array of characters, structures, etc.

For more information on arrays, Refer to Unit-6 of MCS-011, Block-2.

Session 1: Arrays (3 hours)

1. Write a program in ‘C’ language that accepts two matrices as input and prints their product.

Input: Two matrices A and B in two-dimensional arrays

Output: Matrix ‘C’ which is result of A X B in the form of two dimensional array

2. Write a program in ‘C’ Language to accept 10 strings as input and print them in lexicographic order

Input: 10 strings (use array of strings)

Output: 10 input strings in lexicographic order

3. Write a program in ‘C’ Language that accepts two strings S1 and S2 as input. The program should check if S2 is a substring of S1 or not. If S2 is a substring of S1, then the program should output the starting location and ending location

of S2 in S1. If S2 appears more than once in S1, then the locations of all instances have to be given.

4. Write a program to concatenate two strings S1 and S2.
-

1.3 STRUCTURES

A Structure is a collection of element(s) of one or more data types. The value of each element of the structure is referred through its corresponding identifier coupled with the variable name of the structure.

Consider the following example:

```
struct student {  
    char name[25];  
    char *address;  
    int telephone[8];  
};
```

In the above example, the name of the structure is student. There are three elements in the structure (of course, you can declare any number of elements in the structure). They are name, address and telephone number. The first element is an array of characters, the second is address which is a pointer to a string or sequence of characters and the third is an array of integers. It is also permissible to have nested structures. That is, a structure inside a structure.

Consider the following declaration:

```
struct student s;
```

s is a variable of type **student**.

The values of the elements in the above given example can be referred as s.name, s.address and s.telephone.

For more information on Structures, refer to unit-9 of MCS-011, Block, 3

Session 2: Structures (3 hours)

1. Write a program in ‘C’ language, which accepts Enrolment number, Name Aggregate marks secured in a Program by a student. Assign ranks to students according to the marks secured. Rank-1 should be awarded to the students who secured the highest marks and so on. The program should print the enrolment number, name of the student and the rank secured in ascending order.
 2. Write a program in ‘C’ language to multiply two sparse matrices.
 3. Write a program in ‘C’ language to accept a paragraph of text as input. Make a list of words and the number of occurrences of each word in the paragraph as output. As part of the processing, an array and structure should be created wherein each structure consists of two fields, namely, one for storing the word and the other for storing the number of occurrences of that word.
-

1.4 LINKED LISTS

A linked list is a sequence of zero or more number of elements that are connected directly or indirectly through pointers.

A linked list can be a singly linked list or a doubly linked list. Again, a singly linked

list or a doubly linked list can also be a circularly linked list.
For more information on linked lists, refer to unit-3 of MCS-021.
For more information on pointers, refer to unit-10 of MCS-011, Block-3.

Session 3: Linked Lists (3 hours)

1. Write a program in ‘C’ language for the creation of a list. Also, write a procedure for deletion of an element from the list. Use pointers.

You have to write the above program separately for Singly linked list, Doubly linked list and Circularly linked lists (both singly linked and doubly linked). Make necessary assumptions.
 2. Write a program in ‘C’ language that accepts two singly linked lists A and B as input. Now, print a singly linked list that consists of only those elements, which are common to both A and B.
 3. Write a program in ‘C’ language to accept a singly linked list of integers as input. Now, sort the elements of the list in ascending order. Then, accept an integer as input. Insert this integer into the singly linked list at the appropriate position.
-

1.5 STACKS

A Stack is a LIFO (Last In First Out) data structure. A stack can be implemented using arrays or pointers. But, the disadvantages of using arrays is that the maximum number of elements that can be stored are limited. This disadvantage can be overcome by using pointers.

There are two important operations associated with a stack. They are **push** and **pop**. A **push** will add an element to the stack and a **pop** will delete (either really by freeing the memory location or by reducing the counter indicating the number of elements in the stack by one).

For more information on stacks, refer to unit-4 of MCS-021.

Session 4: Stacks (3 hours)

1. Write a program in ‘C’ language to convert a prefix expression to a postfix expression using pointers.
 2. Write a program in ‘C’ language to reverse an input string.
 3. Write a program in ‘C’ language to implement multiple stacks in a single array.
-

1.6 QUEUES

A Queue is a FIFO (First In First Out) data structure. A Queue can be implemented using arrays or pointers. The same disadvantages that are associated with Stacks hold true for Queues. In addition, in the case of Queues, whenever the element at front is deleted, all the elements are to be moved one position forward which leaded to time overhead. Else, there will be a waste of memory. With the help of pointers, these disadvantages can be overcome.

There are two important operations associated with a Queue. They are **Add** and **Delete**. An Add operation will add an element to the end (or rear) of the queue. A Delete operation will delete an element from the front of the queue. One simple way of understanding these operations is to remember that the order of deletion of elements from the queue will be exactly the same as the order of addition of elements to the Queue. You can draw an analogy of Queue data structure to the people standing in a queue for entering a bus etc. Whoever is at the front will get the opportunity to enter the bus first. Any person who is not in the queue and intends to enter the bus should stand in the queue at the rear/last.

A Dequeue is a special form of queue in which addition/deletion of elements is possible to be done at the front as well as at the rear. So, the concept of FIFO does not hold here.

For more information on Queues, refer to unit-5 of MCS-021.

Session 5: Queues (3 hours)

1. Write a program in ‘C’ language to implement a Dequeue using Arrays. All operations associated with a Dequeue are to be implemented.
2. Write a program in ‘C’ language to implement a Dequeue using pointers. All operations associated with a Dequeue are to be implemented.
3. Write a program in ‘C’ language to reverse the elements of a queue.
4. Write a program in ‘C’ language to implement a queue using two stacks.
5. Write a program in ‘C’ language to implement a stack using two queues.

1.7 TREES

Often, there is confusion about the differences between a tree and a binary tree. The major difference is that a Tree is always non-empty. It means that there is at least one element in it. It obviously leads to the conclusion that a Tree always had a root. The existence of the remaining elements is optional. It is possible for a Binary tree to be empty. For a binary tree, the number of children of a node cannot be more than 2. There is no restriction on the number of children to the nodes of a tree.

For more information on Trees and Binary trees, refer to unit-6 of MCS-21.

Session 6: Trees and Binary Trees (3 hours)

1. Write a program in ‘C’ language for the creation of a binary tree. Also, provide for insertion and deletion operations.
2. Write a program in ‘C’ language for pre-order, post-order and in-order tranversals of a Binary tree. Don’t use Recursion.
3. Write a program in ‘C’ language to accept a tree as input and convert it into a binary tree. Print the resultant binary tree.
4. Write a program in ‘C’ language to accept a binary tree as input and check if the input binary tree is a full binary tree or not.
5. Write a program in ‘C’ language to accept two trees as input and check if both of them are the same. Give the appropriate message.

6. Write a program in ‘C’ language to count the number of leaves of a Binary tree.
-

1.8 ADVANCED TREES

The trees that were discussed in this section are Binary search trees and AVL trees.

A binary search tree (BST) is a binary tree in which all the elements of the left subtree of a node are less than the element stored in the node and the elements stored in the right subtree of the node. So, whenever there is a need to insert an element into a BST, the element is compared with the root. If the element is less than the root, then continue the comparison with the left node of the root recursively. If the element is larger than the root, then the same process has to be carried to the nodes at the right of the root recursively. BST is one of the major applications of the Binary Tree.

An AVL tree is a balanced Binary search tree.

For more information on Binary search trees and AVL trees , refer to unit-7 of MCS-21.

Session 7: Advanced trees (3 hours)

1. Write a program in ‘C’ language to create a binary search tree. Also, accept a key value and search for it in BST. Print the appropriate message as output.
 2. Write a program in ‘C’ language to insert 15, 25, 2, 4, 3, 1, 50 into an initially empty AVL tree. Make assumptions, if necessary.
-

1.9 GRAPHS

A Graph is a collection of Vertices and Edges. The set of Vertices is always non-empty. Edges represent ordered or unordered pairs of vertices which are directly connected to each other. If the pairs of vertices are unordered, it means that an edge (v_1, v_2) indicates that v_1 is directly connected to v_2 and vice-versa. If the pairs of vertices are ordered, it means that an edge (v_1, v_2) indicates that v_1 is directly connected to v_2 . V_2 is directly connected to v_1 only if the edge (v_2, v_1) is present in the list of edges.

For more information on Graphs, refer to unit-8 of MCS-21.

Session 8: Graphs (3 hours)

1. Write a program in ‘C’ language to implement Dijkstra’s algorithm.
 2. Write a program in ‘C’ language to implement Kruskal’s algorithm.
 3. Write a program in ‘C’ language to accept an undirected graph as input and print the list of all vertices in the Graph which are articulation points.
 4. Write a program in ‘C’ language which accepts a directed graph as input and prints all the strongly connected components of the Graph.
-

1.10 SEARCHING AND SORTING

For information on Searching, refer to unit-9 of MCS-21.

Session 9: Searching and Sorting (3 hours)

1. Write a program in ‘C’ language to implement linear search using pointers.
2. Write a program in ‘C’ language to implement binary search using pointers.
3. Write a program in ‘C’ language to implement Quick sort using pointers.
4. Write a program in ‘C’ language to implement Heap sort using pointers.
5. Write a program in ‘C’ language to implement 2-way Merge sort using pointers.
6. Write a program in ‘C’ language to implement Bubble sort using pointers.
7. Write a program in ‘C’ language to implement Topological sort using pointers.

1.11 ADVANCED DATA STRUCTURES

The advanced data structures that are dealt in MCS-021 are Splay trees, Red-Black trees and AA-trees.

For more information on the above mentioned data structures, refer to Unit-12 of MCS-021.

Session 10: Advanced Data Structures (3 hours)

1. Write a program in ‘C’ language to insert 15, 25, 2, 4, 3, 1, 50 into an initially empty Splay tree.
2. Write a program in ‘C’ language for the creation of a Red Black tree. Also, implement insertion and deletion operations.
3. Write a program in ‘C’ language for the creation of a AA-tree. Also, implement insertion and deletion operations.

1.12 SUMMARY

In this section, we discussed the data structures that are covered in the course material of MCS-021 briefly. Each discussion is followed by a lab session. The session includes programming problems for the learners. More stress has been made on the programming using pointers as it is regarded as a very special skill. It is very important to attend the lab sessions with the necessary homework done. This enables better utilization of lab time. The learner can execute more programs in a given amount of time if s/he had come with preparation. Else, the learner may not be able to execute programs successfully. If the learner had executed program successfully in lab without sufficient preparation, then, it is very important to assess the efficiency of the program. In most cases, the programs lack efficiency. That is, the space and time complexities may not be optimal. In simple terms, it is possible to work out a better logic for the same program.

1.13 FURTHER READINGS

- The C programming language by Brian W.Kernighan and Dennis M. Ritchie;

Lab Manual

Prentice Hall

- Data Structures and Algorithm analysis in C++ by Mark Allen Weiss; Pearson Education Asia

Web References

- http://www.isi.edu/~iko/pl/hw3_c.html
- <http://www.programmersheaven.com>

SECTION 2 OPERATING SYSTEM AND NETWORKING LAB

Structure	Page No.
2.0 Introduction	13
2.1 Objectives	14
2.2 Overview of Windows 2000	14
2.3 Overview of Unix and Linux	20
2.4 Summary	26
2.5 Further Reading	27

2.0 INTRODUCTION

The main operating systems involved in networking now a days are Windows 2000 and Linux. Both are having their own advantages and disadvantages. The combination of Linux and Apache makes a "strong" and "open" Web server platform, Linux works better on older, less powerful computer hardware because it requires less resources (memory or processing) as compared to Windows. The "free" nature of Linux also attracts some people/industries. Compared to Windows, Linux is virus-free and bugs-free. Windows 2000 provides the user a mature, familiar and interactive interface that is easy to learn and understand for Windows users (those are of course much higher than Linux users) and the high support of Microsoft also makes it popular among people. According to few surveys, Windows 2000 servers are less costly to run and maintain compared to Linux. But we think that a network administrator should have an expertise on both of these leading network operating systems. Even most of the network administrators are running both Linux and Windows on the server for the best networking.

This section provides you the discussions, demonstrations, and lab exercises to sharpen your skills and knowledge necessary to admin and support Windows 2000/Linux networking. It contains an overview of Windows 2000 and Linux/Unix in the beginning to develop your understanding of these operating systems. If you need any details you can always refer to the course material of MCS-022. Further, in this course you have an example to introduce you in the lab. Further, you have different lab exercises on Linux/Unix and on Windows 2000. We hope these exercises will provide you practice for administering, monitoring, and maintaining networks.

To successfully complete this section, the learner should have the following knowledge and skills prior to starting the section. S/he must have:

- studied the corresponding course material of MCS-022 and completed the assignments.
- proficiency to work with Windows/Unix interface
- knowledge of computer hardware components, including memory, hard disks, and central processing units
- knowledge of networking concepts, including network operating system, server-client relationship, and local area network (LAN).

Also, to successfully complete this section, the learner should adhere him/herself to the following:

- Before attending the lab session, the learner must have already written steps/algorithms in his/her lab record. This activity should be treated as home work that is to be done before attending the lab session.

- The learner must have already thoroughly studied the corresponding units of the course material (MCS-022) before attempting to write steps/algorithms for the problems given in a particular lab session.
 - Ensure that you include comments in your lab exercises. This is a practice which will enable others to understand your program and enable you to understand the program written by you after a long time.
-

2.1 OBJECTIVES

- To get familiarity with the basic operations of Unix, Linux & Window 2000.
 - Give exposure to network devices and configurations.
 - To get exposure to networking concepts using Unix, Linux & Windows 2000.
 - To perform advanced networking on Windows 2000
-

2.2 OVERVIEW OF WINDOWS 2000

Background to Windows

In 1981, Microsoft released their Disk Operating System which is popularly known as MS-DOS. It is a command line interface as shown in *figure1*, similar to Unix where you had to use your keyboard to run any command. But the users of MS-DOS have to remember these commands before using their computers.

