



NATIONAL OPEN UNIVERSITY OF NIGERIA

SCHOOL OF SCIENCE AND TECHNOLOGY

COURSE CODE: CIT 213

COURSE TITLE: ELEMENTARY DATA PROCESSING



CIT 213

ELEMENTARY DATA PROCESSING

Course Developer Dr. Ikhu-Omoregbe N. A.

Course Editor

Course Co-ordinator Afolunso, A. A.
National Open University of Nigeria
Lagos.



NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Annex
245 Samuel Adesujo Ademulegun Street
Central Business District
Opposite Arewa Suites
Abuja

e-mail: centralinfo@nou.edu.ng
URL: www.nou.edu.ng

National Open University of Nigeria 2008

First Printed 2008

ISBN

All Rights Reserved

Printed by ..
For
National Open University of Nigeria

TABLE OF CONTENTS	PAGE
Introduction.....	1 - 2
What you will learn in this Course.....	2
Course Aims.....	2
Course Objectives.....	2 3
Working through this Course.....	3
Course Materials.....	3
Study Units	3 - 4
Textbooks and References	4 - 5
Assignment File.....	5
Presentation Schedule.....	5
Ass essment.....	6
Tutor Marked Assignments (TMAs)	6
Examination and Grading.....	6 - 7
Course Marking Scheme.....	7
Cours e Overview	7 - 8
How to Get the Best from This Course	8 - 9
Tutors and Tutorials	10
Summary	10 -
11Figure 1	

Introduction

CIT 213 Elementary Data Processing is a two [2] credit unit course of sixteen units. It deals with the fundamental concepts of data processing, basic components of computer systems, data processing personnel, file access and organization techniques. In addition it deals with the tools for developing data processing application and provides an overview of Visual Basic Language and Structured Query Language. Sample programs and queries are provided to enable the students acquire the necessary skills for solving real life problems. The study material provides adequate background information that is relevant for students pursuit of a career in data processing or related disciplines. The course is divided into four modules:

Module 1 introduces the basic concepts of data processing, explains the meaning of data, information and data processing. A detailed description of the computer hardware and software components required for data processing is covered. The various input and output devices such as pointing devices, pen-based devices, optical character recognition, printers, plotters etc and their suitability for data processing are explained.

Module 2 provides information on data storage devices and the parameters for disk performance. File access and organization techniques - serial, sequential, indexed-sequential and random file organization are covered in this module. In addition, you will learn about data processing techniques such as Batch processing, Online Processing, Transaction Processing, Real-time Systems and Distributed processing.

Module 3 begins with database processing. The meaning of database, components of database and database models i.e. relational database, hierarchical database, network database and object-oriented database are explained. Data processing personnel are covered. In addition, tools for developing data processing applications such as programming languages, report generators, computer aided software engineering, query languages etc are covered in this module.

Module 4 provides an overview of Visual Basic (VB) language and Structure Query Language (SQL). The fundamental concepts of visual and event driven programming are covered. You will learn how to write simple programs that respond to events. Control structure, arrays, file processing and how to use the print statements are covered. Also, you will learn how to write simple SQL queries by using the CREATE, SELECT, INSERT, UPDATE and DELETE commands.

The aim of this course is to equip you with basic information and understanding of data processing concepts as well as acquiring the relevant skills to manage and implement data processing systems. At the end of the course, you would have gained the relevant knowledge and acquired the skills to effectively manage the data processing department of any organization.

This Course Guide gives you a brief overview of the course content, course duration, and course materials.

What you will learn in this course

The main purpose of this course is to provide the required knowledge and business programming skills for you to effectively data processing systems and manage the data processing department of an organization. The basic concepts of data processing, computer systems required for data processing, data processing personnel, tools for developing data processing application, and basic programming skills are covered in this study material. This, we intend to achieve through the following:

Course Aims

- i. Explain the concepts, systems, personnel, tools for data processing
- ii. Describe data processing techniques
- iii Acquire skills for developing, implementing and managing data processing systems or department

Course Objectives

Certain objectives have been set out to ensure that the course achieves its aims. Apart from the course objectives, every unit of this course has set objectives. In the course of the study, you will need to confirm, at the end of each unit, if you have met the objectives set at the beginning of each unit. By the end of this course you should be able to:

- i. Explain the meaning of data, information, and data processing;
- ii. Identify and describe the basic hardware components of a computer system;
- iii Describe some software use for data processing;
- iv Explain file access and organization techniques;
- v Describe batch processing, real-time processing, transaction processing etc;
- vi Explain the meaning of Database Management Systems and Database Model;
- vii Describe various tools for developing data processing applications;
- viii Explain the concept of visual and event driven programming;
- ix Write simple Visual Basic programs;
- x Write SQL statements

Working through This Course

In order to have a thorough understanding of the course units, you will need to read and understand the contents, practice the what you have learnt by studying the network of

your organization or proposing one if there is none in existence, and be committed to learning and implementing your knowledge.

This course is designed to cover approximately sixteen weeks, and it will require your devoted attention. You should do the exercises in the Tutor-Marked Assignments and submit to your tutors.

Course Materials

These include:

1. Course Guide
2. Study Units
3. Recommended Texts
4. A file for your assignments and for records to monitor your progress.

Study Units

There are seventeen study units in this course:

Module1

- Unit 1 INTRODUCTION AND SCOPE OF DATA PROCESSING
- Unit 2 BASIC HARDWARE COMPONENTS AND MEMORY
CAPACITY MEASUREMENT
- Unit 3 BASIC SOFTWARE FOR DATA PROCESSING
- Unit 4 DATA CAPTURING AND REPORTING DEVICES

Module 2

- Unit 1 DATA STORAGE DEVICES
- Unit 2 FILE ORGANIZATION
- Unit 3 DATA PROCESSING TECHNIQUES
- Unit 4 TRADITIONAL FILE SYSTEM PROCESSING

Module 3

- Unit 1 DATA BASE PROCESSING
- Unit 2 DATA PROCESSING PERSONNEL
- Unit 3 TOOLS FOR CREATING DATA PROCESSING APPLICATION
- Unit 4 VISUAL AND EVENT PROGRAMMING

Module 4

Unit 1 CONTROL STRUCTURE AND STRING PROCESSING

Unit 2 ARRAYS AND PRINTING

Unit 3 FILE PROCESSING

Unit 4 STRUCTURED QUERY LANGUAGE (SQL)

Make use of the course materials, do the exercises to enhance your learning.

Textbooks and References

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L Consultancy Services Pvt. Ltd, 1999

Brain K. Williams, and Stacey C. Sawyer, Using Information Technology, A practical Introduction to Computers and Communications, McGraw Hill, 2005, N.Y

Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, Fifth Edition, McGraw-Hill Higher Education, N.Y, 2002

David C. Kreines, Oracle SQL, O Reilly, USA, 2000

Gerald V. Post, Database Management Systems, Designing and Building Business Applications, Third Edition, McGraw-Hill, 2005

George Beekman, Eugene J. Rathswohl, Computer Confluence, Exploring Tomorrow's Technology, IT Edition, Prentice Hall, New Jersey, 2003

Jan L. Harrington, SQL Clearly Explained, Morgan Kaufmann Publisher, An Imprint of Elsevier Science, 2003, USA

Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, Systems Analysis and Design Methods, McGraw Hill, New York, 2004,

Julia Case Bradley, Anita C. Millspaugh, Programming in Visual Basic 6.0, McGraw-Hill Higher Education, 2002

Management Information Systems, Organization and Technology in the Kenneth C. Laudon, Jane P. Laudon, Prentice Hall International, Inc, 2000

Problem Solving Using C: Structured Programming Techniques, Second Edition, Yuksel Uckan, McGraw-Hill, 1999

Yuksel Uckan, Problem Solving Using C, Structured Programming Techniques

McGraw-Hill International Editions, 1999

R. G. Anderson, Data Processing, Volume 1: Principles and Practice, Pitman Publishing, Singapore, 1990

http://www.webopedia.com/DidYouKnow/Hardware_Software/2002/FiveGenerations.asp

<http://computer.howstuffworks.com/hard-disk7.htm>

Assignments File

These are of two types: the self-assessment exercises and the Tutor-Marked Assignments. The self-assessment exercises will enable you monitor your performance by yourself, while the Tutor-Marked Assignment is a supervised assignment. The assignments take a certain percentage of your total score in this course. The Tutor-Marked Assignments will be assessed by your tutor within a specified period. The examination at the end of this course will aim at determining the level of mastery of the subject matter. This course includes seventeen Tutor-Marked Assignments and each must be done and submitted accordingly. Your best scores however, will be recorded for you. Be sure to send these assignments to your tutor before the deadline to avoid loss of marks.

Presentation Schedule

The Presentation Schedule included in your course materials gives you the important dates for the completion of tutor marked assignments and attending tutorials. Remember, you are required to submit all your assignments by the due date. You should guard against lagging behind in your work.

Assessment

There are two aspects to the assessment of the course. First are the tutor marked assignments; second, is a written examination.

In tackling the assignments, you are expected to apply information and knowledge acquired during this course. The assignments must be submitted to your tutor for formal assessment in accordance with the deadlines stated in the Assignment File. The work you submit to your tutor for assessment will count for 30% of your total course mark.

At the end of the course, you will need to sit for a final three-hour examination. This will also count for 70% of your total course mark.

Tutor Marked Assignments (TMAS)

There are seventeen tutor marked assignments in this course. You need to submit all the assignments. The total marks for the best four (4) assignments will be 30% of your total course mark.

Assignment questions for the units in this course are contained in the Assignment File. You should be able to complete your assignments from the information and materials contained in your set textbooks, reading and study units. However, you may wish to use other references to broaden your viewpoint and provide a deeper understanding of the subject.

When you have completed each assignment, send it together with form to your tutor. Make sure that each assignment reaches your tutor on or before the deadline given. If, however, you cannot complete your work on time, contact your tutor before the assignment is done to discuss the possibility of an extension.

Examination and Grading

The final examination for the course will carry 70% of the total marks available for this course. The examination will cover every aspect of the course, so you are advised to revise all your corrected assignments before the examination.

This course endows you with the status of a teacher and that of a learner. This means that you teach yourself and that you learn, as your learning capabilities would allow. It also means that you are in a better position to determine and to ascertain the what, the how, and the when of your course learning. No teacher imposes any method of learning on you.

The course units are similarly designed with the introduction following the table of contents, then a set of objectives and then the concepts and so on.

The objectives guide you as you go through the units to ascertain your knowledge of the required terms and expressions.

Course Marking Scheme

This table shows how the actual course marking is broken down.

Assessment	Marks
Assignment 1- 4	Four assignments, best three marks of the four count at 30% of course marks
Final Examination	70% of overall course marks

Total	100% of course marks
-------	----------------------

Table 1: Course Marking Scheme

Course Overview

Unit	Title of Work	Weeks	Assessment
		Activity	(End of Unit)
	Course Guide	Week 1	
	Module 1		
1	Introduction and Scope of Data Processing	Week 1	Assignment 1
2	Basic Hardware Components and Memory	Week 2	Assignment 2
3	Capacity Measurement	Week 3	Assignment 3
4	Basic Software for Data Processing	Week 4	Assignment 4
	Module 2		
1	Data Storage Devices	Week 5	Assignment 5
2	File Organization	Week 6	Assignment 6
3	Data Processing Techniques	Week 7	Assignment 7
4	Traditional File System Processing	Week 8	Assignment 8
	Module 3		
1	Data Base Processing	Week 9	Assignment 9
2	Data Processing Personnel	Week 10	Assignment 10
3	Tools for Creating Data Processing Application	Week 11	Assignment 11
4	Visual and Event Programming	Week 12	Assignment 12
	Module 4		
1	Control Structure and String Processing	Week 13	Assignment 13
2	Arrays and Printing	Week 14	Assignment 14
3	File Processing	Week 15	Assignment 15

4	Structured Query Language (SQL)	Week 16	Assignment 16
	Revision	Week 16	
	Examination	Week 17	
	TOTAL	17 weeks	

How to get the best from this course

In distance learning the study units replace the university lecturer. This is one of the great advantages of distance learning; you can read and work through specially designed study materials at your own pace, and at a time and place that suit you best. Think of it as reading the lecture instead of listening to a lecturer. In the same way that a lecturer might set you some reading to do, the study units tell you when to read your set books or other material. Just as a lecturer might give you an in-class exercise, your study units provide exercises for you to do at appropriate points.

Each of the study units follows a common format. The first item is an introduction to the subject matter of the unit and how a particular unit is integrated with the other units and the course as a whole. Next is a set of learning objectives. These objectives enable you know what you should be able to do by the time you have completed the unit. You should use these objectives to guide your study. When you have finished the units you must go back and check whether you have achieved the objectives. If you make a habit of doing this you will significantly improve your chances of passing the course.

Remember that your tutor's job is to assist you. When you need help, don't hesitate to call and ask your tutor to provide it.

1. Read this Course Guide thoroughly.
2. Organize a study schedule. Refer to the Course Overview for more details. Note the time you are expected to spend on each unit and how the assignments relate to the units. Whatever method you chose to use, you should decide on it and write in your own dates for working on each unit.
3. Once you have created your own study schedule, do everything you can to stick to it. The major reason that students fail is that they lag behind in their course work.
4. Turn to Unit 1 and read the introduction and the objectives for the unit.
5. Assemble the study materials. Information about what you need for a unit is given in the Overview at the beginning of each unit. You will almost always need

both the study unit you are working on and one of your set of books on your desk at the same time.

6. Work through the unit. The content of the unit itself has been arranged to provide a sequence for you to follow. As you work through the unit you will be instructed to read sections from your set books or other articles. Use the unit to guide your reading.
7. Review the objectives for each study unit to confirm that you have achieved them. If you feel unsure about any of the objectives, review the study material or consult your tutor.
8. When you are confident that you have achieved a unit's objectives, you can then start on the next unit. Proceed unit by unit through the course and try to pace your study so that you keep yourself on schedule.
9. When you have submitted an assignment to your tutor for marking, do not wait for its return before starting on the next unit. Keep to your schedule. When the assignment is returned, pay particular attention to your tutor's comments, both on the tutor-marked assignment form and also written on the assignment. Consult your tutor as soon as possible if you have any questions or problems.
10. After completing the last unit, review the course and prepare yourself for the final examination. Check that you have achieved the unit objectives (listed at the beginning of each unit) and the course objectives (listed in this Course Guide).

Tutors and Tutorials

There are 12 hours of tutorials provided in support of this course. You will be notified of the dates, times and location of these tutorials, together with the name and phone number of your tutor, as soon as you are allocated a tutorial group.

Your tutor will mark and comment on your assignments, keep a close watch on your progress and on any difficulties you might encounter and provide assistance to you during the course. You must mail or submit your tutor-marked assignments to your tutor well before the due date (at least two working days are required). They will be marked by your tutor and returned to you as soon as possible.

Do not hesitate to contact your tutor by tele phone, or e-mail if you need help. The following might be circumstances in which you would find help necessary. Contact your tutor if:

- you do not understand any part of the study units or the assigned readings,
- you have difficulty with the self-tests or exercises,
- you have a question or problem with an assignment, with your tutor's comments on an assignment or with the grading of an assignment.

You should try your best to attend the tutorials. This is the only chance to have face to face contact with your tutor and to ask questions which are answered instantly. You can raise any problem encountered in the course of your study. To gain the maximum benefit from course tutorials, prepare a question list before attending them. You will learn a lot from participating in discussions actively.

Summary

Elementary Data processing provides knowledge and skill for data processing. The skills you need to manage and implement data processing system are expected to be acquired in this course. The content of the course material was planned and written to ensure that you acquire the proper knowledge and skills to solve real life data processing problems. Fundamental programming principles are taught. The sample programs provided are for you to develop your programming skills.

Course Code CIT 213

Course Title Elementary Data Processing

Course Developer Dr. Ikhu-Omoregbe N. A.

Course Editor

Course Co-ordinator Afolunso, A. A.
National Open University of Nigeria
Lagos.



NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Annex
245 Samuel Adesujo Ademulegun Street
Central Business District
Opposite Arewa Suites
Abuja

e-mail: centralinfo@nou.edu.ng
URL: www.nou.edu.ng

National Open University of Nigeria 2008

First Printed 2008

ISBN

All Rights Reserved

Printed by ..
For
National Open University of Nigeria

MO DULE 1 UNIT 1: INTRO DUCTIO N AND SCOP E OF DATA PROCESSING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Text
 - 3.1 Data and Information
 - 3.2 Electronic Data Processing Systems
 - 3.3 The Scope of Data Processing System
 - 3.4 Computer Systems
 - 3.4.1 Computer Generation
 - 3.4.2 Analog and Digital Computer
 - 3.4.2.1 Analog Computers
 - 3.4.2.2 Digital Computers
 - 3.5 The Basic Operations of Computing
 - 3.6 Benefits of Using Computer in Business
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introductio n

The word data refers to raw facts about and object. Data is meaningless until it subjected to manual or electronic processing by a computer to produce information used for decision making. Computers are used to process data because of their speed, reliability and other functionalities they offer. Computers can be distinguished by their sizes, types and generation.

2.0 Objectives

At the end of this unit, you should be able to:

- Define and explain the terms; data and information
- Describe a computer system
- Explain the meaning of data processing
- Distinguish between the different types of computer
- Describe the basic operations of a computer
- List some benefits of using computer in business

3.1 Data and Information

Data exist in a variety of forms such as numbers or text on pieces of paper, as bytes in computer memory or as facts stored in a person's mind. Data is the plural form of the word datum and refers to basic facts about an object. An object refers to a person, place,

or a thing. Data consist of numbers, words, images, etc. Data is meaningless until it undergoes some forms of processing. The outcome of a processed data is information. Although the word data and information are often used interchangeably, there is an important distinction between them. In the strictest sense, data consist of the raw number that computers process to produce information. Data can be looked at as facts out of context. Table 1 captures basic facts about a matriculated student of National Open University of Nigeria. The data items consisting of Matriculation Number, Name, School, Department, Centre, Banker, Status and e-Email are used to hold the students details.

S/no	Data item	Actual Details or value
1	Matriculation Number	NOU050010999
2	Name	Adebayo Musa, Ada
3	School	Science and Technology
4	Department	Computer Science
5	Centre	Lagos
6	Banker	Zenith Bank
7	Status	Returning
8	e-Email	ama@nou.edu.ng

Table 1: Student Registration

Similarly, a commercial bank may request for the following data before the same student can operate an account or enjoy some banking services

S/no	Data item	Actual Detail or value
1	Matriculation Number	NOU050010999
2	Name	Adebayo, Musa, Ada
3	Date of Birth	14th April, 1970
4	Nationality	Nigerian
5	Sex	Male
6	Occupation	Student
7	School	National Open University
8	Matriculation Number	NOU050010999

Table 2: Student Account

The details of this student (Adebayo Musa, Ada) if treated independently may not have specific meanings until it is subjected to some manual or computerized processing. The raw data in the tables may be further processed by a human or input into a computer, stored, processed and transmitted (output) to another human or a computer. A typical data processing function will be to locate a student record or data from a large file based on matriculation number. If this is to be carried out manually a lot of time and efforts will be

required by the data processing personnel. When the method of data processing is manual, it is called manual data processing. On the other hand, when the method of processing is electronic we refer to it as electronic data processing. Computers are used to turn meaningless data by processing them into useful information, such as spreadsheet, graphs, and reports. In short, information is defined as either a meaningful answer to a query or a meaningful stimulus that can cascade into further queries. For example, with the data in table1 and table 2, the bank official may be interested in the number of students from each school that have subscribed to a particular service its offers. In which case, a query will be run on the existing database. The result of the query will provide information which will help the management of the bank to discontinue or continue with the service. Some characteristics of information are: relevance, accuracy, conciseness, timeliness, and completeness.

3.2 Electronic Data Processing Systems

Electronic Data processing (DP) is any computer automated process that converts raw facts i.e data into output i.e useful information. Data becomes more useful only when they have undergone some computer processing and well-presented to allow for decision making. Data processing systems have become indispensable in managing information in modern day organizations. In recent times, data processing systems are quite often referred to as information systems to highlight their practicality. All the same, both terms are more or less the same, performing similar functions; data processing systems typically manipulate raw figures into information, and likewise information systems usually take raw figures as input to generate useful information as output. In many organizations, these systems exist as payroll systems, sales and marketing systems, finance and accounting systems, manufacturing and production system, human resources systems, University management systems etc.

Also, the term data processing can apply to any process of transforming data into information and also the converting of information back to data.

3.3 The Scope of Data Processing System

Conventional usage of the terms data processing and information systems restricts their use to refer to the algorithmic derivations, logical deductions, and statistical calculations that recur perennially in general business environments, rather than in the more expansive sense of all conversions of real-world measurements into real-world information. In the real sense, although, data processing requires the use of computers, the level of computerizations is usually not too computing intensive as may be expected in scientific and engineering applications. Computing is a term restricted to number crunching that is arithmetical calculations. These include adding, multiplying, subtracting and dividing, and exponentiation (raising numbers to specific powers) etc. The basic data processing operations performed on business data include the following:

- Data Collection
- Data Capture

- Data Recording
- Data Entry
- Data Transmission
- Data Sorting
- Updating of Information
- Adding of Data
- Deletion of information
- Data Validation
- Data Verification
- Summarizing of data
- Printing of Results
- Summaries of data for management decision
- Calculating data
- Sorting data
- Classifying data
- Summarizing data
- Comparing data
- Statistical reports
- Data Mining

Thus, data processing consist of those activities concerned with the systematic recording, sorting, computing, modifying, reporting displaying and printing of details relating to business transactions. While a data processing system can therefore be viewed as an administrative system superimposed upon the physical business systems such as banking, engineering, insurance etc. The ultimate goal of data processing is to obtain information with which to control financial and administrative aspects of the business. In addition, managers and administrators with up-to-date information are able to make good decision that will positively affect the growth of organization.

3.4 Computer Systems

A computer is a device which given a set of instructions or data can be used to perform given task or tasks. Also, a computer can be referred to as a programmable, multi-user device that accepts data, raw facts and figures, and processes, or manipulates, it into useful information. Computer are used primarily to speed up the problems solving and increase the overall productivity of its users. The computer reads in data and instructions, does some processing, and stores or outputs desired results. The computer is designed with input mechanism for reading data into the computer, internal storage facilities, and mechanisms for communicating with the outside world (output for writing data out.) In order for data to be processed by a computer, the data needs first to be converted into a machine readable format. Once data is in digital format, various procederes can be applied on the data to get useful information.

3.4.1 Computer Generation

The history of computer development is often referred to in reference to the different generations of computing devices. Computer can be distinguished by these generations since each generation is characterized by a major technological innovation that fundamentally affected the way computers operate, resulting in smaller sized, cheaper, more powerful and more reliable systems

First Generation 1940-1956

The first computers used vacuum tubes for circuitry and magnetic drums for memory, and were often enormous, taking up a large space. They were very expensive to operate and maintain. In addition to using a great deal of electricity, they generated a lot of heat and were unreliable. First generation computers allowed programming in machine language to perform operations, and they could only solve one problem at a time. Input was based on punched cards and paper tape, and output was displayed on printouts. Common examples are The UNIVAC and ENIAC.

Second Generation 1956-1963

In this generation of computers, transistors were used in place of vacuum tubes. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable than their first-generation computers. This generation of computer relied on punched cards for input and printouts for output. This generation of computers supported the use of assembly languages - this allowed programmers to specify instructions in words.