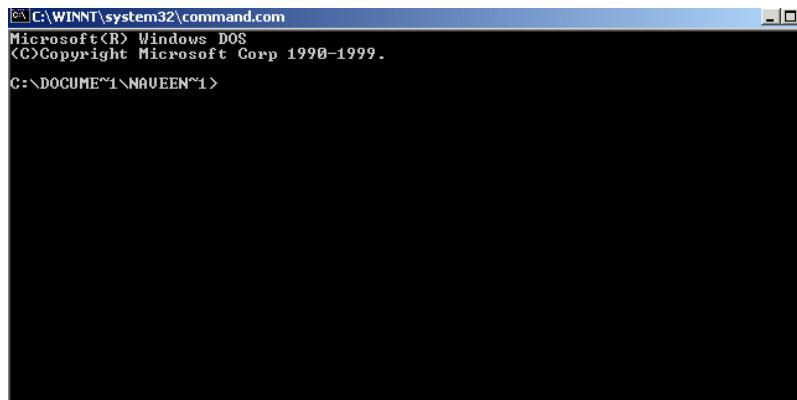


Figure 1: MS-DOS a Command Line Interface

Its functionalities are very limited, also it is time consuming and boring. Because of these reasons slowly Microsoft moved in the GUI-based operating system which enabled users to navigate their computer screen by using a cursor which they could use to “point and click” instead of having to type commands and they won the prize of leadership in the field of commercial operating systems. Window 95 was a very popular architect of Microsoft but they realized that Industries/people want a more secure, network-oriented, robust and powerful operating system and so about the same time, Windows NT 4.0 [*What's a full form of NT?*] was released which looked similar to Windows 95 but had its own strength. Windows NT was aimed at high-end users such as businesses where security and performance are more important than Windows 95’s graphical and sound capabilities. Windows NT (*Ok, I will tell you the full form of NT, NT stands for new technology*) made history again in the operating systems field. It became the standard for Network operating systems along with Unix/Linux. But still more specifications from industries were coming to upgrade and enhance the functionalities of Windows NT, which gave motivation for the development of Windows 2000. Windows 2000 brought together the high-end features of Windows NT with the home user features of Windows 95/98.

In Today's Network Administrators world, Windows 2000 is a popular network operating system (NOS). As you know it was built as a successor to window NT 4.0. It was also known as Window NT 5.0, but keeping in view computational needs of the new millennium Microsoft gave the new name Windows 2000 (which also maintains the naming series with Windows 95 and 98). Before starting your administrative tasks on it, you should know about its family. Windows 2000 family is similar to Windows NT as shown in *figure 2* given below except *Datacenter server*, which is a new family member.

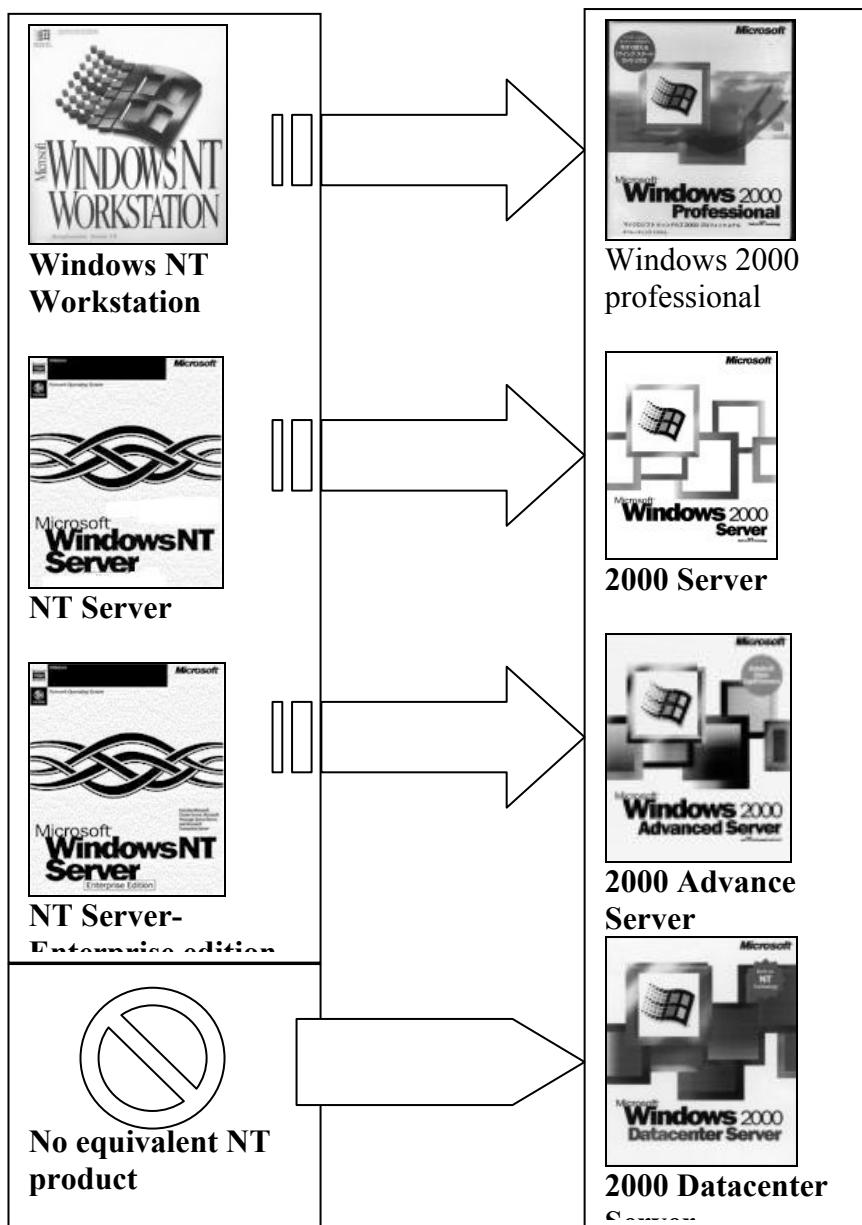


Figure 2: Similarities between Windows 2000 and Windows NT Family

As shown in the above figure Windows 2000 comes in four versions:

- Windows 2000 Professional
- Windows 2000 Server
- Windows 2000 Advances Server
- Windows 2000 **Datacenter Server**

Let me explain some of the important features of these Windows 2000 (W2K) products. If you want to know more about each product, you can open the Windows 2000 home page.

Windows 2000 Professional

Windows 2000 Professional is designed for desktop/laptop users within offices, academic institutions and other similar networked environments. It provides much greater control to network administrators than users. May be you are aware of the networking environment with Windows 95, which provides easy hands to users to change settings without any control. But with Windows 2000 professional users hand are limited to change settings.

However, Windows NT was the first NOS to boast of the same features for users as Windows 95. This has meant that many home users have also adopted W2K Professional as their OS of choice.

Windows 2000 Server

Windows 2000 Server is used on network servers, hosting a whole variety of network services often in conjunction with other pieces of Microsoft software such as:

- SQL Server – database
- ISA (Internet Security & Acceleration) Server – Internet access control
- IIS (Internet Information Server) – website hosting

It also controls who is using the network, how they can use it, provides shared file storage areas and controls connection to network printers. To the user it appears very similar to Windows 2000 Professional. However, it carries many features that desktop and laptop users would not require.

Windows 2000 Advanced Server

Windows 2000 Advanced Server includes all features of Windows 2000 Server but also boasts of additional features for organisations requiring larger scale processing like with additional scalability and reliability features, such as clustering, used by businesses for running transaction-intensive applications such as large e-commerce stores and running in the most demanding scenarios.

Windows 2000 Datacenter

Windows 2000 Datacenter Server is the most powerful and functional server operating system offered by Microsoft till now. Windows 2000 Datacenter Server is specifically designed for organisations which require the highest levels of availability and scalability. It is the best operating system for running mission-critical DBMS, ERP (enterprise resource planning) softwares, and high-volume real-time transaction processing.

Example: Installing and Configuring TCP/IP

We want to show one exercise to you so that you can understand what exactly you should do in your lab session. A very important thing is that before coming to the lab session what all you must prepare so you will not waste much of your computer time in labs. This demo exercise is about installation and configuration of TCP/IP.

Generally TCP/IP is installed as a part of the Windows 2000 set up process, but you can install it manually also.

You follow these steps one by one to install TCP/IP in your computer. If it is already installed you can Uninstall it (with the permission of Lab administrator) and then try to install it again.

1. Open the Network and Dial-Up Connections folder as given below in Figure 3:

Start>Setting >Network an Dial-up connections

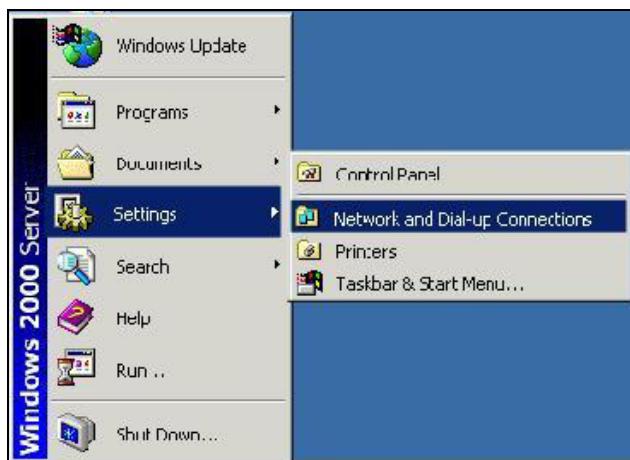


Figure 3: Opening Network and Dial-Up Connections Folder

2. As shown below in Figure 4, right click the Local Area Connection icon and choose the properties.

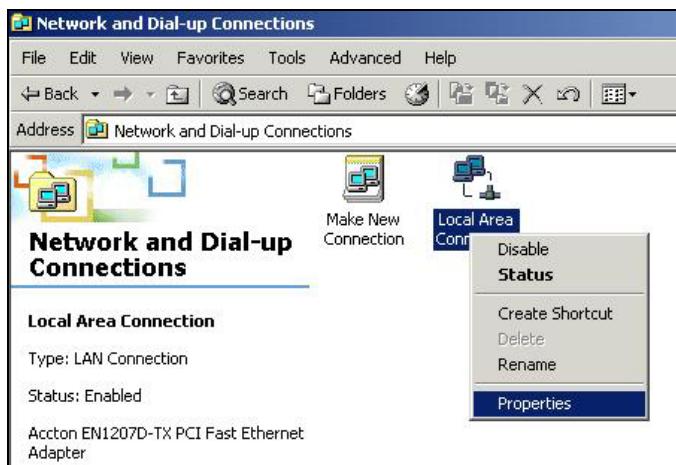


Figure 4: Opening Local Area Connection Properties

The Local Area Connection Properties dialog box appears, as shown in Figure 4. Now Click the **Install...** button, you will see the Select Network Component Type dialog box. Select protocol and click the **Add...** button as demonstrated in Figure 5.

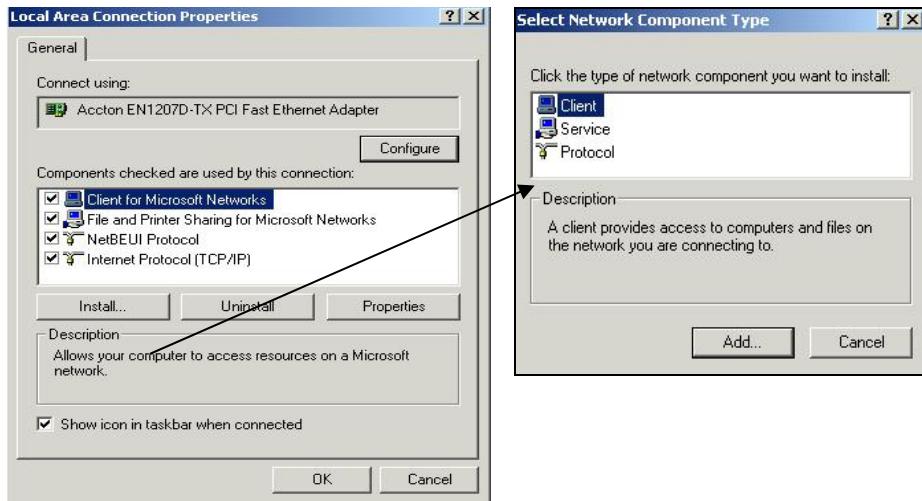


Figure 5: Select Network Component Type and Add Protocol

3. The Select Network Protocol dialog box appears in Figure 6. Choose Internet Protocol (TCP/IP), and then click the OK button.

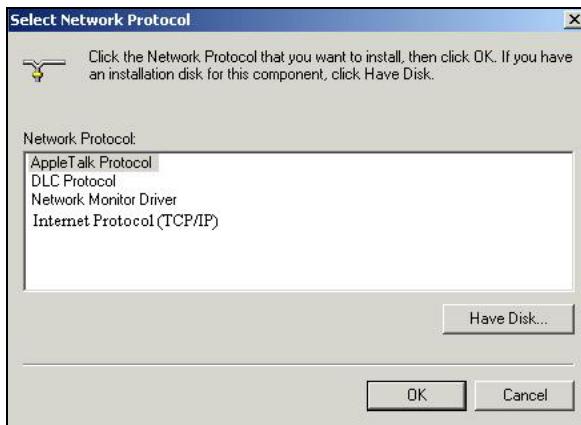


Figure 6: Select Network Protocol Dialog Box

4. If computer ask, insert your Windows 2000 CD and click OK.
5. Click the close button in the Local Area Connection Properties dialog box.

Remember, when you install TCP/IP, it defaults to using DHCP for automatic configuration. But it is always useful to know (specially for network administrator) how to manually configure a TCP/IP connection. Let us see the configuration of the example.

Configuring TCP/IP Settings

Follow these steps to add a second IP address to your existing NIC.

Note: In the given exercise you should assume that you're not using DHCP on the NIC, because you cannot assign additional addresses to a DHCP-enabled NIC.

1. First you select an IP Address, which is not currently in use by another device (e.g., I am selecting “192.162.6.142” in this example). Also ensure that you know the correct subnet mask to use with that IP address (e.g., 255.255.255.0 in this example).
2. Open the Network and Dial-Up Connections folder (Start>Settings>Network

and Dial-UP Connections) similar to the previous example right click the Local Area Connection icon and choose the properties command and Local Area Connect Properties dialog box appears as shown in given Figure 7.

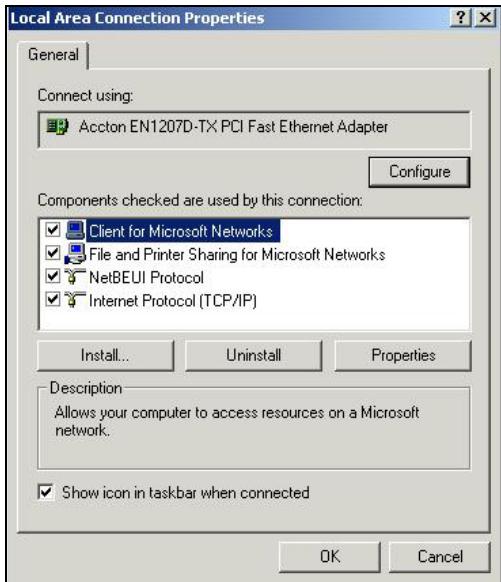


Figure 7: Local Area Connect Properties Dialog Box

3. Select Internet Protocol (TCP/IP) in the Components list, and then click the **Properties** button. The Internet Protocol (TCP/IP) Properties dialog box will appear. (Here in Figure 8 you can see that you can obtain IP address and DNS server automatically if you are a new client). We want to manually configure it so we will move to Advanced setting..