Third Generation - 1964 -1971

Third generation computers used integrated circuits. Transistors were miniaturized and placed on silicon chips, called semiconductors, which drastically increased the speed and efficiency of computers. This invention led to the widespread use of computers today. Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system, which allowed the device to run many different applications at one time with a central program that monitored the memory. High-level programming languages such as early versions of COBOL and FORTRAN were used on these systems.

Fourth Generation - 1971-Present

The microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip. Fourth generation computers use microprocessor chips, which are large-scale integrated circuits containing thousands of transistors. The microprocessor led to the creation of the first personal computer. What in the first generation filled an entire room could now fit in the palm of the hand. As these small computers became more powerful, they could be connected to form networks, which eventually led to the development of the Internet. Fourth generation computers

also saw the development of Graphical Users Interface, the pointer devices such as mouse and personal digital assistants..

Fifth Generation - Present and Beyond: Artificial Intelligence

Fifth generation computing devices, based on artificial intelligence, are still in development. However, applications such as voice recognition are gradually coming into the market. The use of parallel processing and superconductors is helping to make artificial intelligence a reality. One of the goals of the fifth-generation computing is to develop devices that respond to natural language input.

Computers are available in different categories of size, type and capabilities.

3.4.2 Analog and Digital Computer

Early attempts at manufacturing computers used analog techniques, but accuracy, speed and reliability were not very acceptable. Data processed by the computer can be described in two forms: analog or digital. These two forms give an insight to a way of classifying computers today. Computing devices are classified as Analog or Digital according to the means by which they represent data.

3.4.2.1 Analog Computers

Analog refers to non-digital (non-computer-based), continuous variable forms of data transmission, including voice and video. An analog computer represents a datum as a voltage, distance, position, or other physical quantity. Analog - continuously varying in strength and/or quantity. For example, sound, light, temperature, and pressure values can fall anywhere along a continuum or range. Examples of analog devices are a thermometer, a speedometer, a clock (with hour, minute, and second hands that sweep around the dial), and a pressure sensor, which can measure continuous fluctuation.

3.4.2.2 Digital Computers

The term digital describes any system based on discontinuous data or events. For example, electricity is either turned on or turned off. With a two-state on/off arrangement, one state can represent a 1 digit, and the other a 0 digit. Because computers are based on on/off or other two-state conditions, they use the binary system, which is able to represent any number using only two digits 0 and 1. Binary means having two components, alternatives, or outcomes. With this kind of arrangement digital data then, consist of data represented by on/off signal symbolized as 0s and 1s. A digital computer represents a datum as a sequence of symbols drawn from a fixed alphabet. The most common digital computers use a binary alphabet, that is, an alphabet of two characters, typically denoted "0" and "1". Digital computers are more common than analog computers. Digital computer exist in different categories of sizes such as mainframes, minicomputers, microcomputers, workstations and supercomputers

Mainframe

A mainframe is the largest and relatively most powerful computer, a powerhouse with large storage and very rapid processing power and speed. It is used for very large amount of business, scientific and engineering, military data. They are found in many data processing centers such as in Banks, military, airports, nuclear stations, universities, and other organizations where large amount of data are processed.

Minicomputer

A minicomputer is a mid-range, multi-purpose computer, about the size of an office desk produced by a number of manufacturers including Dell, Hewlett Packard, and IBM. Often used in universities, factories, and research laboratories. Applications processed on minicomputers include, Accounting routines, banking, examination processing, hospital management systems, databases, geographical information systems and other management information systems. Some training or professional certifications may be required to handle these systems compared to the operations of microcomputers.

Microcomputers

Microcomputers are sometimes referred to as personal computers (PCs). They are the computers that can be placed on desktops or carried from room to room. Some manufacturers of microcomputers include: IBM, Dell, Hewlett Packard, Apple, and Compaq. As the names apply PCs are used and operated by end users for their own particular processing needs such as payroll, inventory control, asset management, result verification, patient medical records, and other general accounting routines. Laptops and palmtops in particular are the portable types of PCs built with a lot of mobility functionality. Not so much skill is required to operate micro computers as compared with mini and mainframe computers.

Workstation

A workstation is so much like a PC except that it is more powerful and has more capabilities for handling mathematical and graphics-processing than a PC. It can handle more complicated tasks than PC in the same amount of time. Workstations are used mainly in handling computing (number crunching and algorithmic) applications such as in science, engineering, and design work that requires powerful graphics or computational capabilities rather than data processing applications where arithmetic operation may be ideal.

Supercomputer

A supercomputer is highly sophisticated and powerful machine that is used for tasks requiring rapid and complex calculations with hundreds of thousands of variable factors. Supercomputers traditionally have been used in scientific and military work, but they are starting to be used in business as well. A supercomputer is especially sophisticated and powerful type of computer that is used primarily for extremely rapid and complex computations with hundreds of thousands of variable factors. Supercomputers traditionally have been used for classified weapons research, weather forecasting, and

petroleum and engineering applications, all of which use complex mathematical models and simulation. Although extremely expensive, supercomputers are beginning to be employed in business for data mining and the manipulation of vast quantities of data. Supercomputers can perform complex and massive computations almost instantaneously because they can perform hundreds of billions of calculations per second-many times faster than the largest mainframes.

With advances in communications technologies any these categories of computers could be linked in a network enabling users to share files, software, peripheral devices, such as high speed printers, large external storages or other network resources. Special Server computers designed to support network, providing large memory and disk-storage capacity, high communications capabilities, and powerful CPUs.

As technology continues advance, we expect that these distinctions will become less pronounced.

3.5 The Basic Operations of Computing

The computer processes data by performing the following five main operations

- 1) Input
- 2) processing
- 3) Output
- 4) Storage
- 5) Communication

Input Operations:

At this stage, data or instructions are captured electronically or entered by means of the available input device. An input device is hardware such as a keyboard that allows data from the external environment to be entered into the computer for processing. The data after it has been captured is transform into a form in which the computer can process it.

Processing Operations:

At this stage, the already captured and transformed data is manipulated to generate desired result for the end user other processing systems. The data is worked on by the instructions in the form of programs or queries provided by the users. The instruction and the data determine what output is received from to the computer. The processing instructions may be to add, subtract, multiply, find total, summarize, group, select some data based on some condition etc.

Output Operations:

At the stage the result or information obtained from the data is produced in a form acceptable by the user. An example of on output will be a list of all registered students in the department of accounting who are in their second year at National Open University, Lagos Centre. The output can be printed as printed text, played as sound, displayed as

charts or graphs on the computer screen. Output is usually governed by the need to communicate specific information to a specific audience. The only limit to the different forms of output you can produce is the different types of output devices currently available.

Secondary-storage operations:

At this stage of operations, data, information, and the instructions used for the processing of data are stored temporarily or permanently in primary or secondary storage devices. Data and instruction are stored in primary storage devices during processing to allow easy access and fast processing. Secondary storage devices are used for to store data or instructions more permanently. An example of a primary or internal storage device is RAM (Read Only Memory). While an example of a secondary or external storage device is a flash disk.

Communication Operation

There may be a need to transmit already processed data to an end user or a output device in a remote location. Similarly, data may be gathered from a remote location. The communication operation of the computers performs these by using the communication hardware. This facilitates the connection between computer and between groups of connected computers called network. Computers linked together can share hardware, programs and data. Though, computers can operate as standard alone machines i.e. they may not be connected to anything else. By connecting them in a network more benefits are realized.

The five basic operations of a computer system take place so fast that they seem to happen all at the same time.

3.6 Benefits of Using Computer in Business

- It leads to improved customer relations due to its reliability, generation of more timely reports, and speedier responses to enquiries regarding business operations
- Increases the productivity of staff in general by helping to handle boring and routine operations leaving the staff to be engaged in decision making
- Improved cash flows due to improved sales accounting systems particularly those relating to credit control, invoicing and statement preparation
- It guarantees an improved access to information by means of online and real-time access to information system
- Allows for greater degree of systems integration on the basis that the output of one part of the system (sub system) provides the input to a related sub system, which has the effect of eliminating duplication and delay

- Makes the presentation of information for decision making easy to comprehend. Information is presented in simplified formats such as graphs, charts, graphics images etc
- Helps in the simplification of problem solving by the use of problem solving software
- Makes the supply of information for improving managerial decision readily available

4.0 Conclusion

These terms data and information are commonly used by many people as if they mean the same thing. The differences between them have been made clearer. Data processing is not exactly the same as computing. Some basic operations that are involved in data processing include: Data Collection, Data Capture, Data Recording, Data Entry, Data Transmission, Data Sorting. Electronic data processing requires the use of computers which may be distinguished by their sizes, types or generation

5.0 Summary

In this unit, we have explained the basic Electronic Data Processing Concepts. This should provide a good background for the easy comprehension of the contents in other units.

6.0 Tutor Marked Assignment

- Q1 Explain the term Data Processing
- Q2. What is an Electronic Data Processing System?
- Q3. Describe the different sizes of computer systems
- Q4. What are the main operations of computer systems?

7.0 References/Further Readings

R G Anderson, Data Processing, Volume 1: Principles and Practice, Pitman Publishing, Singapore, 1990

http://www.webopedia.com/DidYouKnow/Hardware_Software/2002/FiveGenerations.asp

MODULE 1 UNIT 2: BASIC HARDWARE COMPONENTS AND MEMORY CAPACITY MEASUREMENT

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Input Device
- 3.2 The Central Processing Unit (CPU)
- 3.3 The Control Unit
- 3.4 Arithmetic and Logic Unit
- 3.5 Memory
- 3.6 Measuring the Memory Capacity of Computer System
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

To function properly, the computer needs both hardware and software. Thus a computer system is made up of hardware and software. The hardware consists of the electronic and electromechanical parts of the computer. This is the tangible or touchable part of the computer. These components include: input devices, central processing unit, main memory, secondary memory, and output devices. Two components handle processing in a computer: the central processing unit, or CPU, and the memory. Both are located on the computer system board, or motherboard, the circuit that connects the CPU to all other hardware devices. The processor works hand in hand with other circuits known as main memory and registers to carry out processing. The basic unit of information representation in the computer is the bit.

2.0 Objectives

At the end of this unit, you should be able to:

- Identify and describe the basic hardware components of a computer system
- Explain the units of measuring computer memory

3.1 Input Device

An input device is used to communicate data and instructions or programs to the computer. The input device accepts the data and converts them into machine understandable forms that can be processed by the computer.

3.2 The Central Processing Unit (CPU)

The CPU is the brain of the computer and it is the place where data is manipulated within the computer system. In a microcomputer, the entire CPU is contained on a tiny chip called a microprocessor which is usually mounted on a piece of plastic with metal wires attached to it. The processor consists of two functional units: the control unit and the arithmetic-logic unit.

3.3 The Control Unit

All the computer's resources are managed from control unit. The control unit of the processor coordinates all activities of the computer system. It directs the movement of electronic signals between arithmetic-logic units and main memory and also directs the electronic signals between main memory and the input and output devices.

3.4 Arithmetic and Logic Unit

The computer can perform only two types of operations: arithmetic operations and logical operations. The arithmetic-logic unit is the electronic circuitry capable of performing these two basic logical operations. Arithmetic operations include addition, subtraction, multiplication, division and exponentiation. All data input into the computer system must first be converted into numbers (0 and 1) before they are processed. In addition to arithmetic operations the computer is able to perform logical operations on two or more numbers. Logical operations are comparisons operations. The ALU compares two pieces of data to see whether one is less than (<), equal to (=), greater than (>), the other. Other comparison operations are greater than or equal to (>=), less than or equal to (<=) or not equal to (≠)

3.5 Memory

The CPU main job is to follow instructions encoded in programs but it does not have the capability to store entire programs or large sets of data permanently. However, the CPU has registers which are devices capable of holding a few bytes of data or instructions at a time. CPU needs to have millions of bytes of space where it can hold programs and the data being manipulated while they are being used. This area is called memory. Computers use two types memory- primary and secondary memory. A computer's primary memory is an electronic device that store information necessary for a program to run. This consists of the volatile memory (RAM) and the non volatile memory (ROM).

RAM

RAM is an acronym for Random Access Memory. It is used for short-term storage of data or program instructions. RAM is volatile: Its contents will be lost when the computer's electric supply is disrupted by a power outage or when the computer is turned off. The purpose of RAM is to hold programs and data while they are in use. Physically, RAM consists of some chips on a small circuit board. A computer does not have to

search its entire memory each time it needs to find data. Access to the data is usually direct using its memory address. The main functions of the RAM chip include:

- It holds data for processing
- It holds instructions (the programs) for processing
- It holds data that has been processed (useful information) and is waiting to be sent to an output device.

ROM

This is an acronym for Read Only Memory. This is a nonvolatile type of memory. Nonvolatile chips always hold the same data; the data in them can be read and used—it cannot be changed. One important reason a computer needs ROM is so that it knows what to do when the power is first turned on. Among other things, ROM contains a set of start-up instructions, which ensure that the rest of memory is functioning properly, check for hardware devices, and check for an operating system on the computer disks drives. Unlike the RAM, which is constantly being written on and erased, ROM cannot be written on or erased by the computer user. In addition ROM chips remember, permanently, information supplied by the manufacturer such as the information about the manufacturer.

Three variations of ROM chips are used in special situations—PROM, EPROM and EEPROM

PROM :

This is an acronym for programmable read-only memory. These are blank chips on which the buyer, using special equipment, writes the programs. Once the program is written, it cannot be erased. Some microprocessor software packages come on PROM units. PROM chips are used by manufacturers as control devices in their products.

EPROM

EPROM stands for electronically erasable read-only memory. They are like PROM chips except that the contents can be erased, using special equipment and new materials can be written. EPROM chips are used for intelligent device control, such as in robots, where the program may have to be modified on a regular basis. Programs in EPROM chip can be erased and reprogrammed.

EEPROM

This is an acronym for electronically erasable programmable read-only-memory. The memory chips can be reprogrammed and can be erased by exposing it to an electrical charge. The advantage of EEPROM chips is that they need not be removed from the computer to be changed.

Registers

These are special, high-speed storage area within the CPU. All data must be represented in a register before it can be processed. For example, the control unit might load two numbers from memory into the registers in the ALU. Then it might tell the ALU to multiply the two numbers (arithmetic operations), or to see whether the numbers are equal (a logical operation). The number of registers that a CPU has and the size of each help determine the power and speed of a CPU. For example a 64-bit CPU is one in which each register is 64 bits wide. Therefore, each CPU instruction can manipulate 64 bits of data.

3.6 Measuring the Memory Capacity of Computer System

Bits

The term bit is a short form of binary digit. A bit is the smallest possible unit of data. To represent anything meaningful the computer needs groups of bits.

Bytes

A group of eight bits is called a byte. It is the next larger unit of data representation within a computer system. With one byte, the computer can represent up to 256 different values because it is possible to count from 0 to 255 with 8 binary digits. The byte is an important unit because it can be adequately used to represent any character on the keyboard, including all the letters (uppercase and lowercase), number, punctuation marks, and other symbols.

Kilobyte

A kilobyte, abbreviated K or KB represents approximately 1000 bytes (or characters). The actual value of 1 Kilobyte is 1024 (2¹⁰) bytes

Megabyte:

A megabyte, abbreviated M or MB and sometimes called meg is used to refer to about 1 million bytes of data.

Gigabytes:

A gigabyte, G or GB, often pronounced gig-a-bite, is used to refer to about 1 billion bytes of data.

Terabyte:

A terabyte, T or TB, is used to refer to about 1 trillion bytes, or 1000 gigabytes of data.

Petabyte:

This is a new measurement which accommodates the huge storage capacities of modern database- It is used to refer to about 1 million gigabyte data

4.0 Conclusion

Computers require the hardware components and software to process data. Computer hardware consists of input devices, central processing units, memory and output devices. A bit is the smallest possible unit of data. Larger amount of data are measured in bytes, kilobyte, megabyte etc.

5.0 Summary

This unit has provided you with adequate information on the different hardware components. The units of measuring the memory capacity of computers are also covered.

6.0 Tutor Marked Assignment

- Q1. What are the main functions of the RAM chip?
- Q2. Explain the meaning of the term RAM is volatile
- Q3. Describe three variations of ROM
- Q4. Explain the functions of a register

7.0 References/Further Readings

Brain K. Williams, and Stacey C. Sawyer, Using Information Technology, A practical Introduction to Computers and Communications, McGraw Hill, 2005, N.Y

Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, Fifth Edition, McGraw-Hill Higher Education, N.Y, 2002

MODULE 1 UNIT 3: BASIC SOFTWARE FOR DATA PROCESSING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Software systems
 - 3.1.1 Systems Software
 - 3.1.2 Application Software
- 3.2 Software for Data Processing
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

Apart from the hardware of a computer discussed in Unit 2, a computer system requires software to make the best use of its capability in solving problems. Computers do not have intelligence of their own; they rely on the instructions and data supplied by a user in order to perform a task. These instructions are called software. Software consists of a group of related programs written in a specific code called programming language. Software can be purchased as Commercial, off-the-shelf (COTS) or developed in house for data processing purposes. Software purchased to perform a general business function is often referred to as a software package.

2.0 Objectives

At the end of this unit, you should be able to:

- Explain the term software
- Describe some basic software for data processing

3.1 Software systems

Software consists of a group of related programs written in a specific code called programming language. Software is created by professional software programmer and comes on disk, CD-ROM, or downloadable online across the Internet. Software can generally be divided into two categories.

- System software
- Application Software

3.1.1 Systems Software

This is software designed to allow the computer to manage its own resources and run the hardware and basic operations. They may be complex to develop and are provided by the manufacturers of computer systems to complement computer hardware. However, others

may be purchased from vendors. This software runs the basic operations; it lets the CPU communicate with the keyboard, the screen, the printer, and the disk drive. However, it does not solve specific problems relating to a business or a profession as applications software. Some examples of system software are operating systems and Language translators

a. Operating System

An operating system is the most important software for a computer. It is made up of many component programs and manages the overall operations of a computer system. Some of the tasks performed by the operating system are:

- It controls various input and output devices and coordinates input and output operations
- It Manages the systems resources such as available memory space in the primary and secondary storage devices
- It allocate memory and processor time to programs
- It validates users and ensure that the resources a user is requesting is made available

Some popular operating systems are Windows, Operating System 2, Macintosh Operating System, Novell Netware, UNIX, and Linux.

b. Language Translator

This is software which translates a computer program written in a high-level programming language, or assembly language to a machine understandable form. If you write a program say in BASIC language, it cannot be executed directly without a translator. A program that is written in high level language is a source program. A source program in its translated form made up of machine codes is called object program. High-level languages are either translated from source code to object code by an Interpreter or a compiler. An interpreter translate one line of a high-level instruction and immediate execute the code, proceeding to the next until the entire program is executed. A compile on the other hand, scans the entire program first and then translate it into machine (object) code.

c. Utilities

These are programs which are often required by many application programs. Examples include programs for formatting or defragmenting a disk. Also we have the sort/merge utilities which are used for sorting and merging large volumes of data.

3.1.2 Application Software

Application software includes programs that are developed using systems software in order to achieve some tasks. There are two types of applications software: application programs that you purchase for solving special classes of problems and application programs that you write to solve your own problems. Some commercially available end-user programs (application software) which are developed by professional programmer teams can be used with little or no additional programming skills. Examples of application software include word processing and desktop publishing software; spreadsheet; database programs; graphics programs; communication software; and special-purpose programs suitable for accounting, scientific and engineering application, education, and entertainment and so on.

3.2 Software for Data Processing

The computers in a data processing environment would run one or more of the following software to facilitate data processing activities.

a. Word processing software

This allows you to use computers to create, edit, store and print documents. You can easily insert, delete, move word, sentences, and paragraphs without ever using the eraser. Of all computer applications, word processors are the most common. Word processing also offers a number of features for dressing up documents with variable margins, type sizes, and style. The user can do all these manipulations on a document or report on the screen before printing it. For example, if you make a typing mistake, you simply move the cursor to the position and correct the mistake. You can move a section of your document to another easily and apply different kind of formatting on your document. Word processing software also features for spelling and grammar, Insert, delete, copy, pastes, find, replace, search, cut, copy, format, and printing. Some commonly used word processor are Microsoft Word, WordPerfect, AmiPro, etc

b. Spreadsheet Software

Traditionally, spreadsheet was simply grid of rows and columns on special paper that was used by accountant and others to produce financial projections and report. Electronic spreadsheets application allows users to enter data in rows and column, calculate means perform statistical analyses, create tables and produce other financial schedules. Spreadsheet software also has features that allow the creation of analytical graphics. When viewed on a monitor or printed out, analytical graphics, or business graphics, help make data and the generated report easily to comprehend and analyzed for decision making by management. Electronic spreadsheets have features that allow charts and graphs to be created from a table of numbers to show the significance of a selection of data which can be displayed in a number of ways: bar graphs, line graphs, and pie charts

etc. A spreadsheet document is called a worksheet. Some commonly used Spreadsheet software are VisiCalc, Lotus 1-2-3, and Microsoft Excel etc.