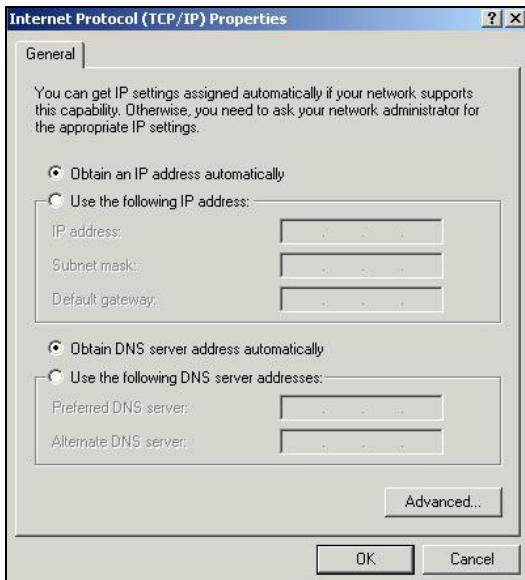


Figure 8: Internet Protocol (TCP/IP) Properties Dialog Box

4. The Advanced TCP/IP properties dialog box will appear when you click the **Advanced ...** button. The screenshot is given below in Figure 9.

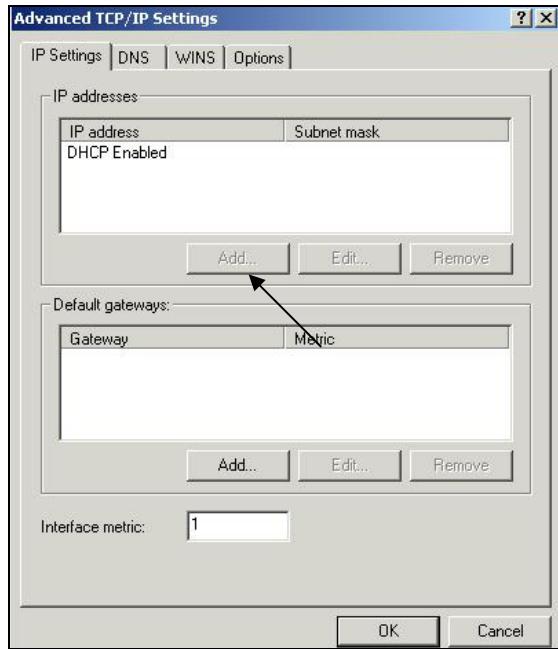


Figure 9: Advanced TCP/IP Settings Dialog Box

5. Click the **Add...** button in the IP Addresses control group. The TCP/IP Address dialog box will appear. In that you type the IP address and subnet mask we selected in Step 1.
6. Click the **OK** button in the Advanced TCP/IP Settings dialog box. Then similarly click the **OK** in the Internet Protocol (TCP/IP) Properties dialog box and in the Local Area Connection properties dialog box.

2.3 OVERVIEW OF UNIX AND LINUX

Even after thirty years of its creation Unix is still regarded as one of the most versatile, flexible and powerful operating systems in the computer world. Before you start swimming in your Unix shell, you must find out why people still regard as powerful. As you know (refer to MCS –011 lab) it was created at Bell Labs in 1970 written in the C programming language, which was developed at the same time. It supports large numbers of simultaneous users, runs with few alterations on many hardware platforms (provides some platform independence) and of course it was and is a simple, elegant, and easy to use (at least compared to its predecessors) operating system. In the early 1980s, the two strands of Unix development – AT&T and Berkeley –continued in parallel. The Berkeley strand got a major boost from Sun Microsystems, which used the Berkeley code as the basis for its Sun OS operating system. The AT&T strand was picked up by companies such as IBM and Hewlett-Packard.

Linux is a relatively new Unix flavor derived from the work of **Linus Torvalds**, who was interested to develop a Unix for academic use. He wrote the basics of the operating system himself. His work attracted followers around the world and, with cooperative effort, they wrote a complete Unix system. Early versions of Linux were primitive, but version 2 and above are powerful and efficient having an amazing range of features. Linux is one of the many versions of the Unix operating system. Working on Linux means working on one of the flavors of Unix. From our System Administrator's point of view, Linux has some striking advantages. The main advantage which separates its category from Unix is that **Linux is absolutely free**, you need not spend even the cost CD it can be entirely free downloadable from the Internet. (No registration fees, no costs per user, free updates, and freely available

source code). It is **portable** (means can be configured on any processor like Intel, Solaris, etc), **dual-bootable**, fast, reliable, secure and versatile. These properties make it popular among the System Administrators. While working on it you may realize many more important features and advantages of Linux. May be you will also contribute to development of Linux. Most of the exercises are command line based but similar exercises you can try on your GUI based Linux.

Session 1: Network Configuration

This session is your first introduction with Windows 2000. You can try different commands available in Windows 2000 for system and network administrator. Let us start:

Exercise 1: Run the following commands and write the use of each command:

ipconfig	ping	telnet	diskperf	netdiag
netstat	pathping	ftp/tftp	fc	sfc
nbtstat	rcp	lpr	tracert	verifier
nslookup	route	lpq	net session	drivers
nettime	rsh	chkdsk	hostname	net account

Exercise 2: Use **arp** command to find your Ethernet physical address.

Exercise 3: Modify the routing table using **ipxroute**.

//Use the Netsh.exe tool in Windows 2000 to perform the Exercise 4-9//

Exercise 4: View the TCP/IP settings.

Exercise 5: Configure interfaces

Exercise 6: Configure routing protocols

Exercise 7: Configure filters

Exercise 8: Configure routes

Exercise 9: Configure remote access

Exercise 10: Use **winchat** command and communicate with your friend sitting on a different machine of Windows 2000.

Session 2: Linux/Unix Operating Systems

In this session you will get introduced to Linux/Unix and you can perform different operations based on the course material you studied in MCS-022.

Exercise 1: First try to execute the following commands on your operating system and write down the results and use of each command.

- man (find manual help)
- cd
- ls, ls -a (try to find out other options of ls using man)
- cd .
- pwd
- cd ..
- ls -al
- ls -al | more
- cat passwd

- cd –
- chmod

We hope you will stop here and you will keep digging more and more commands but do it after the session.

Exercise 2: Try to explore the filesystem, write what is there in /bin, /usr/bin, /sbin, /tmp and /boot. Find and list the devices that are available in your system.

Exercise 3: Make your own subdirectories called uni and linu in your home directory, *Made?* Ok, now delete the subdirectory called uni.

Exercise 4: Create a file called ignou.txt that contains the words "hello I am student of IGNOU". Now copy this file and paste to other director. *Copied?* Can you move the file also from one directory to another?

Exercise 5: In the previous question you have a file *ignou.txt*; change its permission to rwxrwxr-x. You can try different possibilities to changes in its permissions. One possibility may be rwxr-xr-x permissions. Find out what are the different commands available that can be used to change the permissions of a file/files.

Exercise 6: Display the names of all files in the home directory using *find*. Can you display the names of all files in the home directory that are bigger than 500KB.

Exercise 7: Display a sorted list of all files in the home directory that contain the word ignou inside them. Hint: Use *find* and *grep* and *sort*. Can you use *locate* to find all filenames that contain the word ignou?

Exercise 8: Use *egrep* to try to find out which lines in an ignou.txt file are satisfied by the regular expression given: (^[0-9]{1,5}[a-zA-Z]+\$)|none and check the result with different combinations of lines.

Exercise 9: Change your password and write down the restrictions for given password.

Exercise 10: Open ignou.txt using vi editor, go to the end of the file and type in the following paragraph:

*In 1971 Bell Labs releases the first Unix **operating** system. Then 1985 Richard Stallman releases his GNU ("GNU is Not Unix") Manifesto thus starting the open source revolution. He wanted to **creat** an open-source version of Unix **Unix**. Stallman's Free Software Foundation eventually created the GNU General Public License (GPL) which is basically an anti-copyright also referred to as a*

Now you correct spelling errors in the first three lines and remove the extra "Unix" in the 3rd line of the paragraph. Add the words "*copyleft*" to the end of the paragraph. Replace the string "GNU is Not Unix" with a string "Unix is not a GNU". Save the file and quit. Repeat the same exercise with emacs also. Write down the differences between both editors ,also write which one you find easier and why.

Session 3: Linux/Unix Operating Systems

Exercise 1: Find the files (with full path) in your home directory those name are starting with the character ‘s’ and redirect the output into a file redirecting.txt and if you receive any error message on execution of the command redirect into errors.txt.

Exercise 2: Execute sleep 25 in the foreground, suspend it with Ctrl-z and then put it into the background with *bg*. show all process running in background, bring any process back into the foreground with *fg*. Repeat the same exercise using *kill* to

terminate the process and use & for sending into background. (You need to see different options of the kill command)

Exercise 3: Combine the commands *cat nonexistent* and *echo helloIGNOU* using suitable operators. Now reverse the order of the commands and try.

Exercise 4: Write a shell script, which returns the PID of a process and accept the name of the process.

Exercise 5: Use ping to find the round-trip delay to www.ignou.ac.in

Exercise 6: Send a message to all users which are online. Make provision so that you can send messages to other users but others cannot. Use talk to send messages.

Exercise 7: Print a file ignou.txt, and then send multiple files to a printer. Write the command you will execute to remove any file from print queue.

Exercise 8: Send a mail to yourself, and include ignou.txt inside the mail. Read the mail you have sent to yourself. Save the piece of message and file into some folder. Reply to yourself.

Exercise 9: Use telnet and ftp to get connected with other remote machine. Write the problems you encounter during connection with remote machine.

Exercise 10: Use the ls command and grep to display all names starting with "s".

Session 4: System Administration using Unix & Linux

Please do the following exercise with extra care when you are using the root account. If you have any doubt you must clarify it before executing any command.

Exercise 1: Use *finger* or *who* to get a list of users on the machine.

Exercise 2: Add different users, set their passwords and define permissions. Check whether you are able to change the passwords of all users or not.

Exercise 3: Delete the user, which just now you have added.

Exercise 4: Set the execution time of two jobs so that it can run automatically tomorrow one at 11:00 a.m. and another at 1:00 p.m. After this setting, how can you change the time of execution of the job?

Exercise 5: Try to access your account available at a remote machine. Download some file from that machine to your machine.

Exercise 6: Create a *cron* job that sends you a message after every 5 minutes.

Exercise 7: Restart any system daemon like the web server *httpd*.

Exercise 8: Write a message to inform all users that "they should shut down their machine after completing the lab exercises".

Exercise 9: Monitor the log time of users using xargs.

Exercise 10: Eliminate file names from all users home directories containing bad characters and whitespace.

Session 5: Windows 2000: Introduction to Networking

Exercise 1: Use different system tools and administrative tools. Write down the function of each tool in your lab notebook.

Exercise 2: Add different users and groups. Also configure their permissions.

Exercise 3: Install and configure a local printer

Exercise 4: Connect and configure your computer with a Local Network Printer.

Exercise 5: Install and Configure Windows 2000 Active Directory and Domain Controller.

Exercise 6: Create a Hierarchical Directory Tree.

Exercise 7: Share any folder available in your directory, also configure its share permissions for different users.

Exercise 8: Install and Configure TCP/IP.

Exercise 9: Install a caching DNS server and find out how it reduces the network traffic.

Exercise 10: Configure a DNS server as a root name server.

Exercise 11: Implement delegated zones for a Domain Name Server.

Session 6: Windows 2000: Server Management

Exercise 1: Install and Configure Windows 2000 Client

Exercise 2: Install and Configure Windows 2000 Server

Exercise 3: Set your printer on sharing and assign print permissions according to different users, configuring printer priorities for different groups.

Exercise 4: Install and Configure the DHCP Server Service.

Exercise 5: Configure Windows 2000 Client to use DHCP, DNS, and WINS.

Exercise 6: Configuring a Windows Client as a VPN Client.

Exercise 7: Implement Dfs (Distributed file system) replication.

Exercise 8: Install and configure Microsoft Certificate server (MCS).

Exercise 9: Install the Network Monitor Driver and show how to capture data with network monitor.

Exercise 10: Implement different kind of servers like File Server, Print Server, and Application Server. Learn different routine administration tasks for each kind of server.

Session 7: Windows 2000: Advanced Networking

Exercise 1: Implement different Groups in a Workgroup and in a Domain also.

Exercise 2: Show how you can enhance the feature and strength of file and print servers with Active Directory.

Exercise 3: Install the routing and remote access services for IP Routing

Exercise 4: Install the RIP and OSPF protocols.

Exercise 5: Configure web-based printer.

Exercise 6: Install and configure Terminal Services.

Exercise 7: Create a Remote Access Policy. Show how you can change the Remote Access Logging setting.

Exercise 8: Install the routing and remote access services as VPN server. Create a VPN Remote Access policy also.

Exercise 9: Install and configure a Web server.

Exercise 10: Create two global groups and configure so that users from both groups should be able to access some command folders.

Session 8: Windows 2000: Security

Exercise 1: Enable and configure IPsec policy on local computer.(also Enable and configure IPsec policy for an entire domain.)

Exercise 2: Protect client machine by using Internet Connection Firewall (ICF)

Exercise 3: Configure TCP/IP packet filter.

Exercise 4: Monitor the IP Routing status.

Exercise 5: Customize and configure IPsec policy and rules for transport mode on the local computer.

Exercise 6: Configure IPsec for tunnel mode. (Note: You need separate computers to which you have administrative access)

Exercise 7: Audit the IPsec logon activities and event. (Note: you can use two IP capable computers that are able to communicate to each other with there administrative access)

Exercise 8: Install the network monitor application. Show the use of capture filter and display filter with the help of your own examples.

Exercise 9: Configure PPTP packet filter such that it will block every packet stream except PPTP stream.

Exercise 10: Implementing Server Security by using Security Templates

Session 9: Windows 2000: Network Management

Exercise 1: Create a Group Policy Object (GPO) and Console.

Exercise 2: Configuring Software Deployment Settings.

Exercise 3: Configuring Remote and Removable Storage

Exercise 4: Setup the filter options for Advanced users and groups

Exercise 5: Backup and restore all files in a domain.

Exercise 6: Protect Data by Using Encrypting File System (EFS) and Recover Encrypted Data with a Data Recovery Agent.

Exercise 7: Establishing Intrusion Detection for Public Servers

Exercise 8: Configure the administrator account user profile to restrict the dial-up access.

Exercise 9: Use the *Registry Editor* to view and search for information in any registry. Show how to add a value in a registry. Save the registry to some text file.

Exercise 10: Enable network connectivity between NetWare, Macintosh, and Unix networks.

Session 10: Windows 2000: Troubleshooting

Exercise 1: Recover a Windows 2000 Server that Does Not Start

Exercise 2: Troubleshoot the "NTLDR Is Missing" Error Message in machine.

Exercise 3: What you should do when you find that the drive letter (e.g. c:/ drive, A:/ drive) changes after you restart your computer.

Exercise 4: Back up the recovery agent Encrypting File System (EFS) private key.

Exercise 5: Encrypt Files and Folders on a Remote Windows 2000 Server

Exercise 6: If you cannot print to a network printer after adding Internet connection sharing, how will you resolve it?

Exercise 7: When you install Modem, how to enable/disable call waiting on computer.

Exercise 8: If you are having trouble getting a dial-up connection and you want to change the modem speed or you want to check the modem's response how you will check do it. If you are having noisy channel and you are not able to connect write down the series of steps you will be following to detect and correct it.

Exercise 9: When you use a dial-up remote access service (RAS) connection to browse the Internet or to a private network, your computer may hang and return a Stop error: "Stop 0x0000000A". Resolve this problem.

Exercise 10: When you attempt to view a web page and receive an error message "Not accepting cookies", how will you resolve it?

2.4 SUMMARY

In this section you have studied a brief introduction of Windows 2000 and Linux/Unix and the differences between the two. It contains the history as well as the technical description of both network operating systems. It contains different lab exercises to provide you hand-on experience on Linux and Windows 2000. The starting session was about manual configuration of Windows 2000 components, the next two sessions you worked on Linux/Unix operating system, other sessions were about the Windows server management and advance networking includes virtual network and remote access. We have given the main focus on network security issue which is becoming

one of the major challenges for the network administrator now a days. In the last of the sessions you completed lab exercises on the troubleshooting of real time problems or most expected problems generally faced by network and system administrators. After completing all theses lab exercised and MCS-022 course, you will have not only theoretical knowledge but in depth hand on experience also which will definitely serve in real time networking. The next section you are going to study in this course will be related to MCS-023 Relational Database management system where you will design and implement different databases according to the specifications given in the lab exercises.

2.5 FURTHER READING

Following are the books and website references with you can use for learning more in details.

Books

- “The ultimate windows 2000 system administrator’s guide” by Robert Williams, Robert A. Williams, Mark Walla. Addison Wesley Publications.
- Unix concepts and applications by Sumitabha Das, Tata McGraw Hill Publications
- Windows 2000 the complete reference by Ivens, Kathy, Tata McGraw Hill Publications
- Inside windows 2000 server by Boswell, Techmedia Publications.
- A user guide to Unix system by Thomas, Tata McGraw Hill Publications.
- A practical guide to Linux by Sobell, Pearson Education Publication

Reference material and tutorials on web

<http://www.microsoft.com/windows2000/techinfo/default.asp>
<http://www.microsoft.com/windows2000/>
<http://www.tldp.org/>.
<http://members.fortunecity.com/pcmuseum/windows.htm>
<http://www.businessweek.com/1997/16/b352372.htm>
<http://www.perl.com/pub/a/2000/10/begperl1.html>.
<http://pegasus.rutgers.edu/~elflord/unix/intro.html>
<http://www.infosyssec.com/infosyssec/win2000.htm>
<http://www.yolinux.com/TUTORIALS/>
<http://www.unixcities.com/>
<http://www.greatcircle.com/tutorials/ieusa.html>

SECTION 3 DBMS LAB

Structure	Page No.
3.0 Introduction	28
3.1 Objectives	28
3.2 Introduction to MS-Access	28
3.3 Database Creation	36
3.4 Use of DBMS Tools/ Client-Server Mode	38
3.5 Forms and Procedures	40
3.6 Summary	43
3.7 Further Readings	43

3.0 INTRODUCTION

By now, you must have obtained the practical skills of several programming languages. However, when we want to create a secure, managed database application, we need not re-start from scratch and develop a huge system using a programming language, rather we use a database management system: an application software. This software allows us to create database, query, report and many more operations with database. This section attempts to provide you the basic skills of data organisation, including database creation, integrity enforcement, query formulation, forms and report creation, etc. You should write SQL queries as well as work using interface provided in software packages. For the present practical we have selected MS-Access. However, you must try to develop some applications using MySQL.

You must go through the MCS-023 courseware in order to get the best of those sessions. During the practical sessions you can make suitable assumptions if necessary.

3.1 OBJECTIVES

By the end of the practical sessions of this section, you will be able to:

- create databases using a user interface and SQL command;
- create integrity and constraints on databases;
- develop forms/reports using sample interface;
- write SQL queries; and
- provide a practical overview of advanced concepts like triggers, assertion, views, etc.

3.2 INTRODUCTION TO MS-ACCESS

This topic gives you an introduction to MS-Access and the basic components of MS-Access will also be discussed in this section. But before we look at the Access software and its capabilities, let us recollect what databases are, just go back to your school days, when you used to maintain different copies of your ‘Home Work Assignment’ and ‘Class Assignment’. In those copies on the first page you used to make the ‘Index’, which contained the headings as Serial no., Chapter, Date, and Remarks. And under those headings, the details of all the ‘Assignments’ we used to store. Why did we store these details? What was that? Was it a database? Index! You mean to say that ‘Index’ was a database? YES. *A database is a collection of data related to a particular topic.* Employee records in a file cabinet, a stamp collection in an album – each of these collections is nothing but a database. Database, typically

consists of a heading that describes the type of information it contains, and each row contains some information. In database terminology, the columns are called fields and the rows are called records. This kind of organization in a database is called a table. A database management system (DBMS) is a system that stores and retrieves information in a database. Data management involves creating, modifying, deleting and adding data in files and using this data to generate reports or answer ad-hoc queries. The software that allows us to perform these functions easily is called a DBMS.

Microsoft Corporation introduced a Relational Database management system for the windows platform in 1992 called MS-Access. Microsoft Access is a development environment used to create computer databases.

Start the MS-Access

For starting MS-Access you must have a licensed copy of it, which is available along with MS-Office Professional.

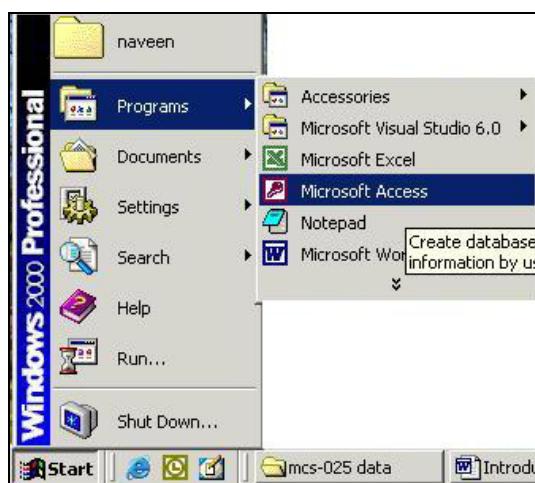


Figure 1: Starting MS-Access

After opening Access as indicated in *Figure 1* above, you will be presented with the Window shown in Figure 2. You can select one of the first two options if you are creating a new database, then go to the second option. If you want to edit an existing database, then go to the third option as shown in *Figure 2*.

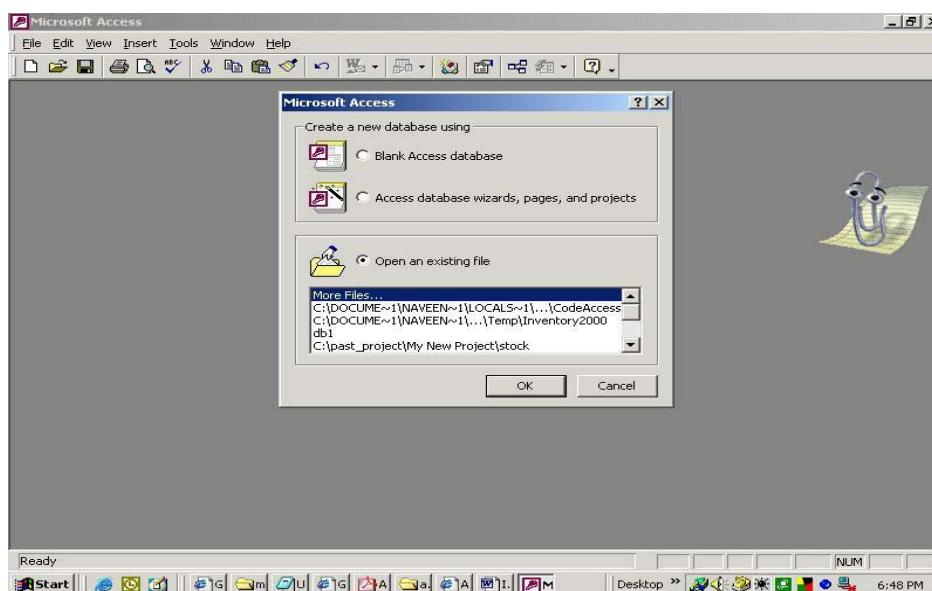


Figure 2: Starting an existing file or creating a new database

Open an existing database

If the database was opened recently on the computer, it will be listed on the main window (as shown in *Figure 2*). Highlight the database name and click OK.

Create new database

Unlike other office software, you must save an Access database before you start working on it. After selecting “Blank Access database”, you will first be prompted to specify a location and name for the database.

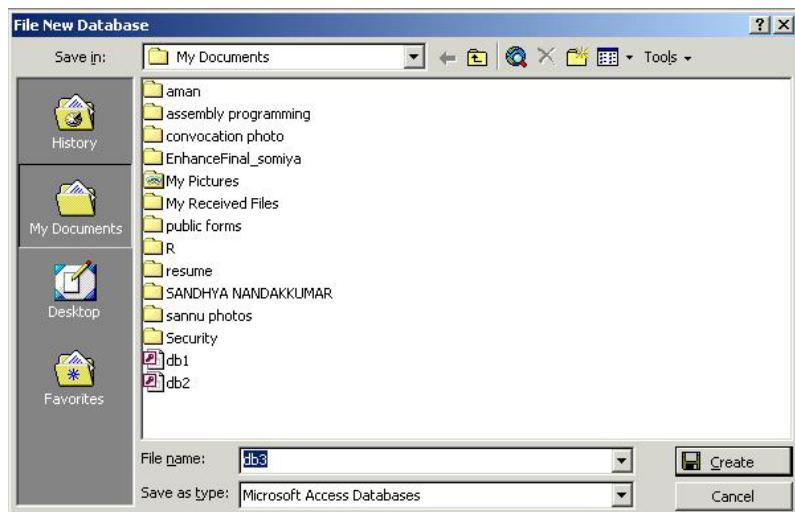


Figure 3: A sample Database Save Screen

You can select the folder where your database should reside and type the name of the database in the **File name** and click the **Create** button.

Database Components

The Database Window as shown below in *Figure 4* organizes all of main objects in the database like tables, queries, form and reports. Further in this we will discuss all these important components of database, which you will need in your lab exercises.

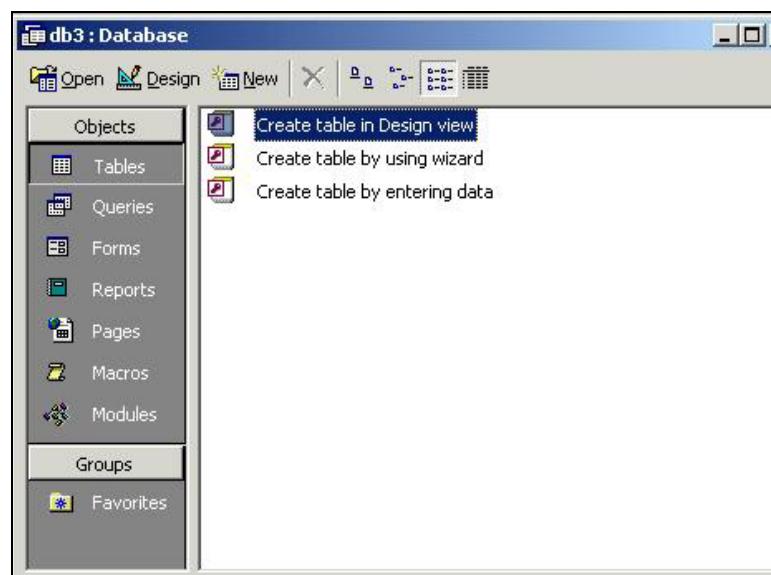


Figure 4: Database Components

Introduction to Tables

A Microsoft Access database is a collection of database files, which are also known as Tables. And each database (a table) is a collection of records, and a record is a collection of fields. You can also understand that the tables are a collection of cells that store information similar to the way an MS-Excel (If you don't know about Excel you can go and check it) worksheet does. MS-Access provides three ways to create a table.

1. Create table in Design view will allow you to create the fields of the table. (Design view is the best way for you).
2. Create table using wizard. (This is best when you are beginning to learn).
3. Create table by entering data, will give you a blank datasheet with unlabelled columns that looks much like an Excel worksheet.

Let us introduce you to Soft Garments, wholesalers for shirts, trousers, and T-shirts. They purchase from various manufacturers and wholesalers. The company has four departments – Sales, Accounts, Stores and Payroll. There are around 2000 employees working under the organization. The company wants to maintain a database, which will store the details and the entire information about all the employees. They want to store the Employee Code, Employee Name, Date of Birth, Date of Joining, Designation, Department and Photographs of the Employees.

Now, if the *Soft Garments* wants to store the employee details, they will have to make a table, which will be a part of some database. The information about one employee will make one record of that table, and the information will be stored under fields as shown in *Figure 5*, fields are *employee ID* and *first name* and others.