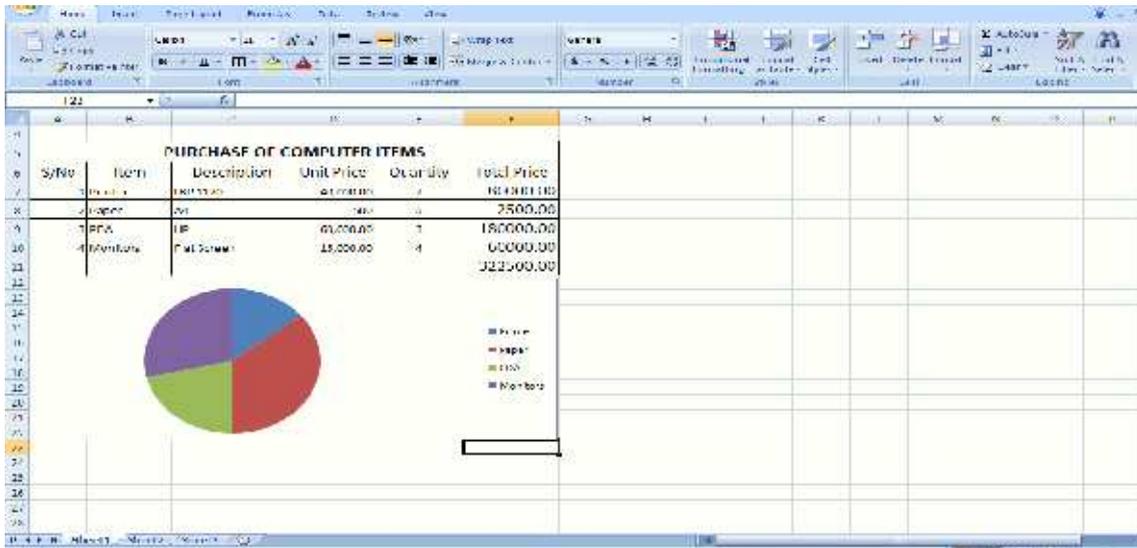


Figure 1: Microsoft Excel Chart

c. Database Software

A database management system (DBMS) is a collection of programs that allows you to store modify and extract information from a database. A database is a collection of interrelated files in a computer system. With a DBMS, activities such as updating, deleting, adding, and amending of records are easily performed. Databases can also be seen as electronic filing cabinets that allow systematic storage of data for easy access, reporting, and retrieval of records. Some commonly used database software are MS Access, Informix, Sybase, Oracle, DB3

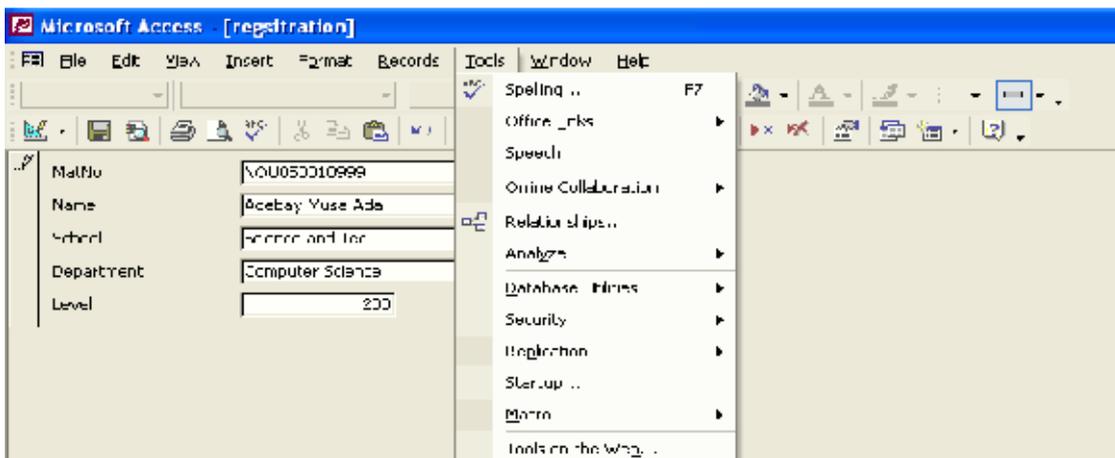


Figure 2: Microsoft Access Database

d. Presentation graphics

Presentation graphics enable users to create highly stylized graphics for slide show and report. Presentations software may make use of some analytical graphics (bar, line, pie chart) and also allow you to use electronic painting and drawing tools for creating lines, rectangles and others forms of shape. Most presentation software have features that allows you to add text, animated sequences, and sound to your report. Your report or presentation can be sent to the screen or printed on transparency acetates. Some presentation software packages provide artwork called clip art that can be electronically cut and pasted into the graphics. Some commonly used presentation graphics are Examples of well known presentation packages are Microsoft PowerPoint, Aldus Persuasion, Lotus Freelance Graphics, and SPC Harvard Graphics

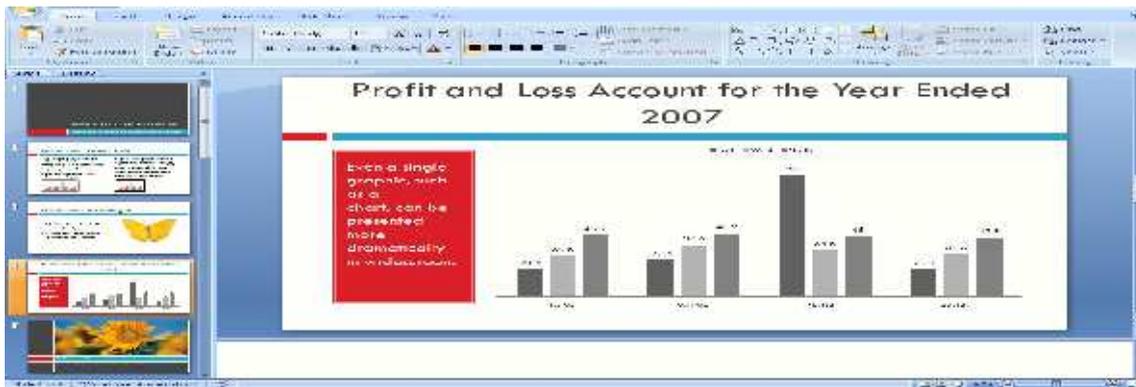


Figure 3: Microsoft Power Point Presentation

4.0 Conclusion

The data processing requirements of an organization is a major factor in determining which software to acquire or use. Software development could be one of the functions of a data processing department.

5.0 Summary

In a data processing center, different computer hardware and software are usually required for the manipulation of data. Some of the commonly used software are: word processor, database, presentation graphic, spreadsheets, operating system etc

6.0 Tutor Marked Assignment

- Q1. State four main tasks of operating systems
- Q2. State some advantages of in-house developed software over COTS

7.0 References/Further Readings

George Beekman, Eugene J. Rathswohl, Computer Confluence, Exploring Tomorrow's Technology, IT Edition, Prentice Hall, New Jersey, 2003

Brain K. Williams, and Stacey C. Sawyer, Using Information Technology, A practical Introduction to Computers and Communications, McGraw Hill, N.Y, 2005

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L Consultancy Services Pvt. Ltd, 1999

MODULE 1 UNIT 4: DATA CAP TURING AND REP ORTING DEVICES

Table of Contents

1.0	Introduction
2.0	Objectives
3.1	Input Devices
3.1.1	Keyboard
3.1.2	Pointing Devices
3.1.3	Pen-Based Devices
3.1.4	Source-Data Entry
3.2	Other input devices
3.3	Output Devices
4.0	Conclusion
5.0	Summary
6.0	Tutor Marked Assignments
7.0	References/Further Readings

1.0 Introduction

Computers interact with its external environment and users via the Input / Output devices. Inputs are the signals or data sent to the system, and outputs are the signals or results received from the system to the outside. Input devices are hardware equipment by which a computer system receives data while output devices are hardware equipment by which results are communicated to the outside world.

2.0 Objectives

At the end of this unit, you should be able to:

- Identify and describe data capturing devices
- Identify and describe data reporting/output device

3.1 Input Devices

For data to be processed, they are first to be captured via input devices and translated to computer understandable form. Input devices are use to capture and translate data into a form the computer can process. Though the data may be captured in the forms of text, characters, letters, images, sound, pictures and video, the translated data will be in the form of 0s and 1s, or off and on signals which can be easily processed by the computer. Input devices exist in the following categories:

3.1.1 Keyboard

Keyboard is used for entering text and numeric values into the computer. It is the most popular means of data entry in microcomputers. The keyboard may look like a typewriter but with some special keys added. The keyboard has facilities for converting letters,

numbers and other characters into electrical signals that are machine-readable the processor of computers. Data entry functionality is also available on via keys on a bank s automatic teller machine or the keypad of a personal digital assistant or a cell phone.

3.1.2 Pointing Devices

These constitute the second category of input hardware. These devices control the position of the cursor or pointer on the screen. Pointing devices include:

a.) Mouse

A mouse is an input device that is popularly used with microcomputers. When used with desktop computers it is rolled about on a desktop to direct a pointer and select commands on a computer display screen. The point-and-click actions of the computer mouse is fast making it an alternative to keyboard and text-based commands since most of the commands executed with a keyboard can almost be done with a mouse. In addition, the mouse can be used to draw images on the screen.

b.) Trackball

This is a variant of the mouse. It can be moved on top a stationary device that is rotated with the fingers or palm of the hand. Trackball are commonly used with hand held devices.

c.) Touchpad

Touch pads allow a user to control the cursor/point on the computer display screen with his fingers. It is about the same size as a mouse, but is usually, rectangular and flat devices that are uses very weak electrical field to be activated.

d.) Joysticks

A joystick is a pointing device that consists of a vertical handle like a gearshift lever mounted on a base with one or two buttons. Joysticks are used principally in video games, in some computer-aided design systems, and in computerized robot systems.

3.1.3 Pen-Based Devices

Pen-based computer devices use a pen-like stylus to capture a user s handwriting and marks into a computer. This is one of the means by which data is entered into small handheld computers such as personal digital assistants (PDAs).

e.) Light Pen

This is a light-sensitive stylus, pen-like device that can be connected by a wire to the computer system. A stylus is a pen-like device with which a user sketches an image. When used to capture data, the user brings the pen to desired point on the display screen and presses the pen button, which identifies that screen location to the computer.

f.) Digitizing Tablets

A digitizing tablet consists of a tablet linked by a wire to a stylus or puck. A puck is a copying device with which the user copies, or traces, an image. Digitizing tablets are used to capture data directly into the computer for processing.

3.1.4 Source-Data Entry

Keyboard entry requires typing; errors could be introduced by the operator during data entry. Non-keyboard source-data entry devices such as scanner are used to minimize data entry errors. The following categories of devices are used:

Scanning Devices

Scanners use laser beams and reflected light to capture and translate hardcopy images of text, drawings, photos, and the like into computer understandable form for processing. Scanning devices include Mark and character recognition devices, Fax machines and Imaging systems. Specifically, the Mark and character recognition devices are usually referred to by their abbreviation OCR, OMR and MICR,

g.) Optical mark recognition

Optical mark recognition (OMR) uses a device that reads pencil marks and converts them into computer-understandable form. This technology is widely accepted by many examination bodies such Joint Admission an Matriculation Board (JAMB), West African Examination Council (WAEC), NECO (National Examination Council Organization) and other Scholastic Examination bodies across the world as a convenient way of processing their candidates results.

h.) Optical Character Recognition

Optical character recognition (OCR) devices are used to translate designed marks character, and code into digital form for processing by the system. The most widely used optical code is the bar code, which is used in point-of-sale systems in supermarket, shopping mall, hospital, libraries, military operations, utility bills and price tags on departmental-stores merchandized and transportation facilities. The code can include time, date, location, and identification data for deriving appropriate information for decision making. It is usually more convenient and faster to use OCR than retyping a document into the computer.

i.) Magnetic ink Character Recognition (MICR)

This Technology is used in cheques processing by most banks and other financial institutions. An MICR device has facilities to interpret and translates magnetic ink character on Bank cheque which contains characters identifying the bank, cheques number etc into computer understandable form.

j.) Fax Machine

A fax machine also referred to as facsimile transmission machine has facilities that scan an image on paper into electrical signal, transmit same over telephone lines and re-creates the image to a receiving fax machine on paper.

3.2 Other input devices

Sensors are devices that collect data directly from the environment for input into a computer system. For instance, atmospheric data can be captured and transmitted via sensor network to a data processing centre for weather forecasting.

h.) Voice input

These devices convert spoken words into computer understandable digital form for processing. When voice recognition systems are used to capture data for processing, the user documents can be created by speaking words into a computer rather than keying them in. The system will have audio facilities such as sound card, microphone, and speakers.

The physical component or materials which data is stored are called storage media. The hardware components that write data to, and read it from, storage media are called storage devices. For example, a diskette is storage medium, whereas a diskette drive is a storage device. Storage media and devices have evolved dramatically since computers were in their infancy, and this pace has accelerated since the introduction and growing popularity of PCs.

3.3 Output Devices

Outputs are the outcome of processed data. An output device is any piece of computer hardware equipment used to communicate the results of data processing carried out by a computer to the outside world. Some common output devices include: Visual display unit
A visual display unit (also called VDU, monitor, or screen), Printers and Plotters.

a.) Monitors

The first is the typical monitor that looks like the television screen and used uses a large vacuum tube, called a cathode ray tube (CRT). Its operation is very similar to that of television picture tube, with an electronic gun shooting a beam of electrons to illuminate the pixels (picture elements) on the screen. The resolution is determined by the number of pixels per screen. The resolution of a monitor indicates how densely the pixels are packed. The second type, known as flat-panel display, is used with laptops or notebook computers. While the earlier monitors could only display one color i.e black, grayscale, etc modern monitors have good support for color and graphics display though this may require more memory.

b.) Printers

Printer is a device that prints text or illustration on paper and in many cases on transparencies and other media. They are the most popular output devices beside monitors. The speed of early printers was measured in units of characters per second while that of modern printers are measured in pages per minute. There are different types of printers based on the technology they use. We shall only describe a few that you will find in a data processing center:

i.) Dot-Matrix Printer

This operates by striking pins against an ink ribbon to create the required characters. Each pin makes a dot, and combination of dots form characters and illustrations. Dot matrix printers are cheap, relatively fast but they do not produce high-quality output.

ii.) Line Printer

This type of printer is able to print an entire line at one time with as much speed as 3,000 lines per minute. The print quality is relatively low and they are very noisy.

iii.) Thermal Printers

This type of printer produce images by pushing electrically heated pins against special heat-sensitive paper. Thermal printers are economical and are used in most calculators and many fax machines. They produce low-quality print and paper requires special handling of papers.

iv.) Ink-jet Printer

This work by spraying ionized ink at a sheet of paper. Magnetized plates in the ink's path direct the ink onto the paper in the desired shapes. They are popular with as portable printers because they could exist in small sizes. Ink-jet printers require a special type of ink that may smudge on low-cost copier paper. They provide an economical way of way to print full-colour documents.

v.) Laser Printers

Laser printer utilizes a laser beam to produce an image on a drum. The light of the laser alters the electrical signal on the drum wherever it makes a hit. The drum is made to roll through a reservoir of toner, which is picked up by the charged portion of the drum. After which, the toner is transferred to the paper through a combination of heat and pressure.

LaserJet printer produces very high quality output. This is as a result of the resolution that is, how many dots per inch (dpi) they lay down. The available resolutions range from 300 dpi at the low end to 1,200 dpi at the high end. The color laser printers are able to produce colored outputs though they are usually more expensive because of the addition toners required.

Laser printers are non-impact printers; they are not as noisy as dot-matrix or line printers. They are relatively fast, although not as fast as some dot-matrix printers. Their speed ranges from 4 to 20 pages of text.

c.) P lotter

A Plotter, though mainly used for engineering applications uses pen to draw picture or lines on paper based on commands from a computer. The can produce continuous lines, whereas printers can only simulate lines by printing a closely spaced series of dots. Plotters are not as fast as printer but are useful in producing large-size chart, maps, or drawings even in color.

d.) Speaker

A voice output device such as a speaker converts digital output data back into intelligible speech. They are use to deliver audio output from animation, multimedia application from the web and music to the user.

e.) Microfilms and Microfiche

Microfilm and microfiche were used to store large quantities of output as microscopic filmed document, but they are now being replaced by optical disk technology.

4.0 Conclusion

Input / Output refers to the communication between a data processing system such as a computer, and the outside world. Users interact majorly with computer systems through input and output devices.

5.0 Summa ry

This unit covers a wide range of devices that can be used for capturing data. The choice of input device will depend on the type source. For example, in capturing bank transaction data from a cheque a MICR device will be most suitable. Also we covered various output devices such as printers, plotters, and speakers.

6.0 Tutor Marked Assignments

Q1. List the input and output devices you would expect to see in an examination processing

IT centre?

Q2. List 3 input devices and 2 output devices that will needed for data processing in a large

Supermarket

7.0 References/Further Readings



PDF
Complete

*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Extended Features](#)

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L
Consultancy Services Pvt. Ltd, 1999

MODULE 2 UNIT 1: DATA STORAGE DEVICES

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Diskette
- 3.2 Hard disks
- 3.3 Removable hard disks
- 3.4 Magnetic tape
- 3.5 Optical Storage
- 3.6 Accessing Data from Disk
- 3.7 Disk Performance
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

The computer primary memory is limited in the amount of information it can store at a time. In addition, the content of the RAM is temporary i.e., once the power of the computer is turned off, all the data and program within it simply vanishes. This is why data must be stored on secondary storage devices which are able to hold data and programs on a more permanent basis. External storage devices consist of large-capacity, slow-access data storage attached to a digital computer and used to store information that exceeds the capacity of main storage. Two main technologies are used to store data today: magnetic and optical storage. Although devices that store data typically employ one or the other, some combine both technologies. The primary types of magnetic storage are:

2.0 Objectives

At the end of this unit, you should be able to:

- Identify and describe data storage devices
- Parameters for disk performance

3.1 Diskette

A diskette or floppy disk is a data storage medium that is composed of a disk of thin, flexible magnetic storage medium encased in a square or rectangular piece of mylar plastic. Floppy disks are read and written by a floppy disk drive or FDD, the initials of which should not be confused with "fixed disk drive", which is another term for a hard disk drive. Floppy disk exist in the following sizes; 8-inch (200 mm), 5¼-inch (133 mm), and the most common 3½-inch (90 mm). Though floppy disks are still used

in some data processing environment, they are now being largely superseded by flash and optical storage devices while some users consider emails as a convenient way of exchanging small to medium size digital files. Floptical drives combine magnetic and optical technologies to store about 21MB of data on a media similar to 3½-inch floppy disks. The name is a portmanteau of the words 'floppy' and 'optical'. This device was introduced in 1989 by Insite Peripherals of San Jose, but did not become popular, because of the limited storage capacity it offer. A similar technology was used in the Laser Servo-120 drive introduced in 1996 with 120MB capacity.

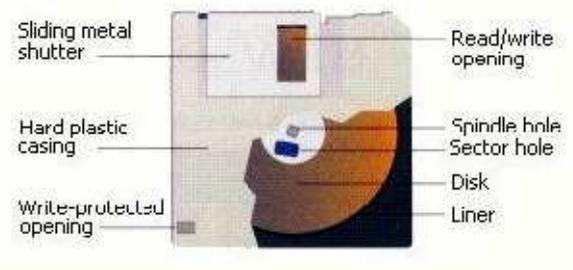


Figure 1: Floppy Disk

3.2 Hard disks

In a data processing environment, a hard disk drive (HDD), commonly referred to as a hard drive, hard disk or fixed disk drive, serve as permanent storage device for large amount of data. Originally, the term "hard" was temporary slang, substituting "hard" for "rigid", before these drives had an established and universally-agreed-upon name. The hard disk drive (often shortened to "hard drive") and the hard disk are not the same thing, they are packaged as a unit and so either term is sometimes used to refer to the whole unit. A hard disk is really a set of stacked "disks," each of which, like phonograph records, has data recorded electromagnetically in concentric circles or "tracks" on the disk. A "head" (something like a phonograph arm but in a relatively fixed position) records (writes) or reads the information on the tracks. Two heads, one on each side of a disk, read or write the data as the disk spins. Each read or write operation requires that data be located, which is an operation called a "seek." (Data already in a disk cache, however, will be located more quickly.) Modern computers come with a hard disk that contains several billion bytes (gigabytes) of storage.



Figure 2: Hard Disk

3.3 Removable hard disks

This is a variation of the hard disk in which hard disks enclosed in plastic or metal cartridges are easily removable like floppy disks. It combines the best features of hard and floppy disks. They are used to provide large economical, high, fast, and portable storage facility for data processing.

3.4 Magnetic tape

Magnetic tape has historically been found more convenient means of large data storage over disk where media portability or removability is required for backup. Magnetic Tape uses the same read/write techniques as disks. Data is stored on flexible mylar tape covered with magnetic oxide. Data is stored in parallel tracks of 9, 18 or 36. Data on tapes are accessed sequentially. Tapes provide slow, very cheap, large capacity backup for data. The rapid advances in disk storage technologies resulting in and the improvement in disk storage density, and reduced price, coupled with arguably declining innovation in tape storage technology, has reduced the market share of tape storage devices.



Figure 3: Tape

3.5 Optical Storage

The optical storage devices such as CDs and DVDs are means in which data is written and read with a laser for archival or backup purposes. The optical storage devices are fast replacing both hard drives in computers and tape backup in mass storage. This is because optical media are more durable than tape and less vulnerable to environmental conditions lasting up to seven times as long as traditional storage media. However, at present optical media are slower than typical hard drive speeds, and offer lower storage capacities. Optical disk capacity ranges up to 6 gigabytes i.e. 6,000,000 bytes which is far more compared to the 1.44 megabytes (MB) i.e., 1,440,000 bytes offered by a floppy disk. A newer technology, the digital versatile disc DVD, has about 4.7 gigabyte storage capacity on a single-sided, one-layered disk compared with 65 gigabyte of storage for a CD-ROM disk. Invariably, they can be used to hold large amount of data.



Figure 4: Optical Storage

A USB flash drive is a NAND-type flash memory data storage device integrated with a USB (universal serial bus) connector. USB flash drives are typically removable and rewritable, much shorter than a floppy disk (1-4 inches or 25-102 mm), and weigh less than 56g. Their storage capacities typically range from 64MB to 32GB or more. They have 10-year data retention. USB flash drives offer potential advantages over other portable storage devices, particularly the floppy disk. They are more compact, faster, hold more data, are more reliable for lack of moving parts, and have a more durable design.



Figure 5: A Flash

3.6 Accessing Data from Disk

Bits of data (0s and 1s) are stored on circular magnetic platters called disks. A disk rotates rapidly (& never stops). A disk head reads and writes bits of data as they pass under the head. Often, several platters are organized into a disk pack (or disk drive). Disk contains concentric tracks. Tracks are divided into sectors. A sector is the smallest addressable unit in a disk.

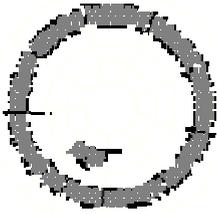


Figure 6: Disk Organization

When a program reads a byte from the disk, the operating system locates the surface, track and sector containing that byte, and reads the entire sector into a special area in main memory called buffer. The bottleneck of a disk access is moving the read/write arm. So it makes sense to store a file in tracks that are below/above each other in different surfaces, rather than in several tracks in the same surface.

Figure 7: Cylinder

A cylinder is the set of tracks at a given radius of a disk pack. i.e. a cylinder is the set of tracks that can be accessed without moving the disk arm. All the information on a cylinder can be accessed without moving the read/write arm.

3.7 Disk Performance

In measuring the performance of a storage device, we may consider the following three parameters.

a.) Rotational delay/Latency

This is the time it takes to position the proper sector under the read/write head. In general, it is used to refer to the period of time that one component in a system is spinning its wheels waiting for another component. Latency, therefore, is wasted time. It makes sense to separate read latency and write latency, and in case of sequential access

storage, minimum, maximum and average latency. Consider a hard disk which rotates at about 5000 rpm i.e. one revolution per 12 msec. The average latency can be calculated as follows:

Min latency = 0

Max latency = Time for one disk revolution

Average latency (r) = $(\text{min} + \text{max}) / 2$

= $\text{max} / 2$

= time for $\frac{1}{2}$ disk revolution

Typically 6 – 8 ms average

b.) Throughput

This is the rate at which information can be read from or written to the storage. It is expressed in terms of megabytes per second or MB/s. A media accessed sequentially, as opposed to randomly, typically yield maximum throughput.

c.) Seek time

This is the amount of time between when the CPU requests a file and when the first byte of the file is sent to the CPU. Seek times between 10 and 20 milliseconds are common.

4.0 Conclusion

Computer data storage, often called storage or memory, refers to computer components, devices, and recording media that retain digital data used for data processing. Secondary storage differs from primary storage in that it is not directly accessible by the Central processing unit. Secondary storage does not lose the data when power suddenly goes down i.e. it is non-volatile. Secondary storage is not as expensive as primary storage consequently, a higher magnitude of data are stored in secondary storage than primary storage in data processing centers.

5.0 Summary

This unit covered computer data storage devices and their performance measurement. The knowledge acquired will be useful in the following units.

6.0 Tutor Marked Assignment

- Q1. When do you require a Secondary storage?
- Q2. Explain the meaning of the term Latency

7.0 References/Further Readings

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L Consultancy Services Pvt. Ltd, 1999

<http://computer.howstuffworks.com/hard-disk7.htm>

MODULE 2 UNIT 2: FILE ORGANIZATION

- 1.0 Introduction
- 2.0 Objectives
- 3.1 What is a Computer file?
- 3.2 File Organization
- 3.3 Classification of Storage Devices
- 3.4 Types of data files
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

It is not enough to acquire excellent computer system; hardware and software. Files within the system must be well maintained, managed and organized. Good file organization leads to productivity. When files are well arranged and maintained, users can easily access and retrieve the information they need in good time. Different ways of file organization are: Serial File Organization, Sequential File Organization, Indexed Sequential Organization, Direct or Random File Organization.