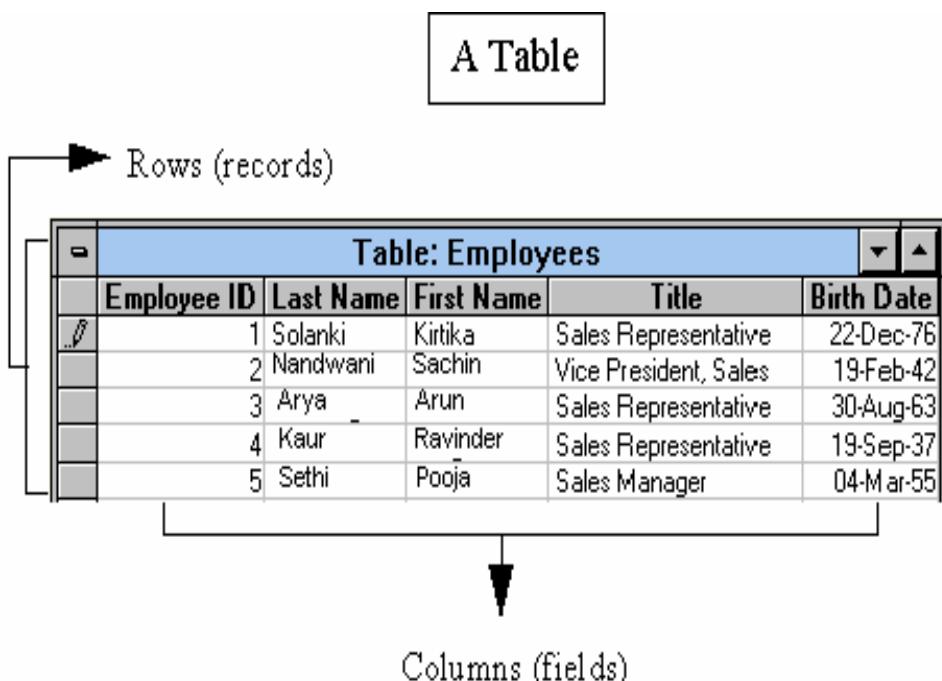


Figure 5: Records and fields of an Employees Table

Each record in a table contains the same set of fields and each field contains the same type of information for each record.

Introduction to Queries

Queries select records from one or more tables in a database so they can be viewed, analyzed, and sorted on a common datasheet. The resulting collection of records, called a dynaset (short for dynamic subset), is saved as a database object and can

therefore be easily used in the future. The query will be updated whenever the original tables are updated. Types of queries are select queries that extract data from tables based on specified values, find duplicate queries that display records with duplicate values for one or more of the specified fields, and find unmatched queries display records from one table that do not have corresponding values in a second table.

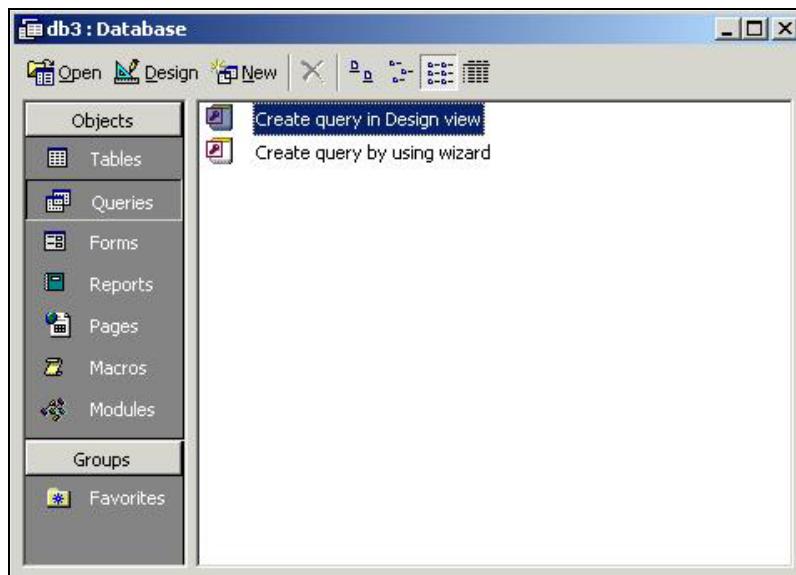


Figure 6: Creating Queries

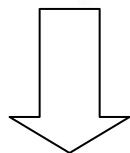
Assume that you are a senior executive in the *Soft Garments* and heading the payroll department. One day the manager of the company calls you, and wants to know how many employees are in 'A' grade. Will you be in a position to answer that Query, right at that moment? May be Yes, May be No. Keeping track of 2000 employees is quite difficult. Not to worry. The manager had a query, he asked you. If you don't know the answer, since you kept your data in database, you can ask the '*Query*' to your database.

In MS-Access, A Query is a question you ask about the data in your database. The data that answers the question can be from a single table or several – the query brings the information together.

For solving the above query asked by the manager you can write the following query in access *SQL* view. As shown in *Figure 7* after performing this query on the Employee table you will get the result showing details about employees who are in Grade A. In this example, you have very few employees listed but it is really helpful when the number of employees is huge like 2000 or 20000.

```
SELECT [Employees].[Grade], [Employees].[EmployeeID],
[Employees].[LastName], [Employees].[FirstName], [Employees].[Title]
FROM Employees
WHERE ((([Employees].[Grade])="A"));
```

	Employee ID	Last Name	First Name	Title	Grade
▶	1	Solanki	Kiritika	Sales Representative	A
	2	Naveen	Kumar	Sales Executive	A
	3	Akshay	Kumar	Sales Manager	B
	4	Shashi	Bhushan	Vice President,Sales	C



	Grade	Employee ID	Last Name	First Name	Title
▶ A		1	Solanki	Kiritika	Sales Representative
A		2	Naveen	Kumar	Sales Executive

Figure 7: Result of query performed on an Employee table

Forms and Reports

Forms are used as an alternative way to enter data into a database table. There are two ways in which you can view the data, stored in a table. Those ways are:

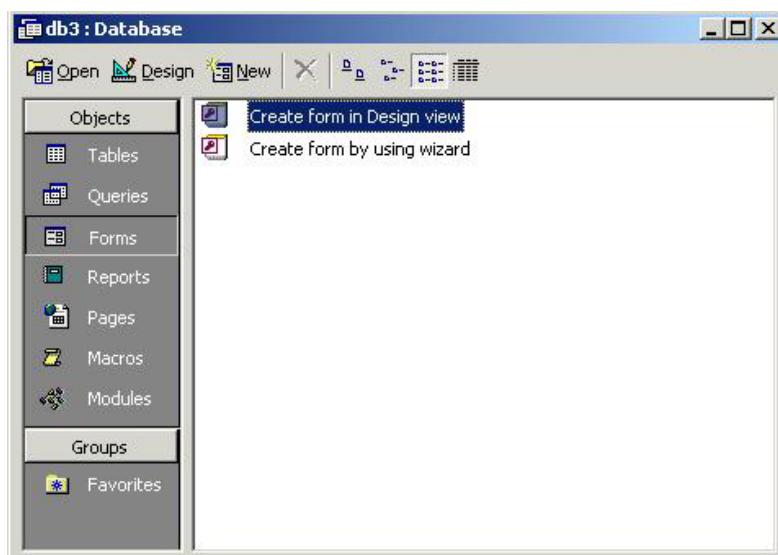


Figure 8: Creating Forms

Create Form by Using Wizard

To create a form using the assistance of the wizard, follow these steps:

Click the Create form by using wizard option on the database window. From the Tables/Queries drop-down menu, select the table or query whose datasheet the form will modify. Then, select the fields that will be included on the form by highlighting each one, the Available Fields window and clicking the single right arrow button > to move the field to the Selected Fields window as shown in *Figure 9*. To move all of the fields to Select Fields, click the double right arrow button >>. After the proper fields have been selected, click the Next button to move on to the next screen.

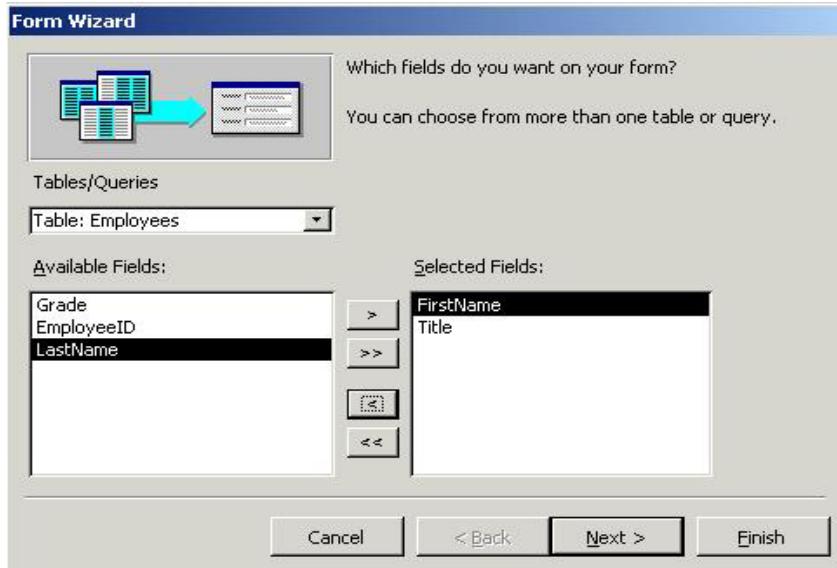


Figure 9: Create Employee Form by Using Wizard

Afterwards select the layout and visual style for the form from the next set of options and click Next. On the final screen, name the form in the space provided. Select “Open the form to view or enter information” to open the form in Form View or “Modify the form’s design” to open it in Design View. Click Finish to create the form.

Create Form in Design View

To create a form from scratch without the wizard, follow these steps:

Select “Design View” and choose the table or query the form will be associated with the form from the drop-down menu. Select View | Toolbox from the menu bar to view the floating toolbar with additional options. The toolbar contains different controls as shown in *Figure 10*.

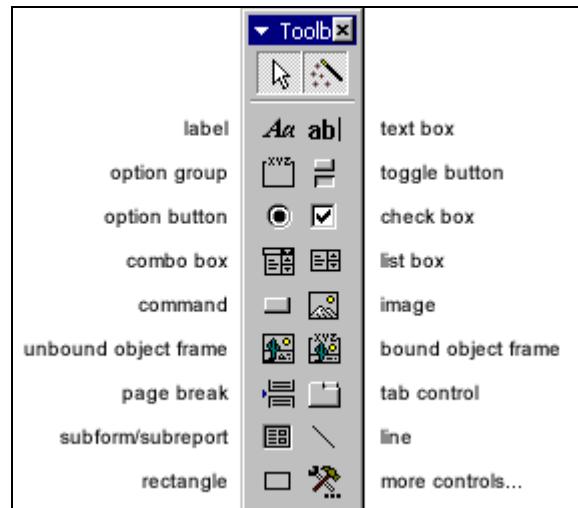


Figure 10: Different controls in Toolbar

As shown in *Figure 11* you can add controls to the form by clicking and dragging the field names from the Field List floating window. Access creates a text box for the value and label for the field name when this action is accomplished. To add controls for all of the fields in the Field List, double-click the Field List window’s title bar and drag all of the highlighted fields to the form.

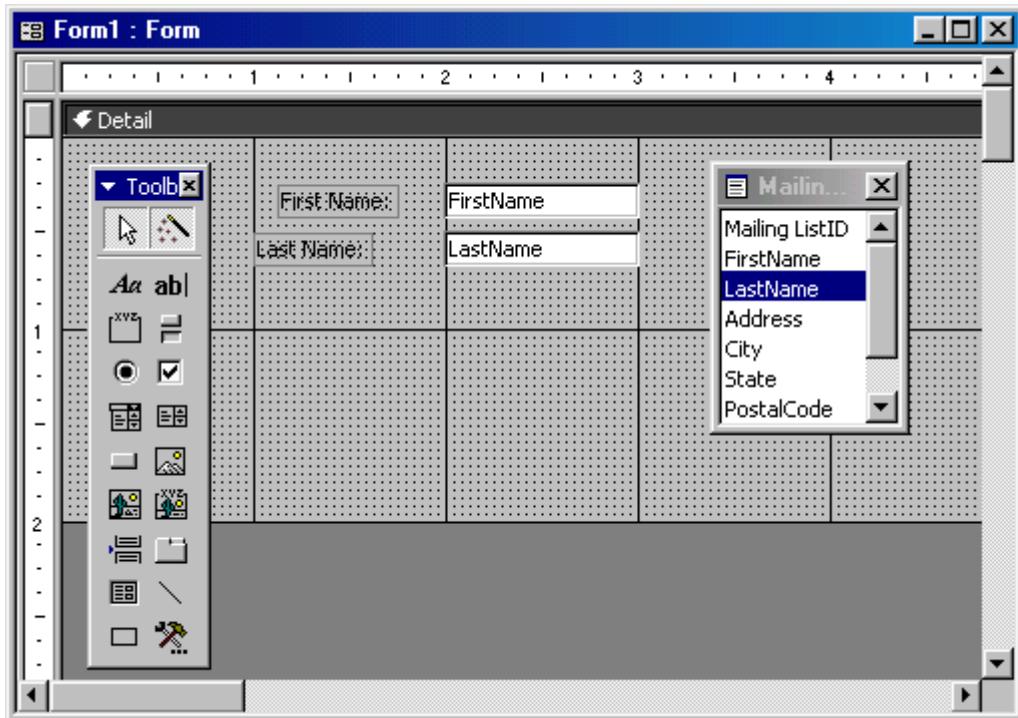


Figure 11: Adding controls to the form

A form is a customized way of viewing, entering and editing records in a database. You can specify how data is to be displayed when you design the form. Form can be created to resemble more closely the way data would be entered on paper form so that the user feels familiar with the operation.

Report

Forms and Queries present the data on screen. Reports are used to present data on printed-paper. It provides a way to retrieve and present data as meaningful information, which might include totals and grand totals, which have to be shown across an entire set of records. Similar to Form in Reports creation also Access provides two ways for report creation. As shown in *Figure 12* you can select any way of report creation. For example in *Figure 13* you can see a report showing summary report of employee sales and category sale.

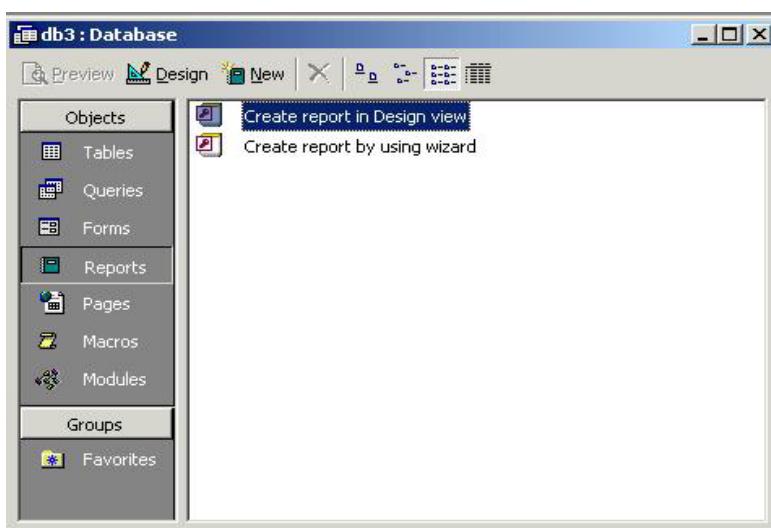


Figure 12: Creating reports

A Report		
Sales Summaries		
Employee Sales:	Employee Name:	Employee Sales:
	Pooja Sethi	47,674
	Sachin Nandwani	59,687
	Anu Solanki	93,932
Category Sales:	Category Name:	Category Sales:
	Beverages	127,189
	Condiments	56,462
	Confectionaries	88,500

Figure 13: A sample report

3.3 DATABASE CREATION

In the next section let us do some exercises relating to DBMS. The sessions are structured for your benefit.

Session 1: In this session you need to create database for an Employee management system of an ABC organisation. The details about different tables are given below. According to that you can proceed further and create tables using MS-Access.

Create the following tables with the specified constraints:

Employee

First name	-	Not NULL
Middle initials	-	
Last name	-	Not NULL
Employee-id	-	Primary Key
Date of Birth	-	
Address	-	
Gender	-	M or F
Salary	-	Range of 5000 to 25000
Date of Joining	-	
Department number	-	Refers to Department Number of Department table.

Department

Department name	-	Not NULL unique
Department number	-	Primary Key
Manager_id	-	Refers to employee-id of employee table.
Manager date of joining	-	Not NULL.

Department location

Department number	-	Refers to Department number of department table.
Department location	-	Not NULL.
Department number & Department location are combined Primary Key.		

Project

Project name	-	Not NULL.
--------------	---	-----------

Project number	-	Primary Key.
Project location	-	Not NULL.
Department number	-	Refers to department number of Department table.

Works-on

Employee-id	-	Not NULL refers to employee-id of employee table.
Project number	-	Not NULL refers to Project number of Project table.
Hours	-	Not NULL.
Employee-id & Project number are combined primary key.		

Dependent

Employee-id	-	Refer to employee table employee id field
Dependent name	-	
Gender	-	M or F
Date of Birth	-	Not NULL
Relationship	-	Not NULL

Now enter a few sets of meaningful data and answer the following queries.

1. List the department wise details of all the employees.
2. Find out all those departments that are located in more than one location.
3. Find the list of projects.
4. Find out the list of employees working on a project.
5. List the dependents of the employee whose employee id is '001'

Session 2:

This session is similar to the previous one, but in this session assume that you are developing a prototype database of the IGNOU library management system, for that you need to create the following tables:

- (a) Book Records
- (b) Book details
- (c) Member details and
- (d) Book issue details

Structure of the tables are given below:**Table Name Attribute Name**

Book Records Accession Number
 ISBN Number

Books ISBN Number
 Author
 Publisher
 Price

Members Member Id
 Member Name
 Maximum Number of books that can be issued
 Maximum Number of days for which book can be issued

Book Issue Member Id
 Accession Number

You must create constraints, including referential integrity constraints, as appropriate. Please note accession number is unique for each book. A book, which has no return date, can be considered as issued book. Enter suitable data into the tables. Now answer the following:

1. Insert data in all the three tables (use insert).
2. Insert appropriate description associated with each table and the column (use comment).
3. Display the structure of the tables.
4. Display the comments that you have inserted.
5. Using SELECT statement, write the queries for performing the following function:
 - (a) Get the list of all books (No need to find number of copies)
 - (b) Get the list of all members
 - (c) Get the Accession number of the books which are available in the library
 - (d) On return of a book by a member calculate the fine on that book.
 - (e) List of books issued on 01-Jan-2005
 - (f) Get the list of all books having price greater than Rs. 500/-
 - (g) Get the list of members who did not have any book issued at any time.
 - (h) Get the list of members who have not returned the book.
 - (i) Display member ID and the list of books that have been issued to him/her from time to time.
 - (j) Find the number of copies of each book (A book accession number would be different but ISBN number would be the same).
 - (k) Find the number of copies available of a book of given ISBN number.
 - (l) Get the member ID and name of the members to whom no more books can be issued, because they have already got as many books issued as the number for which they are entitled.

3.4 USE OF DBMS TOOLS/ CLIENT-SERVER MODE

Session 3:

This session is based on Session 2 where you have created a library management system. In this session you have different **query specification**. You must create appropriate forms, reports, graphs, views and data filtering, use of multilevel report, etc. to answer these queries.

1. Get the list of ISBN-Number, Book name, available copies of the books of which available copies are greater than zero.
2. Get the list of ISBN-Number, Book name, Total copies, available copies of the book of which available copies are greater than zero. List should be displayed in alphabetical order of book name.
3. Get the list of ISBN number, Book name, Author, total copies, cost (cost is price \times total copies). List should be displayed in descending order of cost.
4. Get the list of books issued to each member.
5. Write query to know the maximum and average price of the books.
6. Get the list of all existing members and the number of days for which a member is allowed to keep the book. Also find out the members who have got the maximum number of books issued.
7. Get the list of *member codes* of those members who have more than two books issued.

8. Find the details of the books presently issued to a member.
9. Create the history of issue of a book having a typical accession number.
10. To set the width of the book name as 35.

Session 4:

Create the following table and perform the necessary tasks defined below one by one.
 You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ using client/server wherever needed.

1. Create the following table named **customer**

Column name	type	size
Customer id	Character	10
Name	Character	25
Area	Character	3
Phone	Numeric	7

Insert the appropriate data into table.

- a. Update Phone numbers of all customers to have a prefix as your city STD Code
 - b. Print the entire customer table
 - c. List the names of those customers who have ‘e’ as second letter in their names.
 - d. Find out the Customer belonging to area ‘abc’
 - e. Delete record where area is NULL.
 - f. Display all records in increasing order of name.
 - g. Create a table temp from customer having customer-id, name, and area fields only
 - h. Display area and number of records within each area (use GROUP by clause)
 - i. Display all those records from customer table where name starts with ‘a’ or area is “abc.”
 - j. Display all records of those where name starts with ‘a’ and phone exchange is 55.
2. Answer the following queries using Library system as created earlier. You must create a view to know member name and name of the book issued to them, use any inbuilt function and operators like IN, ANY, ALL, EXISTS
 - a. List the records of members who have not been issued any book using EXISTS operator.
 - b. List the members who have got issued at least one book (use IN / ANY operator).
 - c. List the books which have maximum Price using ALL operator.
 - d. Display Book Name, Member Name, Issue date of Book. Create a view of this query of the currently issued books.
 3. Create a table of Employee (emp-number, name, dept, salary) and Department (dept number, dept name). Insert some records in the tables through appropriate forms having integrity checks. Add some records in employee table where department value is not present in department table. Now answer the following query:
 - a. Display all records from employee table where department is not found in department table.
 - b. Display records from employee table in a report format with proper headings. This report must also contain those records where department number does not match with any value of department table.

- c. Display those employee records who have salary less than the salary of person whose empcode= ‘A100’.
- d. Create another table : Sales_data (Region-code, City, Salesperson-code, Sales-qty).
- e. Display records where salesperson has achieved sales more than average sales of all sales persons of all the regions.

Session 5:

For the following queries use Library System as created by you in earlier sessions. You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ using client/server wherever needed.

1. Get the list of books presently issued to the members, along with the names of the book as well as names of the members.
2. Get the list of the members who
 - (a) are entitled for more books than that the entitlement of member name “abc”.
 - (b) are issued the books for more days than the number of days for “abc”.
3. Find out the history of issuing of a list of books that has been identified during inspection as damaged books. (Create the necessary tables if needed).
4. Create the tables Item master and Transaction having following format:

<u>Item Master:</u>	<u>Transaction:</u>
Item-code	item-code
Item-name	Quantity
Price	Date of transaction

Set the foreign key constraints in the tables and insert at least 5 records having meaningful data. Now answer the following queries.

- a. Display Item-code, Name, Quantity, Date of transaction, where Sales amount = Quantity *Price using a report.
- b. Display all transactions of item ‘X’ using a report.
- c. Display all the items whose price is more than the price of item ‘X’.
- d. Saving the previous query into a temporary file.
- e. Store the database in a new file.
- f. Create all the forms for data entry and create at least 5 meaningful reports.

3.5 FORMS AND PROCEDURES

This topic covers design and implementation of different kinds of forms to create user interactivity. Also, you can design different procedures/triggers to perform different operations on databases.

Session 6:

1. Create the following tables:

Order party: (Order number, Order date, customer code)
Order: Order number, Item code, Quantity

The key to the second table is order-number + item-code
Create a **form** for data entry to both the tables.

2. Create a form for storing Bio-data of students. Create the supporting tables to store the data.

3. Design a suitable form for storing basic information and salary details of employees of an organisation. Design and implement the necessary tables.

Session 7:

1. Write a procedure/trigger on department code so such that the validity of the code is checked and the name of department is automatically displayed on selection of department code. Assume, design and create the necessary tables and constraints.
2. Write a procedure/trigger on a numeric field named value1 to check if the entered value is 1 (Married) or 2 (Unmarried). In case, the entered value is 1 (Married) then the control should pass to a field named ‘spouse name’ or else it goes to a field named: Father’s Name.
3. Employee code must begin with ‘P’ (Permanent) or ‘T’ (Temporary) and its second character must be a digit. Write procedure/trigger to check if the entered value is correct.
4. Write a procedure/trigger to generate Order Number automatically in any of the order tables created in Session 6.

Session 8:

1. Design a form that shows the status of books pending on a member on entering the member-id.
2. Design a **form** that modifies the records of an Item Table having the fields: Item Code, Item Name, Quantity, Price, Re-order Level.
 - (a) Enter the Item Code and get all the details from the tables
 - (b) Check if negative values are entered in the field.
3. Design the **form** to display the leave information of each employee following. The validations must be made for the fields:
 - Leave information of every employee must be display grouped by month
 - Display total of all leave taken.

Let us now perform all the operations you have practiced till now. You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ procedures/ using client/server wherever needed.

Session 9:

1. Add one more table employee with fields employee-number, employee-name, Basic pay, Department in the Library management system.
2. Add a new column Date of Joining in the table.
3. Modify the length of field employee name.
4. Delete the column basic from basic pay.
5. Find the details of members who were issued a book prior to Feb 1st 2005.
6. In previous query 5, list the details of the particular members.
7. In previous query 5, list the details of only two such members.
8. List the details of the persons who have been issued at least one book.
9. List the names of three persons who have not been issued any book.
10. List of members, who are entitled for 5 books or are issued the books for 15 days.
11. List the names of members in fixed length of 30 characters followed by their codes in parenthesis and with first character of the name in capital.
12. Find the list of the members who have been issued the books having the same ISBN number.

13. Display book issue/return data of various books in the following form

Book Accession number.	Book Title	Issued on
------------------------	------------	-----------

Book Accession number.	Book Title	Issued on	Returned on
------------------------	------------	-----------	-------------

Session 10:

1. Create the following tables for a video library after normalizing these tables:

Customer

Customer_id	Primary Key	Not NULL
Name		Not NULL
Area		
Phone_number		

Movie

Movie_no	Primary Key	Not NULL
Title	<film title>	Not NULL
Type	Action or Thriller or Romance or Comedy or Suspense or Horror etc.	
Actors	Not NULL	
Rent-Price	Not NULL	
Rent applicable	data part of primary key	

Issues

Issue_no	Primary Key	Not NULL
Movie_no	Refers to Movie_no of movie table	
Customer_id	Refers to Customer_id of Customer table	
Issue_date	not greater than current date.	
Return_date	not greater than current date.	

Write down SQL statements to perform the following Queries:

1. List the names of all the customers.
2. Print the entire customer table.
3. List the name and area of all the customers.
4. List the various movie types available.
5. List the names of all customers having 'i' in any case as the second letter in their names.
6. List the names of all customers that begin with 's' or 'j'.
7. Print the list of employees whose phone numbers have area code as 011.
8. Print the information of customers who have been issued movies in the month of February.
9. List the movies that have been issued to the customers with customer-id between '9000' and '9999'.
10. List the names of movies whose Rent - price is greater than Rs. 100/-.
11. Increase the Rent-price of each movie by 10%. Modify rent applicable data suitably.
12. List the movies in sorted order of their title, and types of all the movies except Drama.
13. Find the recovery made from each movie.
14. Calculate the total revenue of all movies.
15. Determine the maximum and minimum movie prices and Rename the title as Maximum Price.
16. List the Movies which are issued to customers for more than a week.
17. Print the type, average price, total number of prints, for each type of movie.
18. Find out the movies issued to customer 'X'.

19. Find out the names of the movies that have been issued to the maximum number of customers.
 20. Display the month in which customers take the maximum number of movies.
 21. Display the history sheet of each movie.
 22. List the customers who have not been issued any movie in the last 6 months.
-

3.6 SUMMARY

This section has provided you with problems with respect of creation of database and integrity using constraints and using an interface and also using SQL commands. Some of the exercises provided include creation of forms and reports, creation of SQL queries and an overview of various databases related concepts. We hope by now you must be familiar with at least one database application and would be able to migrate to other DBMSs.

3.7 FURTHER READINGS

Reference Books

- MS-Access user guide.
- Microsoft Access 2000 Bible by Cary Prague and Michael Irwin, IDG Books.
- Access 2003 Bible by Cary N. Prague, Michael R. Irwin, Jennifer Reardon; John Wiley & Sons publication.

Web reference and tutorials

<http://mis.bus.sfu.ca/tutorials/MSAccess/tutorials.html>
<http://www.aspfree.com/c/b/Microsoft-Access/>
<http://netforbeginners.about.com/od/msaccess/>
<http://www.vbtutor.net/vbtutor.html>
<http://www.w3schools.com/sql/default.asp>
<http://sirius.cs.ucdavis.edu/teaching/sqltutorial/>

SECTION 4 JAVA PROGRAMMING LAB

Structure	Page No.
4.0 Introduction	44
4.1 Objectives	44
4.2 Programming with Java	45
4.3 PATH and CLASSPATH Setting	46
4.4 Example Programs	51
4.5 List of Lab Assignments	52
4.6 Summary	55
4.7 Further Readings	56

4.0 INTRODUCTION

Only at the age of ten, Java became master of programming languages. Its interesting success has made Java the fastest growing programming language ever. It is a bouquet of different programming flowers, having peculiar smells, merging the beauty of all programming languages. You must work with this language to enrich your skill set and to become an expert programmer.

In this section, a brief introduction to Java is given to understand the strength of the language you are going to work with. However, if you want to know something in detail you can always see the corresponding course (MCS-024). We have already explained the solution of some obvious problems that you may encounter in the first session while compiling and interpreting your first java program. We have also explained the compilation and interpretation of example programs using of freely available software called *editplus*.

In the end, session wise problems are defined. These problems you should complete properly before moving to another session. During this section you should learn how to design and develop good quality of Java applications/ applets rather than simple scribbling of code.