2.0 Objectives

At the end of this unit, you should be able to:

- Explain the meaning of a computer file
- Identify and describe file organization techniques
- Describe the type of data files used in a Data processing Environment

3.1 What is a Computer file?

A file is collection of data or information that has a name called the filename. Most of the information stored in a computer system is stored as files. A file is often stored with a user's given name and a system-supplied extension. The name of the file should reflect the content of the data stored in the file. For example, payroll or result File. The extension of the file should reflect either the type of file (i.e. program file, image file, audio file etc) or the software used to create the file (e.g. MS WORD document, MS EXCEL Worksheet, BASIC Compiler). For example;

S/No	Type of File / Software	Extension
1	Microsoft Word Document	.doc
2	Audio file / Video	.wav , avi, mp3

3	BASIC compiler	.bas	
4	Text		.txt
5	Microsoft Excel	.xls	
6	Images		.png , jpg , gif

Table 1: Computer Files

The different types of file stored in a computer system are as follow:

a.) Program File

These are files that store sets of instructions written in a programming language. A source program file for examples contains the instructions written in a high level language such as BASIC or FORTRAN programming language by a programmer while the object file is the translated form of the source file in machine code after. The files that contain the machine code are called executable files (or binary files)

b.) ASCII File

ASCII stands for American Standard Code for Information Interchange. ASCII files are text-based files. That is the characters are represented in ASCII code (without formatting such as underline, italics, boldface or graphics). Files stored in this format are used to transfer documents between incompatible computer platforms, such as IBM and Macintosh.

c.) Image File

Documents containing digitized graphics or images are stored in this format.

Image files: Image files contain digitized graphics

Audio and video files: Audio files contain digitized sound while video files contain digitized video images and animation.

d.) Audio and Video File

This is a file that is used to store digitized sound or digitized video images and animation.

e.) Data File

This refers to document files, contain data, not programs. Their content are using application software.

3.2 File Organization

File are created, arranged and maintained in data processing systems for a purpose of easy retrieval. Computer systems store files permanently on secondary storage devices. Records or files are arranged in several ways on the storage media, and the arrangement

determines the manner in which individual records can be accessed or retrieved. Four common ways of file organization and access are:

a) Serial File Organization

In this method of file organization records are not arranged in any specific order. If magnetic tapes are used for the storage of data, it would be necessary to wind the tape forward and backward to access a given record since access can only be made in the sequence in which the records were physically stored on the tape, i.e serially. And if records are stored on disk, a full index will be required to access any given record. This method of file organization is therefore inefficient.

b.) Sequential File Organization

In this type of file organization, data records are normally stored in ascending order of key field. Data must be retrieved in the same physical sequence in which they are stored. It is the only file organization method that can be used on magnetic tapes. Magnetic tape is a sequential storage device. That is, records and files are stored in magnetic tape in a sequential order. They are also read in sequential order. Note that records may also be stored sequentially on disk if desired. Serial and sequential file access mean the same thing in respect of files stored on tapes when stored in sequence but this may not be the same case with disk files as the records accessed serially may not be defined in a key sequence. This sequential file organization method is no longer a popular approach of storing or access record in a file.

c.) Indexed Sequential Organization

This technique of file organization uses both the sequential and direct access methods. It is widely applied to the storage of record on magnetic disk. It allows sequential file to be manipulated serially as the record are stored in ascending order of key field. In addition it allows direct access storage devices, to be accessed directly using the indexed sequential access method (ISAM). This access method relies on an index or key fields to locate individual records. An index to a file is similar to how of a book can be used to locate it physical position //on a library shelf. The method requires that data are stored in magnetic or optical disk. For //example, a university could index certain ranges of student s matriculation number 0000 to 1000, 1001 to 2000 and so on. For the computer to find the record with the key field 8888, it would go first to the index, which would give the location of the range in which the key field appears (for example, 8001 to 9000). The computer would then search sequentially (from 8001) to find field 8888

d.) Direct or Random File Organization

This file organization is utilized with magnetic disk technology. Most computer applications today use this approach for storing records in computer files. In this approach, individual records are stored in a particular sequence of key fields. Thus allowing users to access records in a sequence they desire, without regards to actual physical order on magnetic tapes or disk. With this approach, every record has an address

that makes it possible to locate it independently of other records on the storage media. To allow easy access and retrieval of information an index or table of the key are maintained with the relative record number of the record in storage. In order to retrieve a record, the actual key is looked up in the index with the corresponding record number of the record that matches the key. Once this is found, the address in storage is worked out and the record accessed. Records stored with the technique are much faster to be accessed than records store with the sequential file organization. Though they may be more expensive because optic or magnetic disk that may be involved for their storage.

3.3 Classification of Storage Devices

Storage devices can be classified generally as sequential access or random access. For example, a tape drive is a sequential-access device because to get to record 5 on the tape, the drive needs to pass through points 1, 2, 3 and 4. A disk drive, on the other hand, is a random-access device because it allows the record at any position on the disk to be accessed without passing through all intervening positions

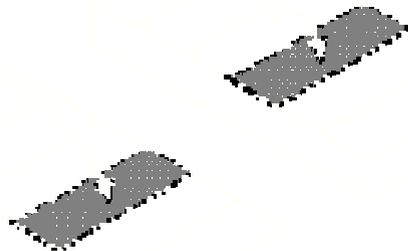


Figure 1: Storage Device Access

3.4 Types of data files

Data stored in a data processing centre could be considered as transaction file and master file

a.) Transaction file

This refers to a collection of transaction records. The transaction file is a temporary holding file that stores records that normally have a limited useful lifetime. For an employee file for a payroll processing, a transaction file would hold, the name, contact

information, hour worked, pay rate, tax, utility bills etc for a staff for a particular month. At the end of every month or so, there will be a need to compute the salaries of the staff from the information in the transaction file. After the transactions are successfully carried out, the information in the transaction file will be used to update the master file. In a data processing system, transaction records may be retained online for some period and later achieved off-line on permanent storage devices. Transaction files can serve as audit trails and history for the organization.

b.) Master File

Master file is a collection of records that are relatively permanent records that are updated periodically. Thus once a record has been added to a master file, it remains in the system indefinitely. The value of fields for the record will change over its lifetime, but the individual records are retained indefinitely. Master files contain descriptive data, such as name and address, as well as summary information, such as students Cumulative, Grade Point Average in an examination processing system or total net pay, total tax deductions in a payroll system. The changes to be made to a master file could be addition of records, deletion of record or update of record. In an organization this will occur when a new staff joins the work force or when a staff resigns his appointment. Furthermore, the master file of a payroll system may be composed of discrete pieces of information (such as a name, address, or employee number etc) called data elements. Data are keyed into the system, updating the data elements periodically. The elements on the master file are combined in different ways to make up periodic reports of interest to management and government agencies or to generate paychecks sent to the staff at the end of the month. Other examples of master file include: Customer, Product, Result or Supplier file.

4.0 Conclusion

An effective data processing system is one provides users with timely, accurate, and relevant information. The type and size of devices used for data storage determines how fast information can be retrieved. Two types of data files are described: Transaction and master file -The data in transaction files are regularly used for generating information while data that require periodic updates are moved to the master file.

5.0 Summary

In this unit, we explained the meaning of a computer file, described the different types of file organization, and explain the meaning of the terms: transaction and master file. These details are vital for the comprehension of the contents in the subsequent units.

6.0 Tutor Marked Assignment

- Q1 Define the term a computer file
- Q2. Explain three main techniques of file organization

7.0 References/Further Readings

Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, Systems Analysis and Design Methods, McGraw Hill, New York, 2004,

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L Consultancy Services Pvt. Ltd, 1999

MO DULE 2 UNIT 3: DATA P RO CESSING TECH NIQUES

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Meaning of Data Communication
- 3.2 Batch Processing
- 3.3 Online processing
- 3.4 Transaction Processing:
- 3.5 Real-time Systems
- 3.6 Distributed Processing
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

The processing of data need not be restricted to a particular environment. Data processed in remote locations can be accessed via telecommunications infrastructure. Different kind of processing techniques are made possible by telecommunications. The techniques are discussed in this unit.

2.0 Objectives

At the end of this unit, you should be able to:

- Definition and explanation of the terms; data and information
- Describe a computer system
- Identify the types of computer
- Describe the basic operations of a computer

3.1 Meaning of Data Communication

Data communication refers to the process of transmitting data / information from a source via a telecommunications system in coded forms to a destination for the purpose of interchanging details of business transaction. The transfer of data require some forms of electromagnetic energy such as electricity, light, radio or waves which is transmitted through a physical medium such as the atmosphere, cable or wire

A telecommunication system refers to a collection of compatible hardware and software required to exchange information from one location to another. A telecommunication system is able to transmit text, graphic images, voice or video information. The major components of a telecommunication system are.

- Computer: This is required for data processing
- Terminals: These are any input/output device that send or receive data
- Communication channel: This is the link by which data, graphic images, voice or video are transmitted between sending and receiving devices in a network. Communication channels use various communications media, such as telephone lines, cable, coaxial cable, fiber-optics and wireless transmission
- Communications software: This controls input and output activities and manages other functions of the communication network

Question: Describe the basic components of a telecommunication system

3.2 Batch Processing

This used to be the dominant form of data processing. It is a method of data processing, in which transaction data such as data from time card is collected over a period say, a month for payroll processing, and then processed all at once as a batch to update the master file. A payroll system keeps tracks of the money paid to employees. With this kind of processing, a user may not be able to receive a reliable response from the data processing system until the batch is processed. For instance, if the pay of a staff is computed at the end of the month, it will not be possible to know exactly how much the staff will earn until all the data elements for computing his pay are collated and processed. Another example is a banking system where reconciliation of accounts is done only after the close of day operations and the entire transactions is processed as a batch. Any queries performed before the batch is processed will certainly not be accurate. However, many banks in Nigeria have begun to integrate real-time modules into their applications.

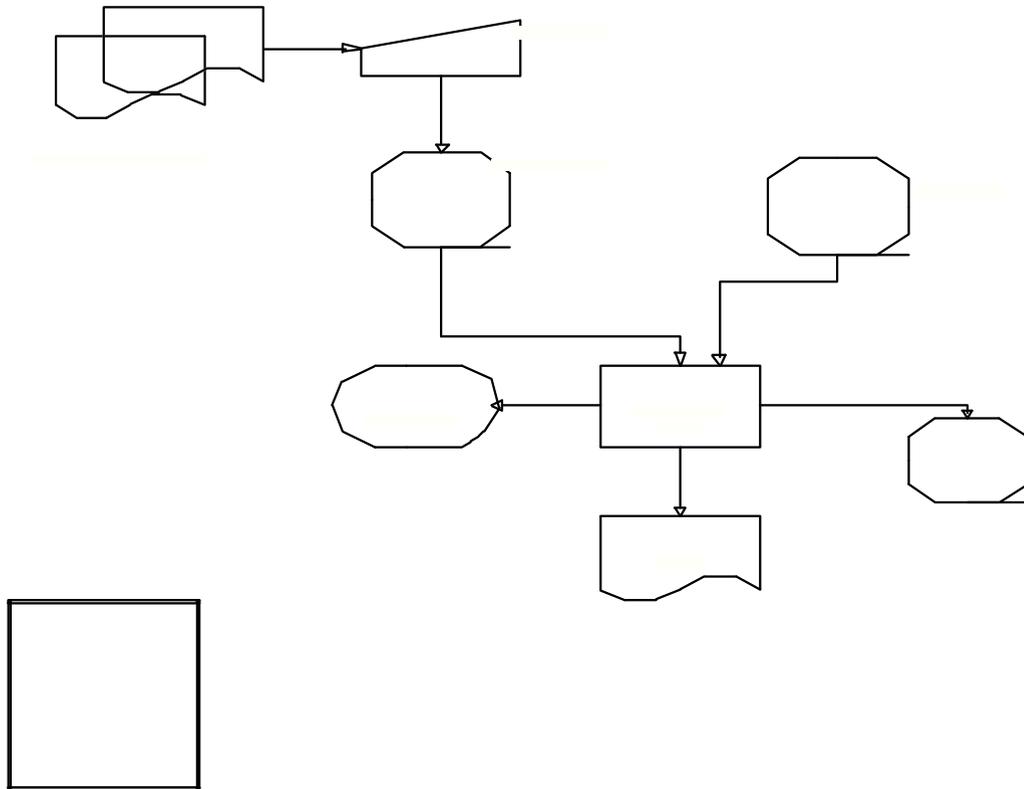


Figure 1: Batch Processing

3.3 Online processing

This is a method of data processing method whereby data about a single transaction is processed immediately it is captured. This method of processing allows transactions to be entered directly to the system via terminals, PCs or workstations as they take place thereby updating the master file immediately as the transactions occur. The point of entry may be remote from the location at which update his made. For example when you withdraw cash from an automated teller machine, your withdrawal is instantly processed and your account balance updated. Other online processing systems include:

- Visa processing
- Result checking
- Banking (account enquiry)
- Air seat reservation
- Application processing
- Examination

Because of the prevalent of PC in data processing we rarely hear the term online processing. The term client / server computing is more popular where the PC is the client.

3.4 Transaction Processing :

Transaction processing system (TPS) is an automated system that performs and records the daily or routine transactions necessary to conduct a business. A transaction may include an order or payment confirmation. Transaction processing entails the updating of appropriate database records as soon as a transaction is captured into the computer. The processing of a transaction may require that a confirmation be sent to the user or a customer immediately. Transaction processing systems are the backbone of an organization because they update constantly. The instant update is a requirement because at any given moment, a customer may need an inventory balance, an account balance or the result of an examination. Transaction processing is also referred to as "online transaction processing" (OLTP). This means that the system should be available 24x7 on reliable computers and networks. Common examples of transaction processing systems include: Sales order entry, hotel reservation systems, payroll, examination processing systems, and shopping systems.

3.5 Real-time Systems

These are systems designed to deal with dynamic situations in order to control a critical operation such as an airline reservation system which must be continually updated as events occur. Seat reservation in flight operations require communication-oriented Server computers supported by network of terminals or PCs serving as clients. These facilitate response to enquiries on seat reservations and ensure that the master file is updated as soon as transactions are completed. The systems ensures that enquires on available seat are responded to instantaneously and prevents double, or overbooking of seat in the aircraft.

3.6 Distributed Processing

Many organizations that were used to centralized systems for data processing are now able to adopt Distributed data processing because of advances in computing technologies. A centralized system consists of a central multi-user computer (usually mainframe) which hosts all components of a data processing system. The users interact with this host via terminals or PCs serving as client, but virtually all of the actual processing and work is done on the host computer. All the devices in the centralized system such as PCs, terminals, network devices, and printer converge on one central computer, even though the users may work from remote locations via terminals. All processing and storage take place at the central location. On the other hand, a distributed system allows the components of data processing system to be made available at multiple locations in a computer network. Which means that, the processing workload required for supporting the components is also distributed across multiple computer on the network. In addition, the computers, storage devices, and even some computer personnel may need to be distributed to separate locations throughout the organization for the efficiency of the system. Distributed data processing allows data processing and storage to occur at several locations in the computer system. There are advantages and disadvantages associated with adopting distributed data processing in an organization. And these are as follow:

Advantages of Distributed Data Processing

- Modularity Easy Integration
- Better response time Data processing is closer to the end user.
- Ability to share data
- Greater reliability
- Local control of data
- Lower cost
- Direct Users interaction

Disadvantages of Distributed Data Processing

- Technical problem of connecting and maintaining dissimilar systems
- Need for sophisticated communication system may incur addition cost
- Data integrity and security problem resulting from possible duplication of data on several systems along the network
- Interoperability issues- technical problem of connecting dissimilar machines
- Lack of skilled professional for systems support

4.0 Conclusion

Some of the processing techniques studied in this unit are closely related but with a little difference in meaning. For example, an on-line system need not be a real-time system but a real-time system must be an online system.

5.0 Summa ry

An important feature in modern organizations is the need for users to access and use data from different computers at anytime. Timely availabilit y of information enhances decision making. In this unit we studied the main data processing techniques.

6.0 Tutor Marked Assignment

Q1. What are the advantages and disadvantages of Distributed Data Process?

Q2 Explain the terms

Batch Processing

Online processing

Transaction Processing:

Real-time Systems

7.0 References/Further Readings

R G Anderson, Data Processing, Volume 1: Principles and Practice, Pitman Publishing, Singapore, 1990

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L
Consultancy Services Pvt. Ltd, 1999

MO DULE 2 UNIT 4: TRADITIO NAL FILE SYSTEM PROCESSING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Traditional File Systems / Processing
- 3.2 Problem with the Traditional File Environment
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introductio n

File systems were really an early attempt to computerize the manual systems that were prevalent in the early days of data processing. The manual system works well while there are not many items, but it breaks down when we have a lot of items and we want to cross reference them. So the file based system was a response to industry needs for more efficient data access. However, rather than have everything centralized, each department would have their own set of files. File system which is also referred to as a traditional approach of file processing is inadequate for many organizations because it involves creating, organizing, storing, manipulating and maintaining records within an organization that leads to each functional area or department creating and maintaining its own data files and programs.

2.0 Objectives

At the end of this unit, you should be able to:

- Explain the concept file system
- Identify and explain the problems with the file system

3.1 Traditional File Systems / Processing

As an organization grows, computer systems and application become more complex. Take an example, a University s computer system that handles student s information. If the data of students are to be kept independently by the units that each student interacts with, for example, Registration, Hostel, Accounts, Examination and Records, Students Affairs, Health Centre etc. Worse still, of each department is allowed to keep students information independently in their application, a time comes, when multiple files containing the same records of students will exist in the different units.

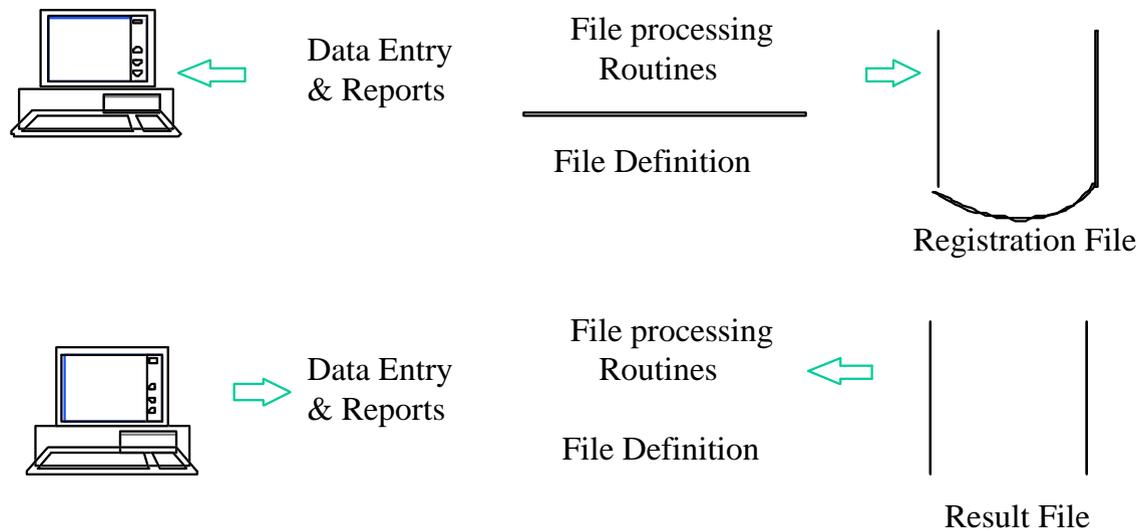


Figure 1: File System

This diagram shows two departments: registration and exams and records with each department having their own set of files and programs written for them.

As this process goes on for more than five years depending on the number of student s enrollment, the university is saddled with hundreds of programs and applications without adequate control on the data The University will be collecting the same information in more than one file. Some of the resulting problems are data redundancy, lack of data integrity and lack program-data dependence, inflexibility, and inability to share data among applications.

3.2 Problem with the Traditional File System

Some problems with the tradition file environment are: Data Dependence, Program-Data Dependence, and Difficulty of Data-Sharing. These are discussed as follow:

a) Data Redundancy:

This means the presence of duplicate data in multiple data files and often in different format. This is often the result, when different departments are allowed to collect the same piece of information about an object. For instance, within the university environment, the hostels and student registration department might collect the same student s information (Name, Mat No, Level and Address). Because it is collected and maintained in so many different places, the same data items may be repeated in different departments. When data fields are repeated in different files, storage spaces is wasted and a lot of time is spent trying to update the records.

b.) Program-Data Dependence

Program-Data Dependence is the tight relationship between data stored in files and the specific programs that process the information in the files. Where computer programs become so data specific, any changes in data would also mean any modification of in the program the processes them. Such changes could be very expensive in terms of the time and cost of re-programming

c.) Difficulty of Data-Sharing

It is difficult to share data in a file environment because it may be difficult to relate the data in one file with that of another within one or several departments were files are kept.

In addition, there is no control over the access to data which makes it further more difficult to retrieve the desired information.

4.0 Conclusion

The file systems lead to a situation, where an organization collects the same information in far too many files. Some of the resulting problems are Data Dependence, Program-Data Dependence, and Difficulty of Data-Sharing amongst several others.

5.0 Summary

In a data processing center, there is a need to plan for the data will be managed form the on set. Most organizations began information processing on small scale, automating one application at a time until there is no more control over the data. Understanding the problems of file based systems may prevent us from repeating these problems. Traditional file based systems are the predecessor to database systems. We shall discuss the database systems in the next unit.

6.0 Tutor Marked Assignment

Q1. Explain three main problems with the Traditional File System

7.0 References/Further Readings

Alexis Leon and Matthew Leon, Fundamental of Information Technology, L & L Consultancy Services Pvt. Ltd, 1999

R G Anderson, Data Processing, Volume 1: Principles and Practice, Pitman Publishing, Singapore, 1990

MODULE 3 UNIT 1: DATABASE PROCESSING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Database Management System
- 3.2 Components of a Database Management System
- 3.3 Database Model
- 3.4. Demerits of DBMS
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

The database technology is an answer to the traditional file system. It helps to solve many of the problems associated with the traditional file organization. A database is a collection of more than one related files which are usually integrated. Also, a database can be seen as a collection of data organized to serve many applications efficiently by centralizing the data and minimizing redundant data. So, instead of storing data in separate files for each application, data are stored physically to appear to users as if they are being stored in only one location. In this way, a single database would be made to service multiple applications. Database technology has a lot to offer in the management of an organization's data though not without some demerits.

2.0 Objectives

At the end of this unit, you should be able to:

- Explain the meaning of the term database
- Meaning of the term Database Management System (DBMS)
- Components of the DBMS environment.
- Identify merits and demerits of DBMS

3.1 Database Management System

A database is a collection of data stored in a standardized format, designed to be shared by multiple users. A database management system (DBMS) is software that defines a database, stores the data, support a query language, produce reports, and create data entry screen. Some of the most challenging problems in building applications arise in storing and retrieving data. Problem include conserving space, retrieving data rapidly, sharing

data with multiple users at the same time, and providing backup and recovery of the data. A Data Base Management System being a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database helps to resolve the challenges of building data processing applications. The DBMS acts as an interface between application programs and the physical data files.

Let take the student information system, instead of storing data in separate system and separate files for medical, hostel, finance, exams and records, registration, the university would only need to create a database.

In the figure below as single database is created for the processing of students results and the handling of registration information.