To successfully complete this section, the learner should adhere to the following:

- Before attending the lab session, the learner must have already written algorithms and programs in his/her lab record. This activity should be treated as home work that is to be done before attending the lab session.
 - The learner must have already thoroughly studied the corresponding units of the course material (MCS-024) before attempting to write algorithms and programs for the programming problems given in a particular lab session.
 - Ensure that you include comments in your program. This is a practice which will enable others to understand your program and enable you to understand the program written by you after a long time.
 - Ensure that the lab record includes algorithms, programs, I/O and complexity (both time and space) of each program.
-

4.1 OBJECTIVES

After going through this lab section you will be able to:

- compile and interpret Java programs in DOS and *editplus*;
- writes 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 and interface;
 - handling exceptions arising in programs;
 - use of multithreading in programs;
 - work on strings;
 - use GUI components in your programs;
 - implement Sockets; and
 - connect databases with your programs.
-

4.2 PROGRAMMING WITH JAVA

The future of computing will revolve around the Internet. Java is a programming language developed by Sun Microsystems to take care of Internet computing requirements. Java is a platform independent language, that is why it is very popular for cross – platform applications and programming on **Word Wide Web (WWW)**.

Java is an Object Oriented Programming Language, which serves the purpose of Object Oriented Paradigm of Programming. An object oriented language uses the concept of abstraction, encapsulation, inheritance, and polymorphism to provide flexibility, modularity, and reusability for developing software. The following features of Java make it popular and a very useful programming language:

Platform Independent: Java programs can run on any machine and operating system that support Java Virtual Machine as we are showing in *Figure 1* where we have shown that a program after compiling converts into a byte code which is able to execute on any platform Windows/Linux/Macintosh.

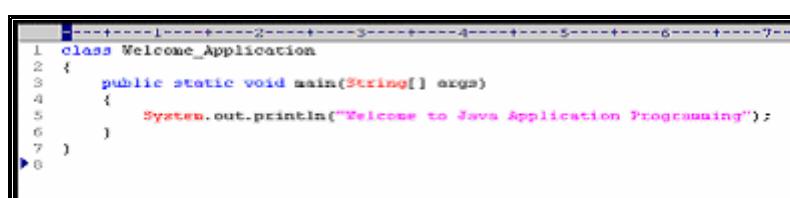
Multithreaded: These capabilities of Java provide the capability to single program to perform several tasks simultaneously. Multithreading is very useful for developing applications like animation, GUI, and networks. Unlike other programming languages, multithreading is integrated to Java. In other programming 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 the Internet. Java provides strong networking features to write distributed programs.

Secure: The design of Java has multiple layers of security which ensure proper access to private data and control over access to 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 for 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 Welcome_Application
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Welcome to Java Application Programming");
6     }
7 }

```

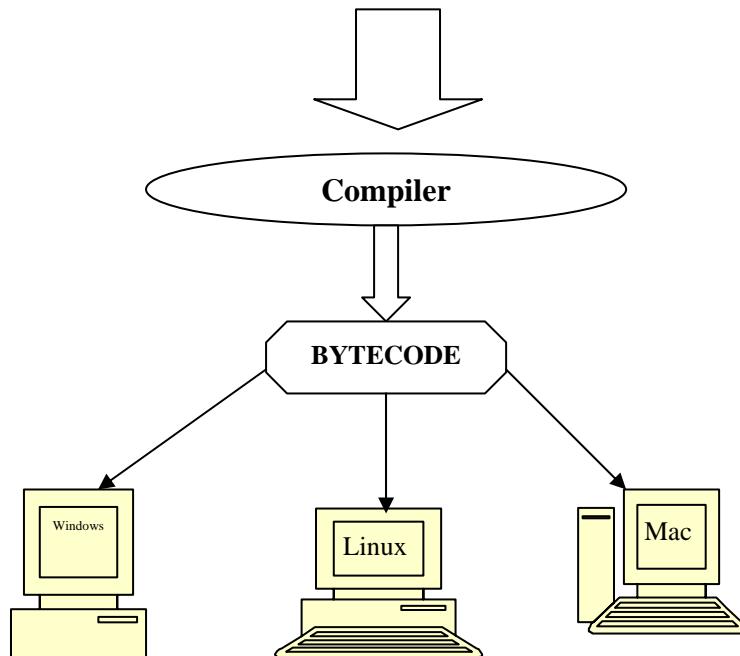


Figure 1: Execution of a Sample Java Program

Now let us see how you will run your Java programs...

4.3 PATH AND CLASSPATH SETTING

To run Java programs you need JDK (Java Development Kit) installed on your machine. The latest version of JDK you can download from java.sun.com for free. It is suggested that you should set PATH and CLASSPATH properly before trying to compile and run your Java programs, otherwise you may not able to compile or run your program because of improper setting.

PATH Setting

If PATH setting is not proper then whenever you try to compile any program on your DOS prompt as shown in *Figure 2*, it does not compile your TestProg program.

```

C:\WINNT\system32\command.com
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

C:\DOCUME~1\NAVEEN~1>javac TestProg.java
  
```

Figure 2: Compile TestProg.java

Instead it gives you some strange message about a “**Bad Command**”. What’s wrong? Can you guess? You must be thinking that you did not install JDK correctly in your system but this is not correct answer. The machine says “**Bad command**” because it recognizes the commands **javac** or **java**. So you simply need to tell DOS that these commands live in Java’s JDK directory and whenever it doesn’t recognise a command like **javac**, it should also check the Java’s JDK directory for a possible interpretation of this command. This problem can be solved in two ways, one by using DOS prompt, and another by using **My Computer** on your window screen.

Type path at your DOS prompt as demonstrated in *Figure 3*, you will get a PATH statement like

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

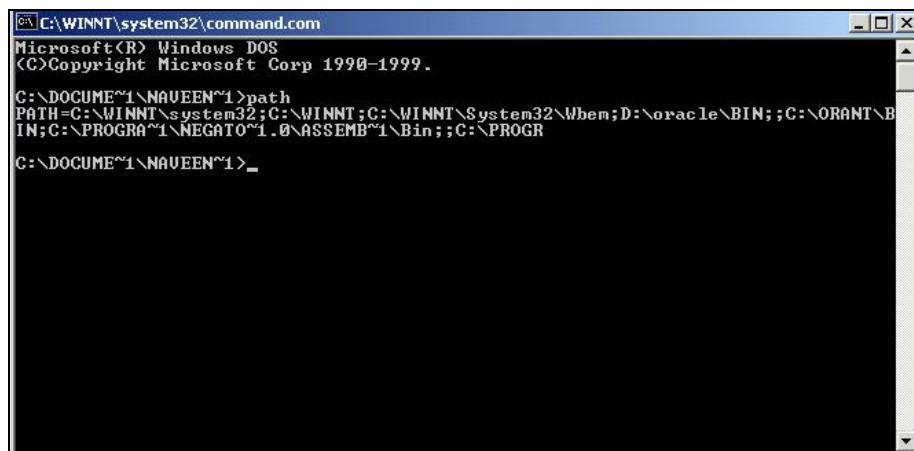


Figure 3: Showing PATH Statement

Here the dots represent all kinds of directory names. This is a list of all the places DOS looks when it is 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%
```

Whatever is the version of JDK, the above command adds your JDK directory 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 the required modifications as told here.

```
C:\copy autoexec.bat autoexecBACK.bat
```

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 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 **autoexecBACK.bat** back into **autoexec.bat** and try the whole process again.

CLASSPATH Setting

You may also need to set the CLASSPATH variable, which contains all of the directories on your machine where the Java compiler and the Java run-time environment are to search for any files it needs, e.g., if you compile or run the Java

class **TestProg.java**, the compiler needs to know that directory. Your autoexec.bat file or control settings also determine where the javac compiler looks for the Java code files. For example, if you are writing your code in the directory **c:\jdk1.2\naveen**, you would add the following line to your **autoexec.bat** file:

CLASSPATH: c:\jdk1.2\naveen

You can list (separated by ;) as many directories as possible in your classpath. One important thing about "." is that it means that javac will always look in the current directory for the files it needs. For example, to list the current directory, the naveen directory, and the diskette drive as places where javac should look, you would have the following classpath:

CLASSPATH: .;c:\jdk1.2\naveen;a:

This environmental variable can be set the same way as the PATH above. But there is one shortcut to avoid the CLASSPATH setting. The javac or Java command has an option that is (classpath option) that allows you to manually configure the classpath during compilation/execution. For example, to specify that the compiler must look in the current directory (that is, the "." directory) for the file TestProg.java when compiling it, you would use the command:

javac -classpath . TestProg.java

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 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 settings discussed above are editor dependent.

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 and you will see the window similar to the *Figure 4*.
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

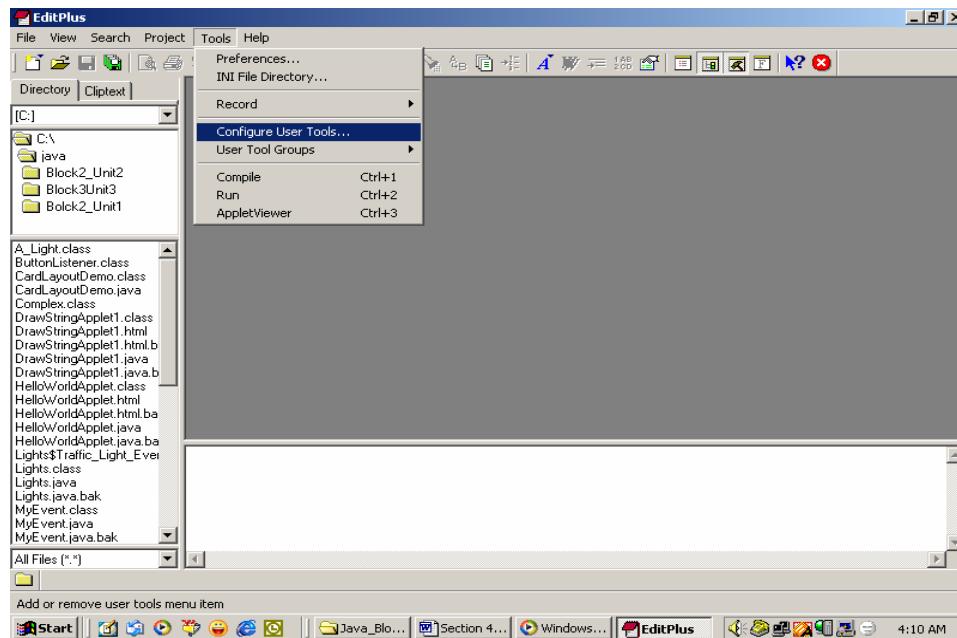


Figure 4: Configure user tools of editplus

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. (As shown in *Figure 5*)
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.

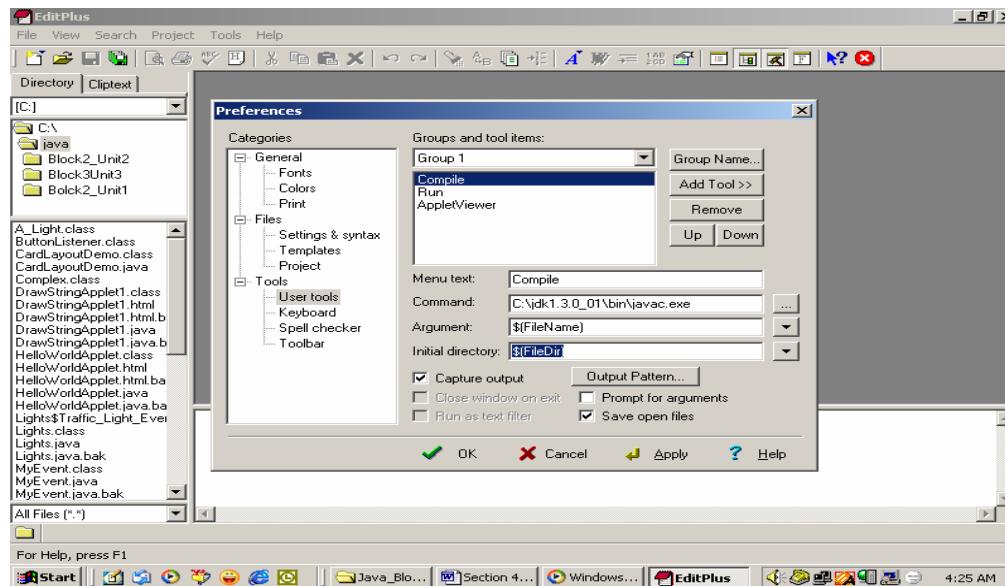


Figure 5: Setting editplus to compile Programs

Setting for Running Java Application Programs

11. In front of Menu text: write Run (as shown in *Figure 6* given below)
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.

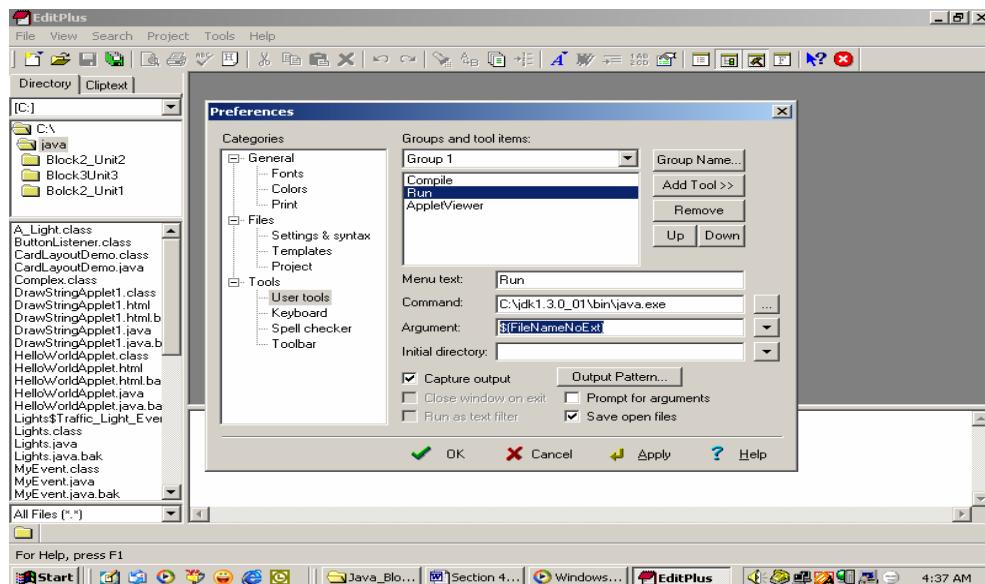


Figure 6: Setting editplus to run Java Application Programs

Setting for running Java Applet Programs

17. In front of Menu text: write AppletViewer (as shown in *Figure 7* given below)
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.