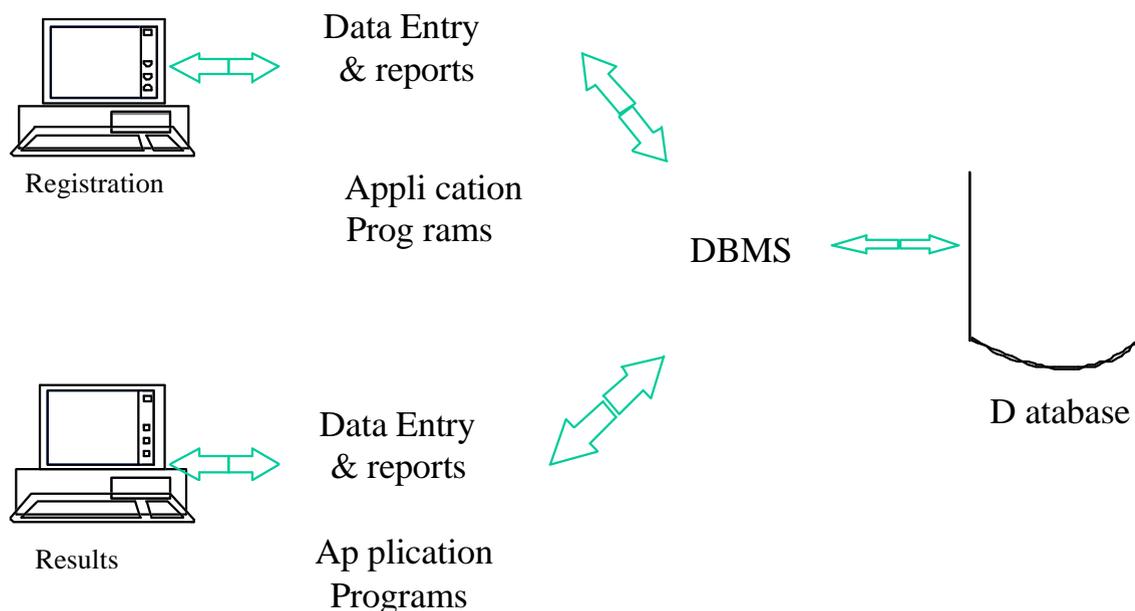


Figure 1: DBMS

When the application programs calls for a data item such as Matric no, the DBMS finds the item in the database and presents it to the application program. With the traditional data files, the programmer would need to provide adequate information in the program of how the computer would locate a data item that has already been defined. With the use of a DBMS, most of the data definition statements found in traditional programs would not be necessary. A DMBS makes it possible to do many routine task that otherwise would be difficult, time consuming and boring without the services of the computer.

A DBMS provides a convenient interface to create and maintain a database and enable individual business applications to extract the data they need without having to create separate files or data definitions in their computer programs. Examples common DBMSs include: Oracle, DB2, Sybase Adaptive Server Enterprise, DataWasp, FileMaker, Firebird, Ingress, Informix, Microsoft Access, Microsoft SQL Server, Microsoft Visual FoxPro, MySQL, PostgreSQL, Progress, SQLite

3.2 Components of a Database Management System

A database management system has three components; A data definition language, Data manipulation language and A dictionary

- a.) A data definition language is the component of a database management system that defines each data as it appears in the database. A data definition language is the formal language used by programmer to specify the content and structure of the database. The data definition language defines each element as it appears in the database before the data element is translated into the forms required by application programs.
- b.) A data manipulation language is a language associated with a database management system that is employed by end users and programmers to manipulate data in the database. Most DBMS have a specialized language called a data manipulation language that is used in conjunction with some conventional third or fourth generation programming language to manipulate the data in the database. The language contains commands that permit end users and programming specialist to extract data from the database to satisfy information requests and develop applications. The most prominent data manipulation language today is Structured Query Language, or SQL. Complex programming tasks cannot be performed efficiently with typical data manipulation language. Most mainframe DBMS are compatible with COBOL, FORTRAN and other modern programming languages, permitting greater processing efficiency and flexibility.
- c.) Data dictionary This is an automated or manual file that stores definitions of data elements and data characteristics such as usage, physical representation, ownership (who in the organization is responsible for maintain the data), authorization, and security. It is the third element of a DBMS. Many data

dictionaries can produce lists and reports of data utilization, groupings, program location, etc.

3.3 Database Model

We shall examine four major database models:

a.) Relational Database

The relational database model is the most recent of the three database model and overcomes some the limitations of the other two models. In relational database, files are organizes in tables called a relation which is a two-dimensional table of data consisting of columns and rows. A relation is also called an entity or record . A record is made up of related fields that are uniquely identified by field name also referred to as data element or attributes. The fields are used to store data containing values relating to a particular relation which may be employee or student registration record. Table 10.1 shows a student record database which holds details of students Matriculation number, Name, department and level.

	Matric_No	Name	Department	Level	
Records	NOU050010456	Sonekan Tunde	Accounting	400	
	NOU050010457	Daramola Kemi	Mathematics	400	
	NOU050010459	Akinlabi Oluwafemi	Economics	400	

Fields

Figure 2: Relational Database

b.) Hierarchical Database

In hierarchical database, records are organized in a treelike structure by type. The relationship between records types is said to be a parent-child relationship, in which any child type relate only to a single parent type. A record is subdivided into segments that

are connected to each other in one-to-many parent-child relationship. The most common hierarchical DMBS is IBM S IMS (Information Management System).

c.) Network Database

The network model is a database model conceived as a flexible way of representing objects and their relationships. It is similar to the hierarchical structure except that any one record type can relate to any number of other record types. Network data model present a logical database model that is useful for depicting many-to-many relationship. Like the hierarchical structure, the network database structure is used in older mainframe system.

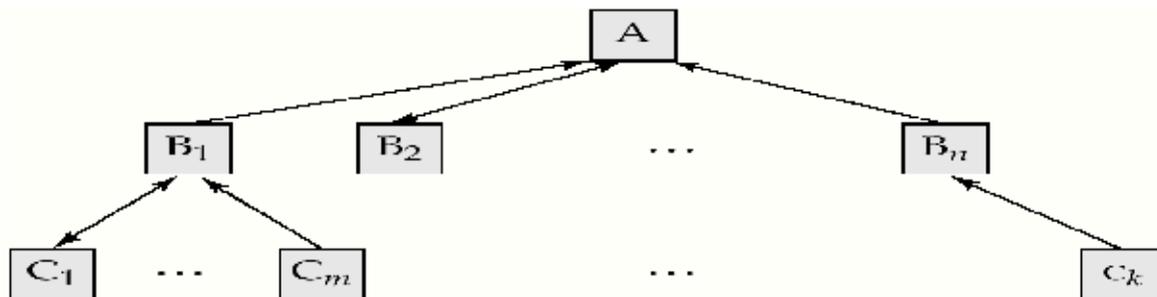


Figure 3: Network Database

d.) Object-oriented database:

An object-oriented database is a newer structure that recently has been generating a great deal of interest. This structure groups data items and their associated characteristics, attributes, and procedures into complex items called objects. Physically, an object can be anything: a product, an event, a house, an appliance, a textile, an art piece a toy, or a customer complaint. An object is defined by its characteristics, attributes, and procedures. Objects characteristics can be test, sound, graphics, and video. Examples of attributes might be color, size, style , quantity, and price. A procedure refers to the processing or handling that can be associated with the object.

3.3 Merits of a DBM S

- Encourages strategic use of corporate data
- Reduced complexity of the organization s information systems environment
- Reduced data redundancy and inconsistency
- Application-data independence

- Reduced application development and maintenance costs
- Improved flexibility of information systems
- Avoids duplicate data
- Increased access and availability of data and information
- Logical & Physical data independence Control of data redundancy
- Data consistency
- Sharing of data
- Improved data integrity
- Improved maintenance through data independence.
- Minimal data redundancy
- Integration of data
- Improved integrity
- Consistent security
- Increased productivity

3.4. Demerits of DBMS

- Complexity
- Cost of DBMS
- Cost of conversion
- Higher impact of a failure
- Additional Hardware Costs
- Take a long time to design and implement
- Performance
- Experts -Specialized Personnel
- Potential organizational Conflict

4.0 Conclusion

In an ideal database management system, the data in the database are defined only once and used for all data processing applications whose data reside in the database, thereby eliminating data redundancy and inconsistency. As data processing applications request data elements from the database, the data elements called for by the application are found and delivered by the DBMS. This enhances both the productivity of the application developer and end user.

5.0 Summary

The use of a suitable Database Management System for data processing can reduce program-data dependence along with program development and maintenance cost. Access and availability of information can be increased because user and programmers

can perform impromptu queries of data in the database. However, the use of DBMS may lead to additional hardware costs, higher impact on failure and the introduction of some complexities amongst other demerits.

6.0 Tutor Marked Assignment

Q1. What do you consider as the major differences between hierarchical database and network database?

7.0 References/Further Readings

Gerald V. Post, Database Management Systems, Designing and Building Business Applications, Third Edition, McGraw-Hill, 2005

Management Information Systems, Organization and Technology in the Kenneth C. Laudon, Jane P. Laudon, Prentice Hall International, Inc, 2000

Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, Systems Analysis and Design Methods, McGraw Hill, New York, 2004,

MODULE 3 UNIT 2: DATA PROCESSING PERSONNEL

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Data Processing Manager
- 3.2 Systems Analyst
- 3.3 Computer Programmers
- 3.4 Operation Manager
- 3.5 Database Administrator
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

The categories and number of staff that are found in a data processing center will largely depend on the size of the organization and the volume of work handled by the department. In any typical data processing center the following categories of staff will be found performing various duties:

Objectives

At the end of this unit, you should be able to:

- Identify and outline the duties of various data processing personnel

3.1 Data Processing Manager

The head of a data processing department: The duties of a data processing manager are summarized as follows:

- Interprets and executes the data processing policy as defined by management
- Controls immediate subordinate in the attainment of project objectives
- Participates in the formulation of data processing policy
- Ensures that security policies are enforced
- Coordinates all data processing operations and ensures the work flow is smooth.
- Ensures that computer operating instructions are modified when the need arises
- Assess and review the effectiveness of all data processing procedures
- Ensures that right software and hardware are purchased and used
- Monitoring test runs;
- Supervises post-implementation evaluation

- Ensures that training policies are implemented for staff
- Assesses the performance of staff for salary review and promotion
- Resolves between subordinates
- Provides leadership on data processing problems
- Develops and implements data processing standards

3.2 Systems Analyst

The duties of System Analyst are summarized as follows:

- Liaises with the user departments to ensure their requirements and problem are fully captured before systems design and implementation takes place.
- Interprets terms of reference before embarking upon systems projects
- Studies the feasibility report before embarking on a systems development project
- Supervises, Organizes and coordinates the activities of subordinate staff
- Reviewing Performance of subordinates for appropriate recommendations to the Data processing manager
- Organizes and reviews systems documentations to ensure its compliance with data processing standards
- Studies, reviews and report progress on projects to the data processing manager
- Coordinates the development and implementation alternative systems
- Reviewing performance of implemented systems and assessing the need for amendments or additional training of staff
- Discusses project proposal with subordinates such as the chief programmer

3.3 Computer Programmers

The principal duties of computer programmers are summarized as follow:

- Liaise with system analyst to determine philosophy of proposed systems and establish the type of programming language to use high level or assembly code
- Reviews systems specification to establish the details of system requirements before actual implementation(coding)
- Converts systems specification and design documents to actual products
- Develops, implements and maintains computer software
- Deploys and configures software in the specified environments
- Handles integration and interoperability associated issues
- Develops software that are users friendly
- Chooses test cases and monitor test runs
- Reviews the performances of subordinates
- Reports status of program development to systems analyst

3.4 Operation Manager

The principal duties of the operations manager are summarized as follow:

- Controls all operations activities such as data preparation, capture and control, processing of data, report generation, storing of data, preservation of backup devices etc
- Develops the operating schedule for all jobs to be run on the computer
- Ensures that data is received on time from users department
- Maintains records on equipment utilizations
- Implements standard procedures to improve the efficiency of operations
- Keeps the inventory of data processing hardware supplies such as tapes, disk, hard drive, Monitors, etc
- Maintains log of computer operations for audit trails
- Reports to superior when system problems and other operational matters arise

3.5 Database Administrator

The principal duties of database administrators are summarized as follows:

- Implements the specific policies and procedure through which data can be managed as an organizational resource.
- Plans for data usage
- Oversees the logical and physical database design and data dictionary development.
- Formulate information policy that specifies its rules for sharing, disseminating, acquiring, standardizing, classifying, and inventorying information in a organization
- Specifies rules governing the maintenance, distribution, and use of information in an organization
- Define and organizes database structure and content
- Develops security procedure to safeguard the database
- Develops database documentation
- Maintains the database management software

4.0 Conclusion

In modern days, the data processing department may be referred to as Information Technology department. Thus, the staff may be known with different nomenclature such as Information Analyst for Systems Analyst; Software Engineer for Programmer; System Support Manager for Operation Manager etc.

5.0 Summa ry

In this unit, the main duties of data processing personnel were covered. In the next unit we shall discuss the tools for creating data processing applications.

6.0 Tutor Marked Assignment

What are the duties of a systems analyst?

7.0 References/Further Readings

R G Anderson, Data Processing, Volume 1: Principles and Practice, Pitman Publishing, Singapore, 1990

MODULE 3 UNIT 3: TOOLS FOR CREATING DATA PROCESSING APPLICATION

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Machine languages:
- 3.2 Assembler Language
- 3.3 High-Level Language
- 3.3 Other Tools
- 3.4 Approaches to Programming
- 3.5 Limitation of Structured Programming
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

No matter the computer infrastructure in place for data processing - a standalone PC, a small peer-to-peer network, or a wide area network, having the right software is a major requirement for good productivity. Depending on the organization, one or more of the following software would be required for data processing:

- Accounting software
- Bank management software
- Retail point-of-sale software
- Financial planning software
- Legal and medical office Management systems
- Medical diagnostic software
- Insurance claims processing system
- Report generator Application

Software is developed by using programming languages or other software development tools. The following section explains how software programs functions and describes the processes and tool that developers use to create software

Programming is way of communicating with the computer via instructions to perform some tasks. Though hundreds of computer programming languages are now use they are broadly classified into three levels: Machine languages, assembly languages, and high-level languages.

2.0 Objectives

At the end of this unit, you should be able to:

- List some data processing software
- Describe some programming languages
- Explain the term structured programming
- Explain the term object oriented programming

3.1 Machine languages:

Machine Language: are the most basic languages for computer systems. Machine languages consist of strings of numbers defined by hardware design.. Each type of CPU has its own machine language. Thus machine language programs vary from computer to computer; that is, they are machine dependent For example, the Intel Pentium and Motorola PowerPC understand different machine languages. The instructions in Machine language are written in 0 and 1 . These binary digits, which correspond to the on and off electrical states of the computer, clearly are not convenient for people to read, write or use. Thus to provide the machine-language instructions to accomplish a simple task such as finding the sum of 2 numbers the program may look like:

```
101100101001 1001011111
111011011101 0101001011
100001010010 1010010010
```

3.2 Assembler Language

Assembly languages were developed by using English-like mnemonics for commonly used strings of machine language. Instead of writing programs as a sequence of bits, assembly language allows programmers to write programs by using abbreviation or easily remembered words instead of numbers. This is just a level above machine language. Assembly language makes use of certain mnemonics (e.g., LOAD, ADD) and assign address and storage locations automatically. We may use ADD to add a value to the content of a memory cell into a register. A program written in assembly language may look like this:

```
Load A
Add B
Store C
```

A programmer can write instructions in assembly language more quickly and much easily than in machine language. Although assembly language gives programmers some level of control, it is still costly in terms of programmer's time and effort: it is also difficult to read, debug, and learn. Assembly language is used primarily today in system software. Another drawback is that, like machine language, assembly language varies from

computer to computer. That is, it is machine dependent. Instructions written in assembly language require an assembler to be translated into machine-language equivalents.

3.3 High-Level Language

High-level languages were developed to make programming easier. These languages are called high level languages because their syntax is closer to human language than assembler or machine language code. They use familiar words instead of communicating in the machine codes which are not general and difficult to comprehend. In writing instructions in these languages, familiar arithmetic operators, such as the plus (+), or minus (-) sign that are used. This makes the reading, writing, and understanding of computer programs much easier with a high level language. Please, note that instructions in written in high-level languages must be translated to machine code equivalents before they can be executed by the computer. Common examples that can be used for application development are discussed below:

a.) FORTRAN (FORmula TRANslator)

This is one of the early languages developed in 1950s to allow programmers write codes faster than using assembly languages. It was invented primarily for the development of scientific and engineering applications. It is a programming language intended for mathematical computation and allows programmer to express numerical equations directly. The language, which enjoyed immediate and widespread acceptance, has been enhanced several times. FORTRAN has not been widely used with personal computers. Instead, FORTRAN remains a common language on mainframe systems especially those used for research and education.

A mathematical expression in FORTRAN can simply be written as:

$C = A + B$

b.) COBOL (Common Business Oriented Language)

This language was developed in 1960 by a government-appointed committee under the leadership of retired Navy Commodore and Mathematician Grace Hopper. The committee was set out to address the problem of incompatibilities among computer manufactures. The committee came up with a platform independent language called **COBOL**. **The language was intended for business data processing applications. It gained a very wide acceptance partly because of the support from the government.** A COBOL program is made up of four divisions Identification, Environment, Data and Procedure. The divisions in turn are broken down into sections, which are divided into paragraphs, which are further divided into sentences.

The Identification Division is used to provide some documentation on the name of the program, what the program does and the author (programmer) and perhaps some other helpful comments. The Environment Division provides a description of the computer on which the program will be compiled and executed. The Data Division provides information that describes what data will be processed. The actual instructions or commands that the computer should follow in accomplishing given tasks are written in the Procedure Division. COBOL is machine independent and can handle many files and records. A typical instruction that may be written in the Procedure Division is as follows:

```
ADD SCORE_IN_PAPER1 TO SCORE_PAPER_2
```

The capability of COBOL for file processing makes it suitable for developing data processing applications. However, the advent of Database Management Systems with Languages that can easily manipulate them has made COBOL unpopular for data processing in recent times. Also, COBOL is not very suitable for handling mathematical processing. A sample code in COBOL language is written below:

```
.  
.
DATA DIVISION.
FILE SECTION.
FD StudFile.
01 StudDetails.
   88 EndOfStudFile VALUE HIGH -VALUES.
   02 StudMatNo      PIC 9(7).
   02 StudName.
     03 Surname     PIC X(8).
   02 DateOfBirth.
     03 BirthYear   PIC 9(4 ).
     03 BirthMnt    PIC 9(2).
     03 BirthDay    PIC 9(2).
   02 CourseCode    PIC X(3).
   02 Gender        PIC X.
.
.

PROCEDURE DIVISION.
Begin.
  OPEN INPUT StudFile
  READ StudFile AT END
  END-READ
  PERFORM UNTIL EndOfStudFile
    DISPLAY StudId SPACE StudName SPACE CourseCode
```

```
READ StudFile
  AT END SET EndOfStudFile TO TRUE
END-READ
END-PERFORM
CLOSE StudFile
STOP RUN.
```

c.) **BASIC (Beginners All-Purpose Symbolic Instructions Code)**

This language was developed by John Kemeny and Thomas Kurtz at Dartmouth College in the mid 1960 as an easy-to-learn, interactive alternative to FORTRAN especially for beginning programmers. It was a suitable tool for teaching programming to students. It was the first high-level language to be implemented on PCs because of its simplicity. This further led to its popularity amongst developers, hobbyists and students. Several versions of BASIC language exist today among which is Visual BASIC (VB) - the most popular programming language ever created. We shall deal more extensively on VB later in this course.

3.3 OTHER TOOLS

There are a good number of tools which are much users-oriented and allow programmers to develop applications with fewer commands, although they also require more computing power. These consist of report generators, query languages and applications generators

d.) **Report Generators:**

These are also called report writers. They are programs for end-user which are used to produce reports. They allow reports to be generated directly to a printout or a screen display. The items to be generated may be the outcomes of a database query. You can specify the format such as header, footer, column, heading, etc in advance. The report generator will produce data in the specified format.

e.) **Query Language:**

A query language is a user-friendly language for retrieving information from a database management system. It serves as a database user interface, hiding the intricacies of the database from the user. The query may be expressed in the form of a sentence or near English-like statements. They can also be obtained as choices from a menu-driven application. Common examples of a query language are SQL (Structured Query Language), INTELLECT, QBE, DATARETRIEVE, PDQ, and INQUIRE

f.) **Application generator:**

Application generator is a programmer's tool that is capable of generating applications program from problem description rather than by traditional programming. With application generators parts of the codes required by a programmer are easily created. This reduces the programming time and efforts required for software development. For example, routine for capturing data, generating report and accessing database can easily be generated by the application generator. This routines can be used at some points the development of current or other applications. Where you application is unique, the application generator will provide a module that permits you to enter program code that will attend to those unique parts. Some examples of application generators are SAS, MATISS, PACBASE, NATURAL, UFO etc.

g.) CASE (Computer-aided software engineering)

CASE is an acronym for Computer-aided software engineering or Computer-aided systems engineering. It is the automation of the step-by-step methodologies for software and system development to enhance the developer's productivity. CASE tools provide automated graphics facilities for producing graphs, charts, and reports, data dictionaries, analysis and checking tools, code and documentation generators for easy software production

3.4 Approaches to Programming

It was only in from the early 1960s, that structure was imposed on how programmers wrote codes. Two popular approaches to programming are Structured Programming and Object Oriented Programming

a.) Structured Programming

Structured programming was developed in an attempt to overcome the use of GO TO statements in programming. Structured programming is a technique to make the programming process easier and more productive. A structured program doesn't depend on the GO TO statement to control the logical flow. Instead, it is built from smaller program called module, or subprograms, which are in turn made up of even smaller modules. The programmer combines module using the three basic control structures: sequence, repetition, and selection. Structured Programming evolved in the 1960 and 1970s.

Sequence Construct

This executes statements in the order in which they appear, with control passing unconditionally from one statement to the next. The program will execute statement X and proceed to statement Y. See Figure 1.

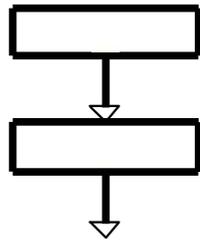
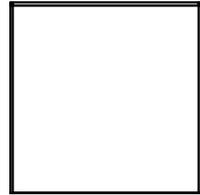


Figure 1: Sequence Construct



Selection Construct

This tests a condition and executes one of two alternative instructions based on the results of the test. A test is performed. If the result is true, Statement X is executed. If the result is false, statement Y is executed. Control then passes to the next statement (Z). See Figure 2

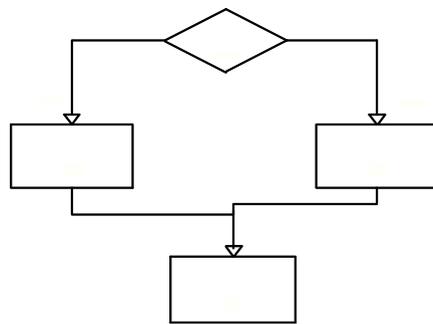


Figure 2: Selection Construct

Iteration construct

The Iteration construct repeats a segment of code as long as a conditional test remains true. A test is performed. If the result is true, statement X is executed and control returns to the test. If the result is false, control passes to the next statement.

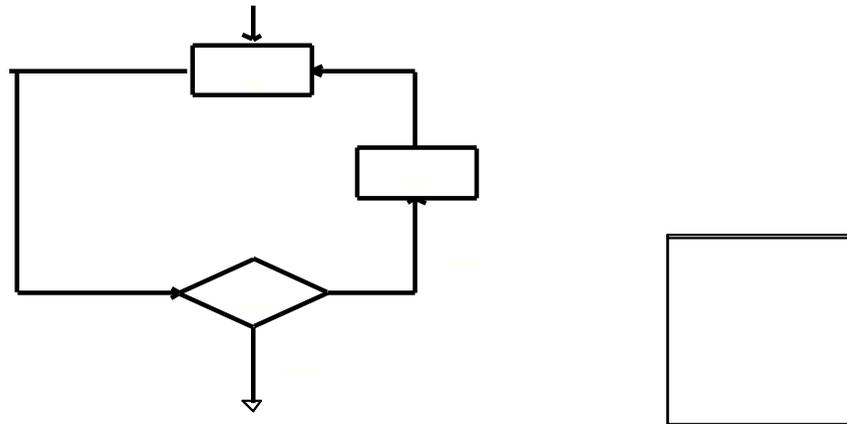


Figure 3: Iteration Construct

3.5 Limitation of Structured Programming

Software developers have found that using structured programming results in improved efficiency, but there had been some challenges with the process of building software quickly and correctly. As applications become larger and more complex, the complexity of the application increases, so does, the number of persons-years required to complete the project. One proven approach to handling this increase in complexity is object-oriented programming. Object oriented programming allows the reuse of code. Reusing code allows programs to be built quickly and correctly. OOP builds on and enhances structured programming.

b.) Object-Oriented Programming:

Object-oriented programming (OOP) is a programming approach that combines data and instructions for processing that data into a self-contained object that can be used in other programs. An object is a block of preassembled programming code that is a self-contained module. The module contains, or encapsulates, both (1) a chunk of data (2) the processing instructions that may be performed on that data.

An object can be used several times in different applications and by different programmers, thereby leading to improved productivity of the software developers. Examples of OOP languages; Smalltalk, c++, Eiffel, C#, Java, etc

4.0 Conclusion

A computer program is a collection of instructions, or statements (also called codes) carried out by the computer's CPU to perform a task. To be sure that the computer and programmers are able to understand themselves, standard language called computer languages such as FORTRAN, COBOL, ADA, etc are used. Apart from programming with standard languages simplified tools such as report generators, query languages and applications generators are also widely used for software development.

5.0 Summary

Object-oriented software development models a system as a series of reusable objects that combine both data and procedure. This leads to a reduction in the time and cost of writing software and also makes maintenance easier.

6.0 Tutor Marked Assignment

Q1. Explain the meaning of the following terms;

- a) Structured Programming
- b) Object Oriented Programming

Q2. What is the meaning of the term CASE?

Q3. State three advantages of Object-oriented Programming

7.0 References/Further Readings

Problem Solving Using C: Structured Programming Techniques, Second Edition, Yuksel Uckan, McGraw-Hill, 1999

MODULE 3 UNIT 4: VISUAL AND EVENT PROGRAMMING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Visual and Event-Driven Programming
- 3.2 Visual Basic Development Environment
- 3.3 Working with control
- 3.4 Working with Properties, Methods, and Events
- 3.5 Writing Procedure for Events
- 3.6 Using Data Types, Constants, and Variables
- 3.7 Assignment Statements and Assignment Expressions
- 3.8 Sample Program in VB
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

Visual BASIC, developed by Microsoft in the early 1990s, is one of the most popular visual programming languages. It offers a visual environment for program construction, allowing users to build various applications components using drag-and-drop tool, buttons, scroll bars, and menu. It is a very suitable language for developing user's friendly data processing applications. In this unit we shall provide an overview of the language and explain the relevant features /statements that will enable you develop good data processing applications.

2.0 Objectives

At the end of this unit, you should be able to:

- Explain the term visual program and event driven programming
- Identify the elements in the Visual Basic environment
- Explain the concept of object, properties and method
- Define the term data types and explain the terms variable and constants
- Use simple assignment statements and expression
- Describe and use mathematical operators
- Write sample program in Visual Basic

3.1 Visual and Event-Driven Programming

The increasing use of Windows environments in the 90s led to development of Visual programming languages such as Visual Basic and Visual C++ which support features for to development of Graphical User Interface (GUI) based systems. These programming languages also support event-driven programming and integrate easily with relational database systems to allow the development of complex user interface systems.

Visual programming is a method of creating programs by manipulating program elements graphically rather than by specifying them textually. The programmer makes connections between objects by drawing, pointing and clicking on diagrams.

VB is termed an event-driven programming language because with it you can write program code that responds to events that are controlled by the system or users.

Examples of event include:

- Clicking a button or menu.
- Opening or Closing a form.
- Moving the mouse over the top of an object such as a text box.
- Moving from one text box to another.

Before we proceed with learning the VB language we shall define some terms that are required for us for to have a good understanding of the language.

Object

An object is a thing like a noun in English. Examples include Forms and Controls on forms such as Buttons, Text boxes, and Icons.

Property

Objects have properties like adjectives in English. Properties describe objects. Examples of properties include Text, Name, BackColor, Font, and Size.

Method

A method is like a verb in English It is an action word that objects exhibit. Examples include methods to Show and Hide Forms and methods to Print Forms.

Event

An event occurs when users takes an action, such as clicking a Button, pressing a key, scrolling, or closing a Window. Events can also be triggered by actions of other objects.

Form

A Form is the window and dialog box where you put all the things that people interact with as they use your program. Forms are the foundations you generally use to build programs. Forms are containers for Controls.

Controls

Controls are the elements; you place inside a Form, such as Text boxes, Command buttons and List boxes. Those things you put on the form are controls, which enable the people who use your program to do things, such as enter text and click buttons.

3.2 Visual Basic Development Environment

The development environment in Visual Basic looks like the environment you have in any other Microsoft product such as Word, Excel, Power point, MS Access etc. It allows you to create and test your programs. The screen shot in Figure 13.1 shows a VB development environment.

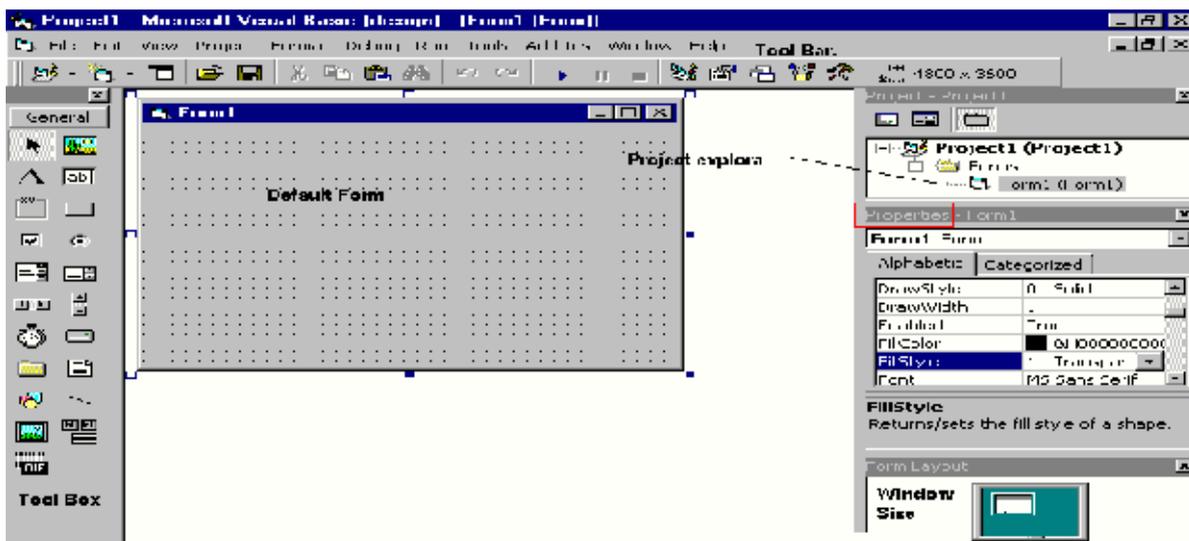


Figure 1: VB IDE

The Tool Box holds the tools you use to place control on the form. All of the controls that appear See Figure xx for detail description of the Control. You may have more or different tools in your toolbox, depending on the edition of Visual Basic you are working with.

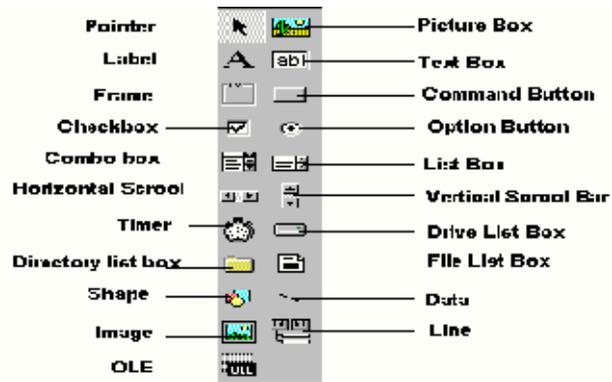


Figure 2: Tool Box

Table 1: The Visual Basic 6 Intrinsic Controls

Control	Purpose
Label	To displays text on a form
Frame	Serves as a container for other controls
CheckBox	Allows users to select or deselect an option
ComboBox	When used it allows a user to select from a list of items or add a new value
HscrollBar	Allows users to scroll horizontally through a list of data in another control
Timer	Lets your program perform actions in real time, without user interaction
DirListBox	Allows users to select a directory or folder
Shape	Used to display a shape on a form
Image	Used to display graphics (images) on a form
OLE Container	Allow you to add the functionality of another Control program to your program
PictureBox	Used to displays graphics (images) on a form and can serve as a container
TextBox	Used to display text but also enables users to enter or edit new or existing text
CommandButton	Allows users to initiate actions
OptionButton	Allows users select one choice from a group; must be used in groups of more than one
ListBox	Allows users to select from a list of items
VscrollBar	Use to scroll vertically through a list of data in another control
DriveListBox	Allows users select a disk drive
FileListBox	Allows users select a file

Line	Used to display a line on a form
Data	Used to connect a database to a program

3.3 Working with control

Two ways to add control to a form are:

- 1) By double-clicking
- 2) By drawing

Steps to draw a control on a form

- i. **Click the control's Toolbox icon.**
- ii. **Move the mouse pointer over your form. Notice that your pointer is now shaped as a crosshair instead of an arrow. Click (and hold) the mouse button where you want the control to go.**
- iii. **Drag the mouse down slightly and to the left. As you move the mouse, notice that a box starts to appear.**
- iv. **When the box is the proper size, let go of the mouse button. The control you selected now appears on the form.**

Steps to remove a control from a form

- i. **Select the control you want to delete by clicking it.**
- ii. **and Press the Delete key.**

You can also remove a control by right-clicking it. From the context menu that appears, select

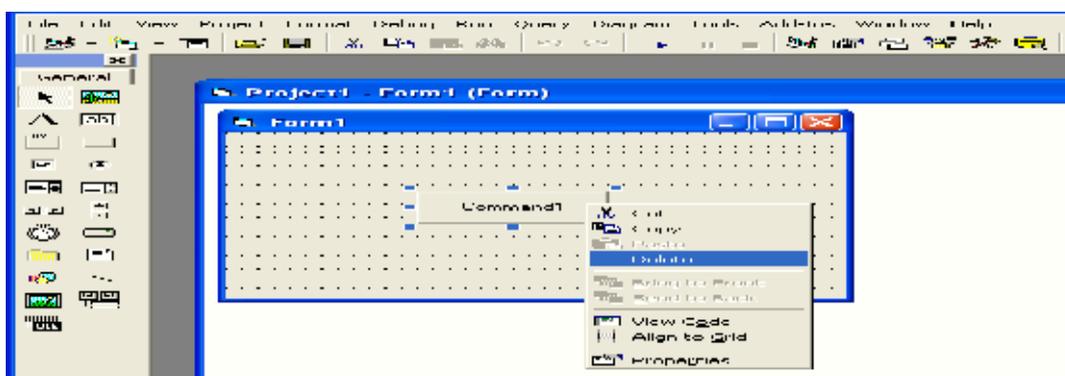


Figure 3: Deleting a Control

3.4 Working with Properties, Methods, and Events

Every object, such as a form or control, has a set of properties that describes it. The set of properties are not the same for all objects though some properties such as those listed in Table 6.1 are common to most controls. To see the properties for a given control, check the Properties window in the IDE.

Table 2: Common Properties of Visual Basic Controls

<i>Property</i>	<i>Description</i>
Left	The position of the left side of a control with respect to its container
Top	The position of the top of a control with respect to its container
Height	The height value of a control
Width	The width value of a control
Name	The string value used to refer to a control
Enabled	A Boolean (True/False) value that determines whether users can manipulate the control or not
Visible	The Boolean (True/False) value that determines whether users can see the control or not

Methods are blocks of code designed into a control that tells the control how to do things, such as move to another location on a form. Just as with properties, not all controls have the same methods, although some common methods do exist, as shown in Table 3

Table 3: Common Methods of Visual Basic Controls

<i>Method</i>	<i>Use</i>
Move	Changes an object's position in response to a code request
Drag	Handles the execution of a drag-and-drop operation by the user
SetFocus	Gives focus to the object specified in the method call
ZOrder	Determines the order in which multiple objects appear onscreen

Not all controls in VB have the same events, but some events are shared by many controls. These are represented in Table 4

Table 4: Common Events of Visual Basic Controls

<i>Event</i>	<i>Occurrence</i>
Change	The user modifies text in a combo box or text box.
Click	The user clicks the primary mouse button on an object.
DbClick	The user double-clicks the primary mouse button on an

	object.
DragDrop	The user drags an object to another location.
DragOver	The user drags an object over another control.
GotFocus	An object receives focus.
KeyDown	The user presses a keyboard key while an object has focus.
KeyPress	The user presses and releases a keyboard key while an object has focus.
KeyUp	The user releases a keyboard key while an object has focus.
Event Occurrence	
LostFocus	An object loses focus.
MouseDown	The user presses any mouse button while the mouse pointer is over an object.
MouseMove	The user moves the mouse pointer over an object.
MouseUp	The user releases any mouse button while the mouse pointer is over an object.

3.5 Writing Procedure for Events

Writing codes VB is an interesting one. Your major task is event-driven programming. VB codes are written to handle events as they occur. The codes are contained within a procedure defined as any block of code that can be called from within your application.

The syntax for a procedure is:

```
[Public|Private] [Static] Sub|Function|Property _
    functionname (arguments) [As Datatype]
{...Your procedure code...}
End Sub|Function|Property
```

An event procedure is the place in your project where you put the code that will run when an event occurs. To write an event procedure, you must access the Code window for your object by doing one of the following:

- Double-clicking the object
- Selecting the object with the mouse and pressing F7
- Selecting the object and choosing Code from the View menu
- Selecting the object's form in the Project Explorer, clicking the View Code button, and choosing the object from the Code window

3.6 Using Data Types, Constants, and Variables

Working with variable

A variable is a temporary storage location for data. Variables are used for representing data of certain types. To use a variable, you will have to declare it. By declaring a variable you are asking the compiler to allocate appropriate memory space for the variable based on its data type. The different type of variables in VB and their storage requirements are in the Table 5.

Table 5: Storage Requirements

Type	Values	Storage Size	Range
Boolean	Logical values	2 Bytes	True or False
Byte	Whole numbers	1 Byte	0 to 255
Currency (scaled integer)	Numbers with up to 15 digits left of the decimal and 4 digits right of the decimal	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Date	Date and time information	8 bytes	1 January 100 to 31 December 9999
Double	Decimal numbers (double-precision floating-point)	8 Bytes	-1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values)
Integer	Whole numbers	2 Bytes	-32,768 to 32,767
Long (long integer)	Whole numbers	4 Bytes	-2,147,483,648 to 2,147,483,647
Single	Decimal numbers	4 Bytes	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
String (fixed-length)	Text information	Length of string	1 to approximately 65,400

Variant	Any of the preceding data types	16 bytes + 1 byte per character	N/A
---------	---------------------------------	---------------------------------	-----

Comments are non executable statements that are included in a program for documentation and clarifications. When used in a program, they describe the purpose of a program, functions, or statement. They do not have any effects on the program. They are represented by REM or `/* */` in program. For examples

Naming a variable

To make your code easy to read and understood your variable names should describe their task but, to make your code easy to type, they should be short as possible. To enhance the understanding of code many programmers prefer to use some standard prefixes on their variables to indicate the type of data of the variable. See Table 6.

Table 6: Variable and Control Prefixes

Variable/Control	Prefix	Example
Boolean	bln	blnLightsOff
Currency	cur	curAmountPaid
Double	dbl	dblNairaPerGallon
Date+time	dtm	dtmDateDue
Integer	int	intQuantitySold
Long	lng	lngPopulation
String	str	strAddress
ComboBox	cbo	cboMyFavourite
Check box	chk	chkDoctorIn
Command button	cmd	cmdCalculate
Form	frm	frmMain
Frame	fra	fraTeams
Horizontal scroll bar	hsb	hsbText
Image	img	imgMain
Label	lbl	lblLastName
ListBox	lst	lstDaysOfWeek
Shape	shp	shpCircle
TextBox	txt	txtAddress

Variable Declaration

There are many ways to declare a variable in VB. The most common way is to use the Dim statement. The syntax of the Dim statement is:

```
Dim Variablename [ As Datatype]
```

The Datatype is optional and when it is not specified the default type Variant is assumed.

Examples of variable declaration statements:

```
Dim intTotalNumberSold AS Integer  
Dim curTotalAmount AS Currency  
Dim strCustomerAddress AS String  
Dim intChanging
```

Constants

There are occasions where data items in a program have values that do not change. Such values are referred to as constants. In Visual Basic, there are two main types of constant: Intrinsic and Named constant.

Intrinsic Constants

These are built-in constants or system-defined constants. They are usually stored as library files and available for use by the programmer when they are called. Common examples are color definition constants such as vbRed, vbBlue and Message statements such as vbOKOnly, vbCritical, vbInformation etc.

Named Constant:

These are programmer-defined constants.

The syntax for Constant declaration is:

```
Const variablename [As Datatype] = value
```

Const is a reserved word i.e. word with a special meaning to the Visual Basic interpreter.

```
Const strCourse AS String = "Data Processing"  
Const strDept = "Computer Science"  
Const curDiscountRate = 0.05
```

Notice that String values are enclosed in quotation marks while numeric values are not so enclosed.

3.7 Assignment Statements and Assignment Expressions

It is not enough to just declare variables, you need to be able to assign data to variables, manipulate the data and use the content of the variable. To do this, you must write some code statements. A statement is a line of code that makes a computer achieve a task. An assignment statement for example is used to assign a value to a variable. The syntax for assignment statement is as follows:

Variable = expression

Examples:

```
intNum1 = 5
frmMain.Visible = False
fAvgScore = nTotal / nNumberofPapers
```

Mathematical Operations

The primary reason for using computer in data processing is for the processing of numerical data. Mathematical operations can be used to determine the total deductions or net pay to be paid to an employee when the payroll is processed. Similarly, customers bill, interest due, account balance, and the total score of student in an examination would need some mathematical operation to be performed by the computer. Mathematical operations are performed by mathematical operator. A list of the available Mathematics operator and their corresponding VB operator are provided in Table 7.

Operation	VB Operator	Symbol
Addition	+	
Subtraction	-	
Multiplication	*	
Division	/	
Integer division	\	
Modulus	Mod	
Exponentiation		

Table 7: Math Operations

3.8 Sample Program in VB

Now that we have learnt some basics of VB, we will illustrate with a Sample Program.

Question: Develop an application in VB to calculate the Total and Average scores of students in Four papers in an Examination

Solution

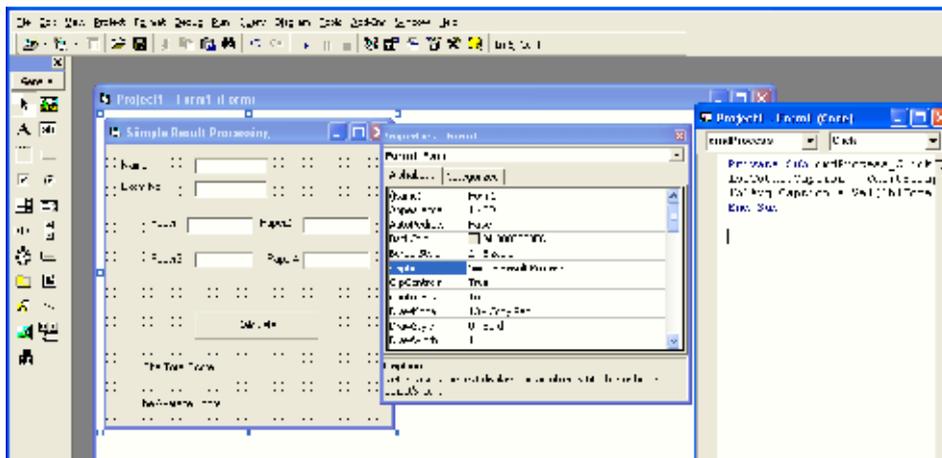
To develop this application you need go through the following steps:

1. Plan and define your users interface (Create your forms and add the appropriate controls)
2. Plan and set the properties (Assign names to your objects. See Figure 5)
3. Plan and Write the Basic codes (Write codes to handle the events. Also see Figure5)

Source Code for command (Process)

```
Private Sub cmdProcess_Click()  
lbltotal.caption = Val(txtPaper1.Text) + Val(txtPaper2.Text) + Val(txtPaper2.Text)+  
Val(txtPaper4.Text)  
lblavg.Caption = Val(lbltotal.caption) / 4  
End Sub
```

Figure 5: Users Interface Design and Codes



4. Run the program

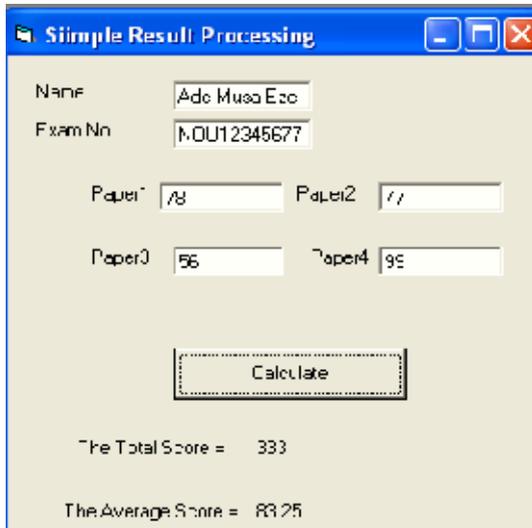


Figure 1: VB Application for Result Processing

Exercise :

Develop the same application by repeating steps 1-4

4.0 Conclusion

Visual Basic is one of the most popular visual languages and very easy to learn. However, in order to be able to practice some of the topics covered in this unit, you will need to install Visual Basic 6.0 or Visual Studio.NET Professional edition on your computer.

5.0 Summary

This unit covers the Visual Basic Development Environment, and explains the fundamental concepts of event driven and visual programming. Applications written in Visual Basic are very users friendly. The concepts learnt in this unit will be very useful in your programming career and the understanding of the remaining parts of this study material.

6.0 Tutor Marked Assignment

Q1. Develop a Visual Basic Application that allows a user to input the cost of items of three products purchased by a named customer from WAZOBIA Nigeria Ltd. When the user clicks on calculate the system would display the total and average costs of the items. Use appropriate controls to make your program users friendly. Hint: See Figure 1 of this unit.

7.0 References/Further Readings

Julia Case Bradley, Anita C. Millspaugh, Programming in Visual Basic 6.0, McGraw-Hill Higher Education, 2002

Jan L. Harrington, SQL Clearly Explained, Morgan Kaufmann Publisher, An Imprint of Elsevier Science, 2003, USA

Yuksel Uckan, Problem Solving Using C, Structured Programming Techniques McGraw-Hill International Editions, 1999

MO DULE 4 UNIT 1: CO NTRO L STRUCTURE AND STRING PROCESSING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Control Structures/Statement
- 3.2 Selection
- 3.3 Iteration
- 3.4 String Processing
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

A piece of Visual BASIC code executes from the first line of code and proceed line by line until the end of the statement is reached if no control statements are used. Control statements are used to transfer control to particular section of program. They may also be use when a particular segment of a program needs to be repeated. This unit describes the various control statements, and operators that are required in a control structure. Also string manipulation covered in this unit.