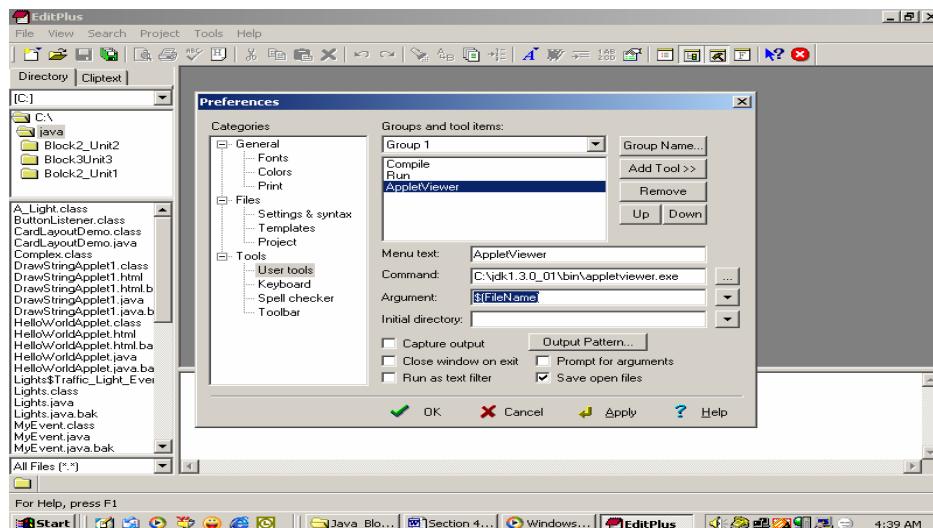


Figure 7: Setting editplus to run Java Applets

Now you will find that three more items Compile, Run, and AppletViewer are added to the Tools menu of EditPlus. For Compile $\text{ctrl}+1$, for Run $\text{ctrl}+2$, and for AppletViewer $\text{ctrl}+3$ can be used.

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

4.4 EXAMPLE PROGRAMS

In this we will explain how to compile and run application program as well as applets.

1. Write your program Welcome_Application.java in EditPlus as demonstrated in *Figure 8*.
2. Compile using ctrl+1 button
3. Run using ctrl+2 button

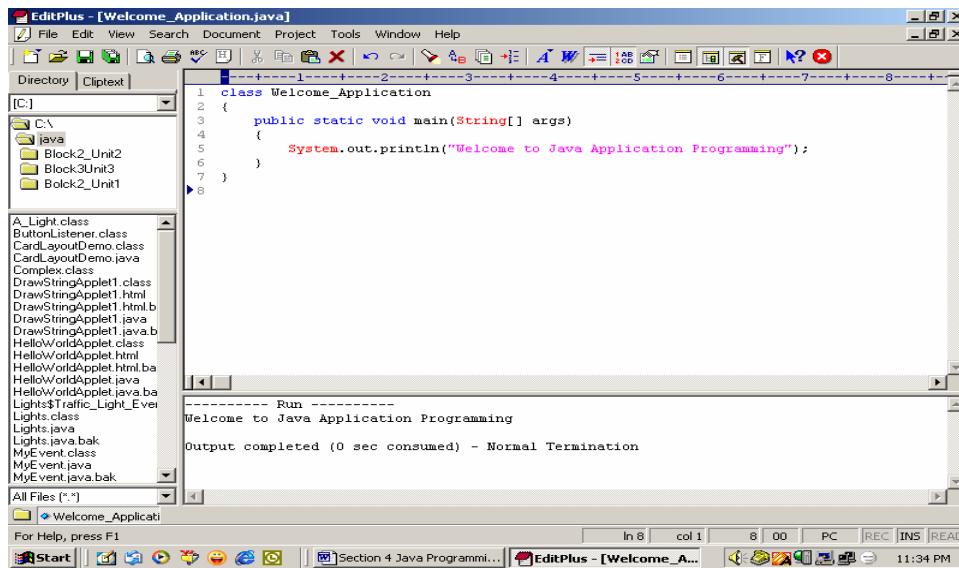


Figure 8: Compile and run application program in EditPlus

In case of Applet first you write applet as shown in *Figure 9*.

1. Compile using ctrl+1 button similar to the previous way.
2. Run it using ctrl+3 button as shown in *Figure 10* but take care you are applying ctrl+3 on appropriate HTML file or not.

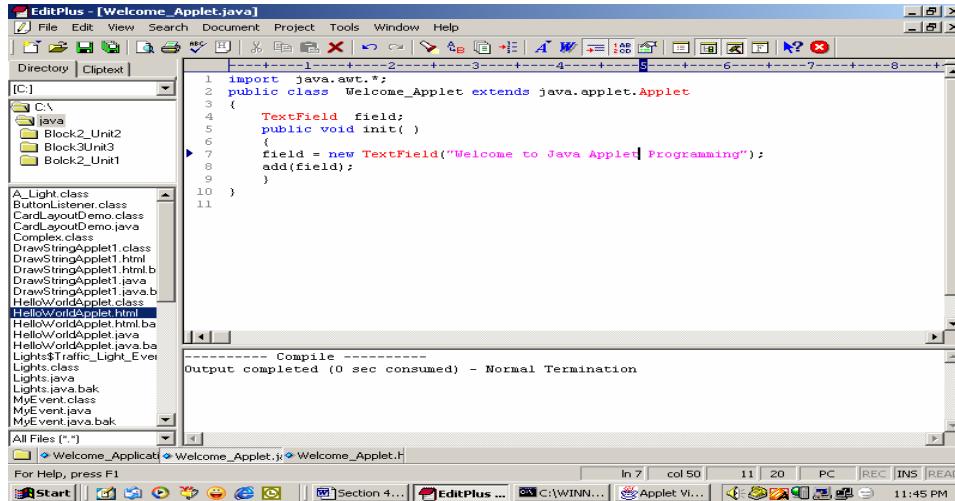


Figure 9: Compile applet in EditPlus

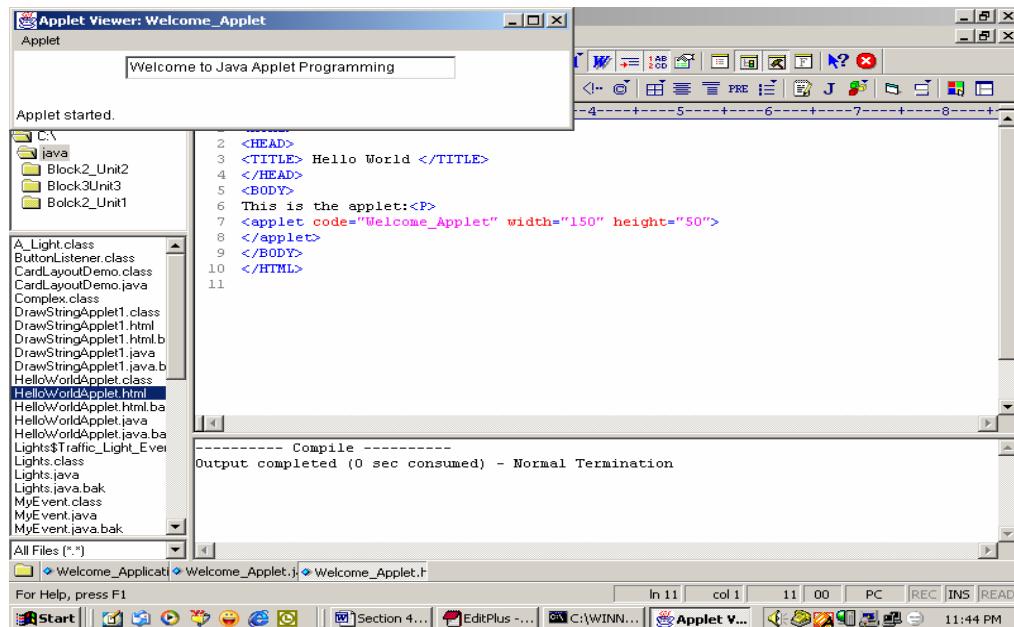


Figure 10: Run applet in applet viewer using editplus

4.5 LIST OF LAB ASSIGNMENTS

Session 1:

Data types, variables and operators

Exercise 1: Write a program in Java to implement the formula ($\text{Area} = \text{Height} \times \text{Width}$) to find the area of a rectangle. Where Height and Width are the rectangle's height and 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.

Exercise 4: Write a program in Java to find the average of marks you obtained in your 10+2 class.

Session 2: Statements and array

Exercise1: Write a program in Java to find $A \times B$ where A is a matrix of 3×3 and B is a matrix of 3×4 . Take the values in matrixes A and B from the user.

Exercise 2: Write a program in Java to compute the sum of the digits of a given integer. Remember, your integer should not be less than the five digits. (e.g., if input is 23451 then sum of the digits of 23451 will be 15)

Session 3: Class and Objects

Exercise 1: Write a program in Java with class Rectangle with the data fields width, length, area and colour. The length, width and area are of double type and colour is of string type .The methods are set_length(), set_width(), set_colour(), and find_area(). Create two object of Rectangle and compare their area and colour. If area and

color both are the same for the objects then display “Matching Rectangles”, otherwise display “Non matching Rectangle”.

Exercise 2: Create a class Account with two overloaded constructors. The first constructor is used for initializing, the name of account holder, the account number and the initial amount in the account. The second constructor is used for initializing the name of the account holder, the account number, the addresses, the type of account and the current balance. The Account class is having methods Deposit (), Withdraw (), and Get_Balance(). Make the necessary assumption for data members and return types of the methods. Create objects of Account class and use them.

Exercise 3: Write a program in Java to create a stack class of variable size with push() and pop () methods. Create two objects of stack with 10 data items in both. Compare the top elements of both stack and print the comparison result.

Session 4: Inheritance and polymorphism

Exercise 1: Write a Java program to show that private member of a super class cannot be accessed from derived classes.

Exercise 2: Write a program in Java to create a Player class. Inherit the classes Cricket _Player, Football _Player and Hockey_ Player from Player class.

Exercise 3: Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days s/he works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

Exercise 4: Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Writ a program using the concept of polymorphism in Java to calculate the charges.

Session 5: Package and Interface

Exercise 1: 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 2: Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program.

Exercise 3: Create an Interface having two methods division and modules. Create a class, which overrides these methods.

Exercise 4: Write a program in Java which implements 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: Exception Handling

Exercise 1: Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsException, so that any such problem doesn't cause illegal termination of program.

Exercise 2: Write a Java program to enable the user to handle any chance of divide by zero exception.

Exercise 3: Create an exception class, which throws an exception if operand is non-numeric in calculating modules. (Use command line arguments).

Exercise 4: On a single track two vehicles are running. As vehicles are going in same direction there is no problem. If the vehicles are running in different direction there is a chance of collision. To avoid collisions write a Java program using exception handling. You are free to make necessary assumptions.

Session 7: Multithreading

Exercise 1: Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting.

Exercise 2: Write a program to launch 10 threads. Each thread increments a counter variable. Run the program with synchronization.

Exercise 3: Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.

Exercise 4: Write a Java program using thread synchronization in multithreading (You can take some objects visible on screen for real time effect).

Session 8: Reading, Writing and String handling in Java

Exercise 1: Write a program in Java to create a String object. Initialize this object with your name. Find the length of your name using the appropriate String method. Find whether the character ‘a’ is in your name or not; if yes find the number of times ‘a’ appears in your name. Print locations of occurrences of ‘a’. Try the same for different String objects.

Exercise 2: Write a program in Java for String handling which performs the following:

- i) Checks the capacity of StringBuffer objects.
- ii) Reverses the contents of a string given on console and converts the resultant string in upper case.
- iii) Reads a string from console and appends it to the resultant string of ii.

Exercise 3: Write a program for searching strings for the first occurrence of a character or substring and for the last occurrence of a character or substring.

Exercise 4: Write a program in Java to read a statement from console, convert it into upper case and again print on console.

Exercise 5: 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 6: Write a Java program to copy a file into another file.

Session 9: Applets and its applications

Exercise 1: Write a Java Applet program which reads 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 your applet look good.

Exercise 2: Create an applet which displays a rectangle/string with specified colour & coordinate passed as parameter from the HTML file.

Exercise 3: Create an applet which will display the calendar of a given date.

Exercise 4: Write a program to store student's detail using Card Layout.

Exercise 5: Write a Java Applet program, which provides a text area with horizontal and vertical scrollbars. Type some lines of text in the text area and use scrollbars for movements in the text area. Read a word in a text field and find whether the word is in the content of the text area or not.

Session 10: Networking and other advanced feature and JAVA

Exercise 1: 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.

Exercise 2: Create an applet which takes name and age as parameters and display the message “<name> is <age> year old.”. Print the URL of the class file.

Exercise 3: Write a program to test Socket functionality for appropriate hostname and port number.

Exercise 4: 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.

4.6 SUMMARY

In the beginning of the section we aimed to provide you a first step assistance to Java programming. In this section, we discussed the basics and importance of working with Java. Though some of these topics you have already studied in your course material of MCS-024, working on something and simply studying have many differences. In the beginning of this section we laid emphasis on the most fundamental concepts and mechanisms provided by Java language. How you will start working on Java (starting from downloading jdk) this section provided you interactive guidance so you can start working on Java. We showed you how you can start compiling and executing your program using freely downloadable software known as editplus or DOS with the help of a suitable example. More stress has been laid on the compiling and executing of the first program and related troubleshooting. This enables better utilization of lab hours and learners will feel motivated to work with software without getting trapped in the problem (at least not in the beginning). Further, in this section the learner had ten sessions including programming problems which s/he should complete in lab. More stress has been laid on programming using multithreading, strings, inheritance, exception handling and applets as it is regarded as a very special skill. It is very important to attend the lab sessions after doing the necessary homework.

You must have completed all your lab sessions successfully. You should take printouts of all your lab exercises with the output of program. (It is better to make a file for all your lab exercises.) If you had executed programs successfully in lab without sufficient preparation, as indicated in the beginning, then it is very important to assess the efficiency (space and time complexities) of the program.

4.7 FURTHER READINGS

The following are the books and website references with you can use in your lab course:

Books on Java

1. Java: An Introduction to Computer Science and Programming by Walter Savitch.
2. Problem Solving with Java by Elliot B. Koffman and Ursula Wolz.
3. Introduction to Programming Using Java: An Object-Oriented Approach by David M. Arnow and Gerald Weiss.
4. David M. Arnow and Gerald Weiss, Introduction to Programming Using Java: An Object-Oriented Approach, Addison-Wesley.
5. Ken Arnold, James Gosling, and David Holmes, The Java Programming Language (Third Edition), Addison-Wesley.
6. Judith Bishop, Java Gently: Programming Principles Explained (Third Edition), Addison-Wesley.

Tutorials on web

- <http://java.sun.com/docs/books/tutorial/index.html>
<http://www.ibiblio.org/javafaq/javatutorial.html>
<http://herzberg.ca.sandia.gov/JavaCourse/>
http://www.suncity.com/Tutorial_Java/partOne/start.html
<http://scitec.uwichill.edu.bb/cmp/online/CS24L/java/jdkintro.htm>