2.0 Objectives

At the end of this unit, you should be able to:

- Describes control structure
- Describes relational and Logical Operators
- Use block statements
- String processing

3.1 Control Structure

Control statements are used to introduce some branch or iteration (repetition) into our codes. Controls can be implemented in our programs by using any of the three fundamental Control Constructs :

- Sequence : statements are executed in sequence
- Selection: one instruction or group of instructions is executed or another
- Iteration: statement or statements are repeated

Before we consider the various control structure in VB we shall examine the relational and logical operators. Relational and logical operators are often used to implement selection and Iteration. Conditional expressions are formed using relation and logical operators.

Relational Operators

Relational operators are used form statements that allow values to be compared. Every relational statement evaluates to True or False. Study Table ..

Table 13: Relational Operators

Symbol	Meaning	Example	Result
=	Equal		3=4 False
>	Greater Than	3>4	False
<	Less than	3<4	True
>=	Greater than or equal to	3>=3	True
<=	Less than or equal to	2<=2	True
<>	Not equal to	2<>3	True

Logical Operator

In writing your program, you may want to test two or more relations. For example, you may want to test the that the age of a student is more than 15 years and his examination score is greater than 260 before he can be considered for admission into the university. Logical operators are used to combine the two relations. The VB logical operators are:

Operator	Meaning	Examples	Result
Not	Provides the logical negative of the input	Not (3>2)	False
And	Combines two expressions. The resulting value is true only when the two expressions evaluate to True	(8>5) And (5>2) (6<5) And (3>2)	True False
Or	Combines two expressions. If either of the expressions evaluate to true, the result will be true	(6<5) Or (3>2) (8>5) Or (5>2)	True True
Xor	Combines two expressions. If only one is true, it evaluate to true otherwise it evaluates to false	(6<5) Xor (3>2) (8>5) Xor (3>2)	True False

Notice that logical expression like relation expression (statement) must always evaluate to True or False.

The VB expression for the combined relation for the student admission into the university can be expressed as

```
intAge > 15 And intScore > 260
```

Where intAge and intScore would be been earlier declared.

3.2 Selection

i) If Statement

VB language has two statements for implementing the selection control: if and Case. For many decisions you may want to execute a statement (or group of statement) if a particular condition is true. Two forms of the if statement handle True conditions: The single-line if statement and the multiple-line if statement. Each uses the if statement to check a condition and execute the corresponding code when the condition is true. See the examples

Single line If

```
If a > 10 Then Label1.Caption = "A is greater than 10"
```

Multi-line If

```
If a > 10 Then  
    Label1.Caption = "A is greater than 10"  
    Label2.Caption = "B is less than 20"  
End If
```

Handling True and False Conditions

The else statement is used with the If statement to handle false condition.

If and Else

```
If a > 10 and b < 20 Then  
    Label1.Caption = A is greater than 10  
    Label2.Caption = B is less than 20  
Else
```

```
Label1.Caption = means than A is less than or equal to 10 or
Label2.Caption = B is greater than or equal to 20
End If
```

Please note the use of the AND operator to combine the two expression (a>10) and (b<20). An If...Then...Else must always conclude with End if. The word Then must appear on the same line as the if with nothing following Then (expect a remark). Else and End if (if used) must appear alone on a line

ii.) Select Case Statement

The Select Case structure is similar to a series of If/Then/ElseIf statements. Thus program decisions that can be code with a Case structure can also be coded with nested if statement, but usually the Case structure is simpler and clearer. The Select Case structure allows you to conditionally execute a series of statement groups based on the value of a test expression, which can be a single variable or an arithmetic expression.

The syntax of the select Case Statement is:

```
Select case expression
  Case constant list
  [statement(s)]
  [Case constant list
  [statement (s)]]
.
.
[Case Else]
[Statement(s)]
End Select
```

The expression in a Case structure is usually a variable or property you wish to test. The constant list is the value that you want to match; it may be numeric or string constant or variable, a range of value, a relational condition, or a combination of these.

There is no limit to the number of statement that can follow a Case statement.

Example 1

```
Select Case intExamscore
```

```
Case 70 To 100
    lblMessage.Caption = A
Case 60 To 69
    lblMessage.Caption = B
Case 50 To 59
    lblMessage.Caption = C
Case 40 To 49
    lblMessage.Caption = D
Case Else
    lblMessage.Caption = F
End Select
```

Example 2

```
Select Case strCurrentMonth
    Case 1
        strMonthName = January
    Case 2
        strMonthName = February
    ...
    Case 12
        strMonthName = December
End Select
```

End Select statement defines the end of the Case block. To trap any values not picked up by earlier cases use the Case Else construct:

```
Case 12
    strMonthName = December
Case Else
    MsgBox Cannot be a Month, 12 Month in a year
End Select
```

The elements used for the constant list may have any of the forms. The following example is used to test for a string

Example 3

```
Select Case txt.SchoolName.Text
```

Case Science_and_Technology

(Code for school of science and technology)

Case Education

(Code for school of Education)

Case Humanities

(Code for school of Humanities)

Case Law

(Code for school of law)

Case Else

(code for others)

End Select

3.3 Iteration

i.) Do Loop statement

In a program, you may want the computer to execute a group of instructions many times until a certain task is accomplished. The process of repeating a series of instructions is called looping. The group of repeated instructions is called a loop. Iteration is a single execution of the statement(s) in the loop. The repetition is controlled by a condition that is associated with the repetition statement.

There are two types of repetition statement

- a.) Pretest repetition statement
- b) Posttest repetition statement

A pretest repetition statement computes a value for the condition expression before entering the loop. The loop may never be executed if the terminating condition is true the first time it is tested.

A posttest repetition statement first executes the loop body and then computes a value for the condition expression. If the value satisfies a true condition, the loop body is executed again; otherwise the repetition is terminated. The statements in the loop will always be executed at least once.

A DO/Loop terminates based on a condition that you specify. Execution of a DO/Loop continues while a condition is True or until a condition is True. The DO/loop command is used in two forms:

(Syntax):

```
Do [{While | Until} condition]
    [statements]
Loop
```

Example:

```
month = 1
Do While month <= 12
    month = month + 1
Loop
```

Do while tests the loop condition at the beginning of the loop. With this type of loop, the statements inside the loop may never be executed if the condition for executing them is not satisfied.

ii.) Loop While Statement

This tests the completion at the bottom of the loop, which means that the statement within the loop will be executed at least once.

(Syntax):

Do

 [statements]

Loop [{ While | Until } condition]

Example:

```
intTemp = 0
```

```
Do
```

```
    intTemp = intTemp + 1
```

```
Loop While intTemp < 10
```

iii.) For Loop Statement

This construct uses the FOR..NEXT and a counter index to repeat the statement(s) in a loop for a given number of times. The value of the counter variable is tested to determine, the number of times the statements within the loop should be executed. A counter-controlled loop generally has three elements:

1. Initialize the counter
- 2 Increment/Decrement the counter
- 3 Test the counter to determine when it is time to terminate the loop

The syntax for the For loop is:

```
For counter = start To end [Step increment]
```

```
    Statements
```

```
Next [counter]
```

Example 1

```
curTotal = 0
```

```
For intConter = 1 To 10
    curTotal = curTotal + intCounter
Next intCounter
```

Example 2

```
curTotal = 0
For intConter = 10 To 1 step -1
    curTotal = curTotal + intCounter
Next intCounter
```

Note the use of step in this example to indicate a decrement. When it is not used an increment is assumed. as in the example 2

3.4 String Processing

Apart from manipulating numbers, VB has features that make string processing easy for a programmer. You may need to know how many characters are in a string; convert the case of a string data; work with part of a string. VB provides some functions that will help you to achieve your desired task. A Visual Basic function performs an action and returns a value. The expressions named in parentheses are called arguments.

The Syntax of the Left function:

i.) **Left(String Expression, noofcharacterstoextract)**

This is used to extract the leftmost characters of a string up to a specified position. The String expression may be a string variable, string literal, or text property.

Example

```
Dim strCoursetitle As String
strCoursetitle= Data Processing
```

```
Left (strCoursetitle, 3)    this expression will return the string Dat
Left (strCoursetitle, 12)  this expression will return the string Data Process
Left (strCoursetitle, 4)   this expression will return the string Data
```

ii.) **Right (StringExpression, noofcharacterstoextract)**

This is used to extract the rightmost characters of a string up to a specified position. The String expression may be a string variable, string literal, or text property.

Example

```
Dim strCoursetitle As String  
strCoursetitle= Data Processing
```

Right (strCoursetitle, 3) this expression will return the string ing
Right (strCoursetitle, 12) this expression will return the string a Processing
Right (strCoursetitle, 4) this expression will return the string sing

iii.) Mid (StringExpression, Startposition, [noofcharacterstoextract])

This is used to extract the rightmost characters of a string up to a specified position. The String expression may be a string variable, string literal, or text property.

Example

```
Dim strCoursetitle As String  
strCoursetitle= Data Processing
```

Mid (strCoursetitle, 3) this expression will return the string a Processing
Mid (strCoursetitle, 7) this expression will return the string ocessing
Mid (strCoursetitle, 6,12) this expression will return the string Processing

This is used to determine the length of a string expression
Syntax

iv.) Len(String Expression)

Example

Len (strCoursetitle). This expression will return the value 15

Note that spaces are also counted as part of the characters

4.0 Conclusion

The ability of computers to process data such as numbers and strings make them very suitable tool for data processing. For you to be able to practice programming in VB you must have a co

5.0 Summary

As your lines of codes continue to increase, you certainly would need to execute or repeat the execution of a section of the codes before proceeding to another. The control statements covered in this unit help you to accomplish just what you need.

6.0 Tutor Marked Assignment

Q1. Develop an interactive application in VB that allows a user to input the scores of students in an examination while the system displays the student s grade. Assume that: Scores 70-100 = A , Score 60 – 69 = B , Score 50-59 = C , Score 40-49= D , Score 0-48 = F

7.0 References/Further Readings

Julia Case Bradley, Anita C. Millspaugh, Programming in Visual Basic 6.0, McGraw-Hill Higher Education, 2002

Jan L. Harrington, SQL Clearly Explained, Morgan Kaufmann Publisher, An Imprint of Elsevier Science, 2003, USA

Yuksel Uckan, Problem Solving Using C, Structured Programming Techniques McGraw-Hill International Editions, 1999

MODULE 4 UNIT 2: ARRAYS AND PRINTING

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Working with Arrays
- 3.2 Declaration of Array
- 3.3 Looping through the Elements of an Array
- 3.5 The MsgBox Statement
- 3.7 Sending Information to the Printer
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introduction

We can use arrays to introduce some efficiency to the use of variables names. An array is a series of variables with the same name. Sometimes array are referred to as table, or subscripted variable. The individual value is referred to as elements, and each element is accessed by its subscript, which is a position number.

2.0 Objectives

At the end of this unit, you should be able to:

- Describe and explain the meaning of array
- Types of array
- Describe and use the MsgBox statement
- Explain how to print in VB

3.1 Working with Arrays

Consider an array of Student name:

strStudentName Array

[0]	Adebayo
[1]	Ayodele
[2]	Musa
[3]	Jeremiah
[4]	Okereke
[5]	Anthony

[6]	Isaac
[7]	Ronke
[8]	Osarobo
[9]	Yobo

Table 1: Student Array.

When you use a control array, the individual controls are referenced as strStudentName (0), strStudentName (1), strStudentName (2), etc. Each individual variable is called an element of the array. The subscript inside the parenthesis is the position of the element within the array. For example, strStudentname (2) = Musa while strStudentname (7) = Ronke

3.2 Declaration of Array

When array are declared, the number of elements in the array must be specified so that the computer system will be able to reserve the required storage for the processing of the array.

For example the following statements allocate storage for the arrays specified.

```
Dim strStudentName (0 to 9) As String
Dim curPrice(10) As Currency
Dim intScore As Integer
```

The dimension of an array may have an empty parenthesis as in the following statement

```
Public strSchool ( ) as String
```

The array declared above without a specification of its dimension is a dynamic array because the number of elements may change during the execution of the program. A static array is one with a fixed size. Any attempt to change the size while the program is running would result in an error message.

3.3 Looping Through the Elements of an Array

To loop through the element of an array you use the For/Next or FOR Each and Next statement. The For / Next works well to traverse the elements in the array. With the For Each/Next you wouldn't need to manipulate the subscript of the array. The general syntax of the For/ Each Next

Statement is:
For Each ElementName In ArrayName
Statements(s) in loop
Next [ElementName]

The statement above allows a Visual Basic program to automatically loop through each element of the array, assign value to ElementName, and makes one pass through the loop. If the array has say, 10 elements, the loop will execute 10 times. ElementName must be declared as type Variant.

Example

```
Dim strStudentName (0 to 9) As String  
Dim vnt1Nme As Variant  
For Each vnt1Nme in strStudentName  
Printer.Print vnt1Nme  
Next vnt1Nme  
(statement(s))
```

When the loop has finished execution depending on the numbers of elements in the array, the statement(s) following the Next statement will be executed. However, an EXIT For statement may be used within the loop to exit early.

3.4 Multidimensional Arrays

A one-dimensional array is a one-row collection of variables. In Visual Basic, however, you can create arrays that have up to 60 dimensions. Usually, two-dimensional arrays will suffice for most introductory programming projects, and you won't likely need to work with more than three dimensions.

A two-dimensional array is made up of columns and rows that intersect to form a grid. Each grid cell has a location defined as ColumnNumber, RowNumber. Notice that each element is defined by the coordinates of the column position and the row position. For example, the array element intScores (0, 0) is 59 and the element intScores (2,2) is 79.

59	69	56	
70	45	78	
23	77	79	
45	67	56	

Table 2: Two Dimensional Arrays

To define a two dimensional array table, the Dim statement specifies the number of rows and columns in the array. The row is horizontal and the column is vertical. The following table has four rows and three columns

To create a two-dimensional array, use the following syntax:

```
Dim|Public|Private ArrayName(SubscriptOfCols, _  
                               SubscriptOfRows) As DataType
```

In this syntax,

- Dim, Public, and Private are Visual Basic keywords that declare the array and its scope. If you use Dim, the array is private to the procedure in which it is declared. Public makes the array visible from anywhere in the program, and Private makes the array visible only to the form or module in which it is declared. Using Dim within a module automatically makes the array available anywhere in the program, as if the Public keyword were used.
- ArrayName is the name of the array.
- SubscriptOfCols is the number of the highest column in the array.
- SubscriptOfRows is the number of the highest row in the array.
- DataT ype is any valid Visual Basic data type.

Therefore, to declare the array, you would use this syntax we will have:

```
Dim intScores (1 To 4, 1 To 3) as Integer
```

Whereas you can consider a two-dimensional array to be a rectangle, you can consider a three-dimensional array to be a rectangular block.

3.5 The MsgBox Statement

The message box is a simple form that displays a message and at least one command button. You can display a message, an optional icon, a title bar caption, and command button(s) in a message box

The MsgBox Statement

```
MsgBox Message string [, Buttons/icon][, Caption of title bar ]
```

The message string is the message you want to appear in the message box. The button portion is optional; it determines the command buttons that will display and any icons that will appear. If you omit the Caption of title bar, the project name will appear in the message box title bar.

E.g

```
If txtAddress.Text = "" Then
    MsgBox "Please Enter your address. ", vbOKOnly, "Address required"
Endif
```

3.6 Retrieving a Value from the MsgBox() Function

The MsgBox statement works fine for informing users of problems or prompting them to take an action. If you need to obtain a decision from users, however, you must return a value by using the MsgBox() function.

- OK--Displays a single button with the caption OK.
- OK, Cancel--Displays two buttons in the message box, letting users choose between accepting the message and requesting a cancellation of the operation.
- Abort, Retry , Ignore--Displays three buttons, usually along with an error message.
- Yes, No, Cancel--Displays three buttons, typically with a question.
- Yes, No--Displays two buttons for a simple yes-or-no choice.
- Retry, Cancel--Displays the two buttons that allow users to retry the operation or cancel it.

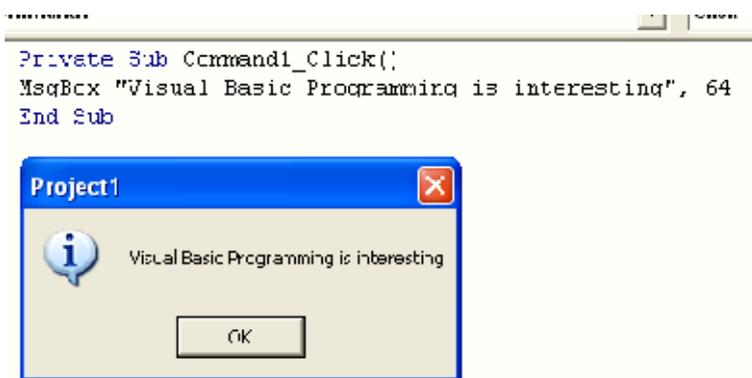


Figure 1: Show a VB source code and the generated message box

3.7 Sending Information to the Printer

When you print using PrintForm all output is sent as a graphics. The Printer.Print is used to set up output to a printer. When the statement Printer.Print is encountered, your desired output will be added to the Printer object. VB only sends the contents of the Printer object when it encounters an EndDoc or NewPage method.

i.) Formatting Line

To enhance the presentation of your output there is the need to format your object texts or objects.

The comma, semicolon, Tab function and Spc functions are used to format text in VB

The print page takes five columns per line. Each of the columns is referred to as a print zone. A comma is used to advance the position of printed output to the next print zone within a page.

Example

Printer.Print , Processing

This prints Processing in the next print zone.

Printer.Print Data , Processing

Prints Data in print zone 1 and Processing is placed in print zone 2

The output will look like

Data Processing

ii.) Semicolon:

Semicolon is used if you need to separate printed items without advancing to the next print zone.

Printer.Print Department: ; Computer Science

The output will look like

Department: Computer Science

Compare this output with the one generated with the use of a comma

Printer.Print Department: , Computer Science

The output will look like
Department: Computer Science

Please note that if the last character of a line of output is a semicolon or a comma, the next Print command will continue on the same line without moving to the next line provided the printer has not reach the last print zone for that page.

For example:

```
Printer.Print Department: ;  
Printer.Print Computer Science
```

Will output as

Department: Computer Science

To print a blank line use the Printer method without any items to print

Example:

```
Printer.Print
```

Thus the statements

```
Printer.Print Department: ;  
Printer.Print Computer Science  
Printer.Print  
Printer.Print Level: ;  
Printer.Print 300
```

Will output as

Department: Computer Science

Level: 300

**iii.) The Tab Function
Syntax**

Tab (Columnposition) : This is used to place a the printing of an item in as specified position on a printed line. The column position specified in the Tab function is a absolute one that is, position 1 1 means column 1 while position 30 means column 30 on the line

Example

```
Printer.print Tab (30) School of Science and Technology  
Printer.Print Tab (10) Department: ; Tab (22) Computer Science
```

Will output as

```
                School of Science and Technology  
Department: Computer Science
```

Notice that since column position in a tab function is relative, the Tab function for the Computer science print item could only start from position 22 to create room for the Department: starting from column position 10.

The Spc Function

Syntax

Spc(Noofspacesadvance)

This function is used to advance a number of spaces from the last item in a printed line

Example

```
Printer.Print Department: ; Spc(5) Computer Science
```

Will output as

```
Department: Computer Science
```

iv.) The NewPage and EndDoc Statement

The NewPage is used to send the current page to the printer and clears the printers object in memory so that a fresh page can begin. The EndDoc method sends tha current page to the printers and terminates the current printing process.

Example

```
Private Sub PrintDetails()
```

```
Printer.Print "Department", Level
Printer.Print "Mathematics , 300
Printer.Print "Economics , 400
Printer.Print "Accounting , 100
Printer.Print "Nursing , 400
Printer.EndDoc 'This will tell the printer it has finished.
```

End Sub

Everything that appears in position will get printed by the default printer The printer enddoc is used to tell the printer the job is finished.

Will output as

```
Department    Level
Mathematics   300
Economics     400
Accounting    100
Nursing       400
```

4.0 Conclusion

As programming become more complex, it may be necessary to work with variables that hold more than one value. In the case, arrays can be used to streamline your code by allowing you create shorter and more efficient variables.

5.0 Summa ry

In this unit we covered arrays and described how to display messages using the MessageBox command, and how to send documents to a printer. Printed copies of documents are also referred to as hardcopies.

6.0 Tutor Marked Assignment

Q1. Write a VB program sent your name, address and other personal details to a printer?

7.0 References/Further Readings

Julia Case Bradley, Anita C. Millspaugh, Programming in Visual Basic 6.0, McGraw-Hill Higher Education, 2002

Jan L. Harrington, SQL Clearly Explained, Morgan Kaufmann Publisher, An Imprint of Elsevier Science, 2003, USA

Yuksel Uckan, Problem Solving Using C, Structured Programming Techniques McGraw-Hill International Editions, 1999

MO DULE 4 UNIT 3: FILE PRO CESSING

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Working with Files
- 3.2 Working with Sequential Files
- 3.3 Working with Random Files
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References/Further Readings

1.0 Introductio n

A data file is a named collection of related records normally kept in external storage. Data stored in files are permanent until you decide to delete them. Data files once they have been created can be used as input for the same program for as many program run as you want without having to retype data. Data files facilitate the sharing of data by many applications. Each record is broken down into fields. For example, in the Student registration file, information on Matriculation Number, Name, School, Department, and Level, make up a record.

Matriculation Number	Name	School	Department	Level	

Table 1: Students record

2.0 Objectives

At the end of this unit, you should be able to:

- Explain why data are stored in files
- Use various VB statements to implement file processing
- Provide sample codes in VB to demonstrate file processing

3.1 Working with Files

This refers to the manner in which data are organized, stored and retrieved. Two major ways of file organization are sequential and random

The three main steps used for processing files are as follow:

1. Open the file: A file must be open for data to be read or written to it.
2. Read or Write records to a file
3. Close the file

i.) The Open Statement

The Open statement makes a file ready for use. The syntax of the Open statement is

Open Filename For mode As #Filenumber [Len = Recordlength]

- Filename is the name of the file to be open and may include the drive and the path if not in the current directory
- The mode is used to indicate the way that the file is accessed and could be INPUT, OUTPUT, APPEND or RANDOM

File Mode	Description
Input	This is used to read data already stored in a disk
Output	Used when data are output from a program and written on the disk. New data are written at the beginning of the file, overwriting any existing data.
Append	Used when data are to be added to the end of an existing file.
Random	Allows input and output of data and records to be accessed in any order.

Table 2: File modes

- Filenumber is a number associated with the opened file. It value can be from 1 to 511.
- The Recordlength is the number of characters in a record and can be up to 32,767 characters. This is usually associated with Random files.

Examples of Open Statement

Open Studrec.dat For Input As #1

This opens a file named Studrec as an Input file in the current directory calling it file #1

Open A:\library.dat For Output As #2

This opens a file named library.dat as an Output file in drive A calling it file #2

Open A:\Inventory.dat For Random As #3 len = 30

This opens a file named Inventory.dat as a Random file in drive A calling it file #3 with record length of 30.

ii.) Close Statement

The close statement is used to close the opened file(s) and terminates the processing of a disk file. The syntax of the close statement is

Close #Filenumbr

Filenumbr is the number associated with file opened.

Examples

a) Close #1

This closes file #1

b) Close #1, #2, #3

This Closes file #1, #2 and #3

c) Close

This closes all opened files

iii.) Using FreeFile Function

For a small project it may be easy for you to assign file number 1, 2 3 as so on to your files as they occur without conflict. However, for large projects this may not be feasible in which case you can use the Freefile function to assign the next available file number to your file as they occur.

3.2 Working with Sequential Files

Sequential files contains data elements stored one after the other according to a particular order. Data are read from a sequential device in the same sequence that they were written. To locate a record in a sequential file you must start from the beginning, and move from one record to the other until the record is found or you get to the end of the file.

i.) Reading Data in a Sequential File

To read data in a sequential file, the file must be opened first in a Input mode. Then use the input statement. The Syntax for the Input Statement is:

Input #Filenumber, listoffields

Example

Input #1, lblMatricno.Caption, lblName.Caption, lblSchool.Caption,
lblDepartment.Caption, lblLevel.Caption

Input #2, intScore1, intScore2, intTotalscore

ii.) The EOF Statement

This statement is used to mark the end of file. The general Syntax is:
EOF (filenumber)

iii.) Writing Data to a Sequential File

Before data can be placed in a sequential file it must be opened first in either Output or Append mode. The write statement is used to place data in a sequential file. The general syntax of the Write statement is

Write #Filenumber, listoffields

Examples

Write #1, txtMatricno.Text, txtName.Text, txtSchool.Text, txtDepartment.Text,
txtLevel.Text

Write #2, intScore1; intScore2; intTotalscore

3.3 Working with Random Files

Random files allow data to be read in any order unlike a sequential file. Random files allow data to be accessed directly without starting from the beginning and moving from one record to the other until the data is found or one gets to the end of the file.

Before a record can be used as a random file you need to set up its structure with a Type statement and then dimension a record variable of the data type. The following code uses the Type statement to declare Student record type:

```
Private Type Student
    Matricnum As String *12
    Name As String *20
    School As String * 20
    Dept As String * 20
    Level As Integer
End Type
```

To access a custom record type use the dot notation. This is similar to accessing object properties

```
Dim Stud1 As Student
Stud1.Matricnum = NOU050010456
Stud1.Name = Sonekan Tunde
Stud1.School = Social Sciences
Stud1.Dept= Accounting
Stud1.Level =400
```

i.) Reading and Writing a Random file

The Put and Get statement are used to access random files.

ii.) Using the Put Statement

The put statement is used to add or store record into a random file. The syntax of the put statement is as follow

```
Put filenumber, [recnumber], listoffields.
Put #2, intScore1, intScore2
Examples
```

iii.) Using the Get Statement

The get statement is used to retrieve record from a random file. The syntax of the get statement is as follow

```
Get filenumber, [recnumber], listoffields.
```

```
Examples
Get #1, intScore1, intScore2, intTotalscore
```

iv.) Using the Seek statement

The seek statement is used to move from record to record. The seek statement has two parameters which are the file number and the record number.

Example

Seek #1, 5

This causes the next Put or Get to access record number 5.

4.0 Conclusion

One of the features of a computer that makes it suitable for data processing is its ability to store data either temporarily or permanently. Data files are used to store information more permanently in secondary storage devices. Data files once they have been created can be used as input for the same program for as many program run as you want without having to retype data. Data files are very essential for data processing

5.0 Summary

Data processing applications such as inventory, payroll, accounting, budgeting, and student's information systems require that data are stored permanently in a file. In this unit we explained the meaning of a file and covered how Visual Basic language is used to implement the processing of files.

6.0 Tutor Marked Assignments

- Q1. What are the differences between a sequential file and a random file?
- Q2. What is the purpose of the EOF statement?

7.0 References/Further Readings

Julia Case Bradley, Anita C. Millspaugh, Programming in Visual Basic 6.0, McGraw-Hill Higher Education, 2002

Jan L. Harrington, SQL Clearly Explained, Morgan Kaufmann Publisher, An Imprint of Elsevier Science, 2003, USA

MODULE 4 UNIT 4: STRUCTURED QUERY LANGUAGE (SQL)

Table of Contents

- 1.0 Introduction
- 2.0 Objectives
- 3.1 Structure of a SQL Statement
- 3.2 Using SQL Statements
- 3.3 Using the SELECT Statement:
- 3.4 Using the CREATE Command
- 3.5 Using the INSERT Statement
- 3.6 Using the UPDATE Statement
- 3.7 Using the DELETE Statement
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignments
- 7.0 References/Further Readings

1.0 Introduction:

There are many relational DBMSs. All relational database management systems speak a common language called SQL (Structured Query Language). The aim of this language is to store, manipulate, and retrieve data stored in relational databases. SQL combines the familiar database concept of tables, rows (records), and columns (field) and the mathematical idea of a set. Typical SQL queries define the fields to be processed, the table containing the fields, the range of records to be included, and, for records retrieval, the order in which the returned records are presented.

2.0 Objectives

At the end of this unit, you should be able to:

- Explain the meaning and use of SQL
- Give examples of basic SQL statement
- Demonstrate the use of SQL statements with examples

3.1 Structure of a SQL Statement

A SQL statement consists of the following three main components:

- i. The SQL operation
- ii. The target
- iii. The Condition

i) The SQL Operation

The four basic operations performed by a SQL Data Manipulation Language statement are:

SELECT, INSERT, UPDATE and DELETE.

ii) The Target

All SQL DML statement operates on at least one database tables or views. The essence of the target component is to identify those tables or views. In specifying out the target component you may use the FROM clause

iii) The Condition

The condition component is specified by the WHERE clause and identifies the specific rows to be operated on by SELECT, UPDATE, or DELETE statement.

3.2 Using SQL Statements

In the following sections we shall cover five basic SQL statements

- i. SELECT
- ii. INSERT
- iii. CREATE
- iv. UPDATE
- v. DELETE

3.3 Using the SELECT Statement:

This is probably the most common and widely used of all SQL statements. It is used to retrieve data from a database. The retrieval of data may be done from one or more tables or views. A select statement always has a target component specified in by the FROM clause and often has a condition statement specified by the WHERE clause. Study the following uses of the SELECT statements.

Syntax

```
SELECT "column_name" FROM "table_name"
```

To illustrate with the above syntax, assume that we have the following table:

Table: Student_ Information

Matric_No	Name	Department	Level	
NOU0400010311	Adamu	Computer	Science	400
NOU0500011211	James	Economics	300	
NOU0600012311	Peter	Mathematics	200	
NOU0700034435	Musa	Nursing	100	
NOU0700042341	Sule	Computer	100	

		Science	
NOU0600051332	Adebayo	Accounting	200

Also, let us assume that we are interested in the matriculation number of all the students in the Student_Information table: To accomplish this, we use the command:

a.) SELECT Matric_No FROM Student_Information

The result of the selection will be:

NOU0400010311
 NOU0500011211
 NOU0600012311
 NOU0700034435
 NOU0700042341
 NOU0600051332

Similarly to select all the departments from the student_Information table you can use the command:

b.) SELECT Department FROM Student_Information

The result of the selection will be:

Computer Science
 Economics
 Mathematics
 Nursing
 Computer Science
 Accounting

Notice that the result of the SELECT statement in ii.) produced some redundancies i.e., the department Computer Science appears twice. To avoid this, you may wish to apply the command DISTINCT after SELECT. That is to select all distinct departments in table Student_Information write

c.) SELECT DISTINCT "department " FROM "Student_Information"

The result of the selection will be:

Computer Science
 Economics
 Mathematics

Nursing
Accounting

Next, we might want to conditionally select the data from a table. For example, we may want to only retrieve student names with Level above 200. To do this, we use the WHERE clause. To select all Names with Level above 200 in the Student_Information Table, we apply the commands

d.) SELECT Name FROM Student_Information WHERE Level > 200

The result of the selection will be:

Adamu
James

So far, we have seen how to retrieve relevant data from a table using SELECT and WHERE commands. Sometimes, however, you may want the result of your query to be in a particular order. This could be in ascending order, in descending order, or could be based on either numerical value or text value. To achieve these, you can apply the ORDER BY clause to achieve your goal. The syntax for an ORDER BY statement is as follows:

```
SELECT "column_name"  
FROM "table_name"  
[WHERE "condition"]  
ORDER BY "column_name" [ASC, DESC]
```

In the syntax written above:

[] means that the WHERE statement is optional. When it is used it appears before the ORDER BY command.

ASC means Ascending

DESC means Descending

When neither ASC nor DESC is specified, ASC is taken as the default. It is possible to order by more than one column. In this case, the ORDER BY clause above is written as:

```
ORDER BY "column_name1" [ASC, DESC], "column_name2" [ASC, DESC]
```

For example, we may wish to list the contents of the Table, Student_ Information by Name, in ascending order: We use the command

e) SELECT Name, department, level FROM Student_Information ORDER BY Level, Name

Result

Adamu	Computer Science	400
Adebayo	Accounting	200
James	Economics	300
Musa	Nursing	100
Peter	Mathematics	200
Sule	Computer Science	100

f.) Using the COUNT Command

The Count command is an arithmetic function that allows you to count up the number of row in a certain table.

The syntax is: `SELECT COUNT ("column_name") FROM "table_name"`

For example, if we want to find the number of departments in the table we use the command

SELECT COUNT (Department) FROM Student_ Information

The result of the selection will be:

Count(Department)
6

g.) Combining the COUNT and DISTINCT Commands

COUNT and DISTINCT can be used together in a statement to fetch the number of distinct entries in a table. For example, if we want to find out the number of distinct department, we type in the following

```
SELECT COUNT (DISTINCT Department) FROM Student_Information
```

The result of the selection will be:

```
Count(DISTINCT Department)
5
```

3.4 Using the CREATE Command

Tables are the basic structure where data is stored in the database. When developing data processing applications, the programmer or Data processing manager must first create database and then tables within the database. Though many database tools allow you to create tables without writing SQL, it is important to note that SQL uses the CREATE TABLE command for creating table.

A table as discussed earlier is divided into rows and columns. Each row represents one piece of data, and each column can be thought of as representing a component of that piece of data. So, for example, if we have a table for recording student information, then the columns may include information such as Matriculation Number, Name, School, Department and Level. As a result, when we specify or create a table, we include the column headers and the data types for that particular column.

The SQL syntax for CREATE TABLE is:

```
CREATE TABLE "table_name"
("column 1" "data_type_for_column_1",
"column 2" "data_type_for_column_2",
... )
```

Thus if we are to create the Student_information table used in the preceding examples, we type

```
CREATE TABLE Student_information (
MatricNo varchar(12),
Names varchar(50),
Dept varchar(50),
Level integer )
```

Now that you know how to write simple queries to retrieve information from a table and the command for creating a table, we shall look at the INSERT, UPDATE and DELETE command.

3.5 Using the INSERT Statement

The INSERT statement is used to create new rows of data in a target database table or view. The statement provides a list of columns that will receive the data provided and the corresponding list having the data elements to be placed in each column.

The general syntax for inserting data into a table one row at a time is as follows:

```
INSERT INTO "table_name" ("column1", "column2", ...)
VALUES ("value1", "value2", ...)
```

Assuming that we have a table that has the following structure,

Table Student_Information

Column Name	Data Type
Matric_No	Varchar (12)
Name	Varchar (50)
Dept	Varchar(50)
Level	Number

and now we wish to insert one additional row into the table representing the data for student with Name: Sonekan Tunde
Matriculation number: NOU050010456
Department: Accounting
Level: 400

The SQL statement will be written as:

```
INSERT INTO Student_Information (MatricNo , Name, Dept, Level)
VALUES ('NO U050010456 ', Sonekan Tunde , Acco unting , 400)
```

3.6 UP DATE Statement

This statement is used to modify the data already in a table or view. The syntax of the Update statement is:

```
UPDATE "table_name" SET "column_1" = [new value] WHERE {condition}
```

For example, say we currently have a table with the following data:

Table Student_Information

Matric_No	Name	Dept	Level		
NOU050010456	Sonekan Tunde	Accounting	400		
NOU050010457	Daramola Kemi	CIS	400		
NOU050010459	Akinlabi Oluwafemi	Economics	400		

If at a point in Sonekan s Tunde career, he decides to have a change of department from Accounting to Economics. His record would need to be updated. To do this, use the following SQL statement:

```
UPDATE Student_Information
SET Dept = Economics
WHERE MatricNo = "NOU050010456 "
```

The resulting table would look like

Table Student_Information

Matric_No	Name	Dept	Level		
NOU050010456	Sonekan Tunde	Economics	400		
NOU050010457	Daramola Kemi	CIS	400		
NOU050010459	Akinlabi Oluwafemi	Economics	400		

Note that the matriculation number which is unique to all students in a university was used as the search key.

3.7 Using the DELETE Statement

The DELETE statement is used to remove rows from a database table or view. The syntax for this is DELETE FROM ""table_name WHERE {condition}

For example, if Daramola Kemi is withdrawn from the university for poor academic performance or voluntarily, her data may be deleted from the student records. To do this, type

```
DELETE FROM Student_Information WHERE Matric_No = " NO U050010457 "
```

The new content of the database will look like:

Table: Student_Information

Matric_No	Name	Dept	Level		
NOU050010456	Sonekan Tunde	Economics	400		

4.0 Conclusion

The use of SQL statement increases the productivity of programmers in the development of applications. SQL statements allows table be created, allow insertion, update, deletion of records, etc to be performed easily.

5.0 Summa ry

This unit covers describe and explain the use of basic SQL statements. Most modern day programming languages support the use of SQL statements. The SQL statement allows easily manipulations of record in database. You will find the skills acquired in this unit very useful in your career as a Data Processing personnel

Exercise

Develop a simple data processing application that have features to add, delete, search, edit, access first, access next, access previous and last student record from a database. The user s interface should look like Figure 1. The information stored for each student contains Matriculation Number, Name, School, Department and Level. Hint: Adapt the sample codes and follow the procedures given below to write your application.

STUDENT REGISTRATION INFORMATION

Figure 1: User Interface

Sample codes

Source Code for command (First)

```

=====
=====
Private Sub Command1_Click()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic
With rstud
If .RecordCount <> 0 Then
.MoveFirst
txtdept.Text = !Dept
txtmat.Text = !MatricNo
txtlev.Text = !Level
txtsch.Text = !School
txtname.Text = !Names
End If
End With
End Sub
=====
=====

```

Source Code for command (Next)

```

Private Sub Command2_Click()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic
With rstud
If .RecordCount > 1 Then
.MoveNext
If Not .EOF Then
txtdept.Text = !Dept
txtmat.Text = !MatricNo
txtlev.Text = !Level
txtsch.Text = !School
txtname.Text = !Names
Else
MsgBox "No more Records Found ", vbInformation, "Message Box"
End
End If
End If
End With
End Sub
=====
=====

```

Source Code for command (Previous)

```

Private Sub Command3_Click()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic
With rstud
If .RecordCount > 1 Then
.MovePrevious
If Not .EOF Then
txtdept.Text = !Dept
txtmat.Text = !MatricNo
txtlev.Text = !Level
txtsch.Text = !School
txtname.Text = !Names
Else
MsgBox "No previous record", vbInformation, "Message Box"
End
End If
End If
End With
End Sub

```

Source Code for command (Last)

```

Private Sub Command4_Click()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic
With rstud
If .RecordCount <> 0 Then
.MoveLast
txtdept.Text = !Dept
txtmat.Text = !MatricNo
txtlev.Text = !Level
txtsch.Text = !School
txtname.Text = !Names
End If
End With
End Sub

```

=====

=====

Source Code for command (Add)

```

Private Sub Command5_Click()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic

```

With rstud

```
If txtdept.Text = "" Or txtlev.Text = "" Or txtmat.Text = "" Or txtname.Text = "" Or  
txtsch.Text = "" Then
```

```
MsgBox "Incomplete Registration", vbCritical, "Message Box"
```

```
Else
```

```
.AddNew
```

```
!MatricNo = txtmat.Text
```

```
!Dept = txtdept.Text
```

```
!School = txtsch.Text
```

```
!Names = txtname.Text
```

```
!Level = txtlev.Text
```

```
.Update
```

```
MsgBox "Registration Successful!!!", vbInformation, "Message Box"
```

```
Call clr
```

```
End If
```

```
End With
```

```
End Sub
```



=====
=====

Source Code for command (Edit)

```
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
```

```
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,  
adOpenDynamic, adLockOptimistic
```

```
With rstud
```

```
If txtdept.Text = "" Or txtlev.Text = "" Or txtmat.Text = "" Or txtname.Text = "" Or  
txtsch.Text = "" Then
```

```
MsgBox "Incomplete Registration", vbCritical, "Message Box"
```

```
Else
```

```

!MatricNo = txtmat.Text
!Dept = txtdept.Text
!School = txtsch.Text
!Names = txtname.Text
!Level = txtlev.Text
.Update
MsgBox "Update Successful!!!", vbInformation, "Message Box"
Call clr
End If
End With
End Sub

```

```

=====
=====

```

```

Source Code for command (delete)
Private Sub Command6_Click()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic
If txtdept.Text <> "" And txtlev.Text <> "" And txtmat.Text <> "" And txtname.Text <>
"" And txtsch.Text <> "" Then
MsgBox "Sure you want to delete this Record", vbYesNo, "Message box"
If vbYes Then
With rstud
.Delete adAffectCurrent
.Requery
MsgBox "Recod Deleted", vbInformation, "Message Box"
End With
End If
End If
End Sub

```

Student Registration Information

Matric No

Name

School

Dept

Level



=====
 =====
 Source Code for command (Search)

Private Sub Command8_Click()

If datstud.constud.State <> adStateOpen Then datstud.constud.Open

If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
 adOpenDynamic, adLockOptimistic

With rstud

.Filter = "MatricNo=" & txtmat.Text & ""

If Not .EOF Then

txtname.Text = !Names

txtdept.Text = !Dept

txtlev.Text = !Level

txtmat.Text = !MatricNo

txtsch.Text = !School

Else

MsgBox "This name was not found in the database", vbCritical, "Message Box"

End If

End Sub

Student Registration Information

Matric No:

Name:

School:

Dept:

Level:

Student Registration Information

Matric No:

Name:

School:

Dept:

Level:

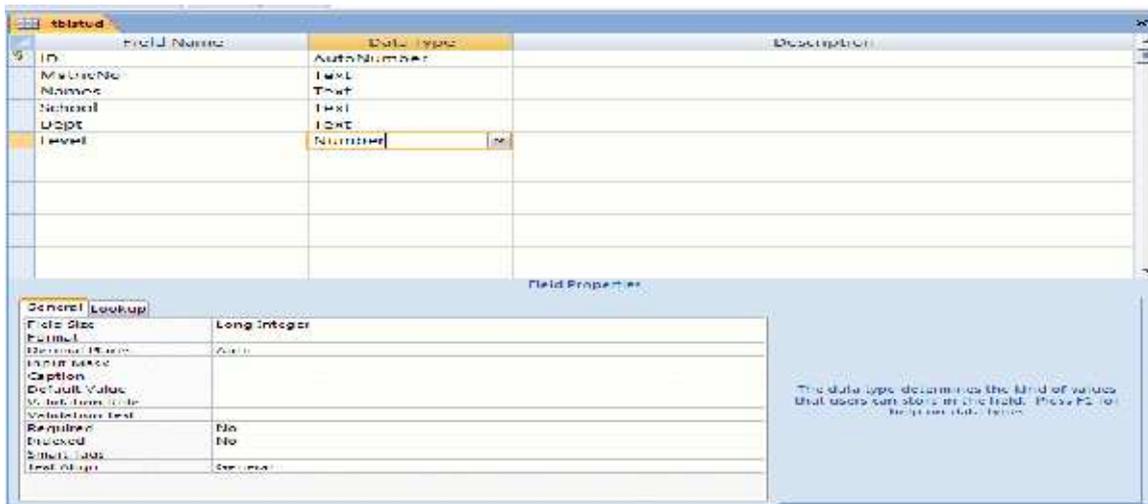
=====

=====

Database Connection

Step 1 Create your database using Microsoft Access or SQL

SCREEN SHOTS



Data base Design view in Microsoft Access

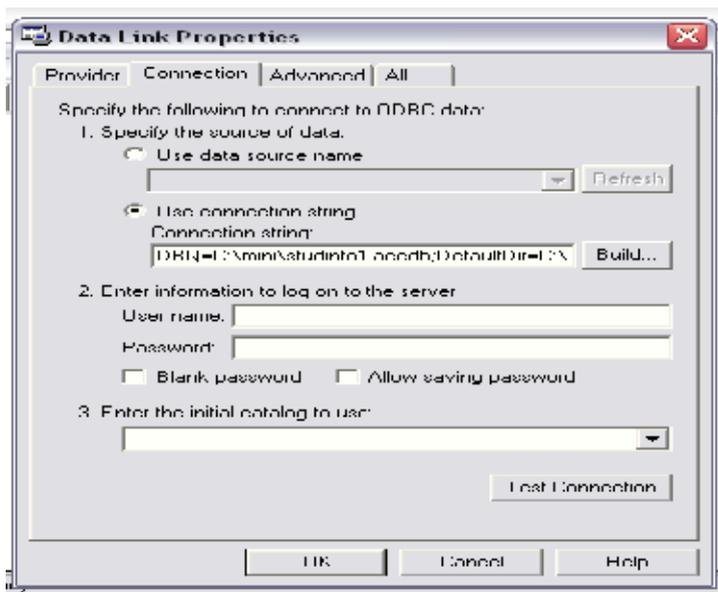
MatricNo	Names	School	Dept	Level
NOU050010456	Sonekon Iunde	Social Sciences	Accounting	400
NOU050010457	Daramola Kemi	Science & Technology	CIS	400
NOU050010458	Femi okeowo	Science & Technology	CIS	400
NOU050010459	Akinlabi Oluwafemi	Social sciences	Economics	400

Step 2 Right click in the project explorer then Add a data environment this is what it looks like:



You could rename it if you wish to but this has been rename form data environment1 to datstud likewise connection1 has also been renamed to constud.

Step 3 Right on the constud an click on the properties this is what you will see:



Step 4 Use connection string as shown on the screen shot then build. To build, you need to supply your data source name. If you have not created any, enter a new data source name.

Step 5 Select the database used from the outlined databases then proceed

Step 6 Repeat step 4 because you don't have any created data source.

Step 7 You then select your database that has been created then you click ok!

Step 8 Test your connection. If successful, it means that your data source has been successfully created otherwise repeat the step again.

Step 9 - Right click in the project explorer and add a module.

Step 10 Declare your recordset in this module as follows

```
Public rstud As New ADODB.Recordset
Public rsass As New ADODB.Recordset
```

Step 11 Link each recordset with the corresponding table that has been created in the database in your form load() as follows:

```
Private Sub Form_Load()
If datstud.constud.State <> adStateOpen Then datstud.constud.Open
If rstud.State <> adStateOpen Then rstud.Open "Select * from tblstud", datstud.constud,
adOpenDynamic, adLockOptimistic
End Sub
```

6.0 Tutor Marked Assignments

Q1. What is a DBMS? Give 5 Examples

Q2 Using any of the DBMSs create table 1(i.e Student_Information) and execute the SQL

SELECT statements (a-g) in section 3.3

7.0 References/Further Readings

Jan L. Harrington, SQL Clearly Explained, Morgan Kaufmann Publisher, An Imprint of Elsevier Science, 2003, USA

David C. Kreines, Oracle SQL, O Reilly, 2000, USA