COMPUTERIZED PC TROUBLESHOOTING AND REPAIRS

BY

ODIFA ENIOLA ORE-OLUWA
PGD/MCS/99/2000/884

DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE
FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

SEPT 2001

COMPUTERIZED PC TROUBLESHOOTING AND REPAIRS

BY

ODIFA ENIOLA ORE-OLUWA
PGD/MCS/99/2000/884

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE, IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF A POST GRADUATE DIPLOMA IN COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGER STATE, NIGERIA.

SEPT 2001

CERTIFICATION

This is to certify that this project carried out by Miss. Odifa Eniola Ore-Oluwa meets the requirements for the award of a Post Graduate Diploma in Computer Science of Federal University of Technology, Minna, Niger State.

________________________ __________________

Prince R. O. Badmus Date

(Project Supervisor)

________________________ __________________

Dr. S. A. Reju Date

(Head of Department)

________________________ __________________

External Examiner Date

DEDICATION

This project is dedicated to the Lord Almighty with whom all things are made possible.

ACKNOWLEDGEMENT

With all humility, I express my profound gratitude to Almighty God for His unparallel love, affection and infinite mercy that make all things possible for those that believe in Him.

I am also grateful to Prince R. O. Badmus, who supervised this project, for his ever-ready posture to attend to me even at odd hours and places and also for his untiring disposition at offering advise on the project.

My profound gratitude also goes to Head of Department, Dr. S. A. Reju for his willingness to make all his students master of their vocation. Many thanks also to all the lecturers in the department: Dr Yomi Aiyesimi, Mallam Audu Isah, Dr. E. N. Ezeako and others, I appreciate your effort and love, God bless you all.

I want to appreciate the effort of Miss Matemi Yemisi for her sleepless night to make sure that this project work out. Also, to the entire member of R. C. C. G, Chapel of Abundance, Bosso parish, Minna.

Finally, a big thank you to my loving husband, Mr. Oke Olusegun Kolawole and our kids, Yetunde Folusho and Oluwatofarati for their prayers and endurance while the course lasted.

ABSTRACT

For an efficient financial management in any organization, financial record keeping is very important. More so, a computerized system of record keeping will go a long way in alleviating the problem associated with the present manual system being operated, misplacement of files or missing of some vital information from staff from staff and other related files.

For the purpose of this project, the proposed computerized system has been designed using the database management system with special preference to Visual FOXPRO. Also, the program is designed such that data can easily be added, deleted and modified.

TABLE OF CONTENT

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

Appendixes:

CHAPTER ONE

1.1 INTRODUCTION

Prior to introduction/invention of computer, jobs were performed manually with so many personnel involved. These jobs were prone to error and can be slow in operation due to sluggishness of some workers. Hence the era was being referred to as 'Dark ages' of Data.

Nowadays, computer is the order of the day; it has over taken manual processing in some organizations like banking, industries, ministries, and Research centres, space mission etc. This is as a result of its accuracy, speed, and its effectiveness.

With the computers becoming an integrated part of our daily activities we need to focus on how it can improve our life. It would improve our life by improving our ways of thinking, our ways of working and lastly our day-to day decision.
This is the drive to encourage the use of new technologies on a regular bases.

Taking all these into consideration, one can easily visualise what would happen if the applied computer should collapse (i.e. faulty); there would be bulk of papers lying all over untreated, airline operation could be grounded etc. Hence, the need to recover the fault machine brings about this project.

The idea behind the design of the project is to provide formalized procedures, steps or guide for PC troubleshooting and repairs.

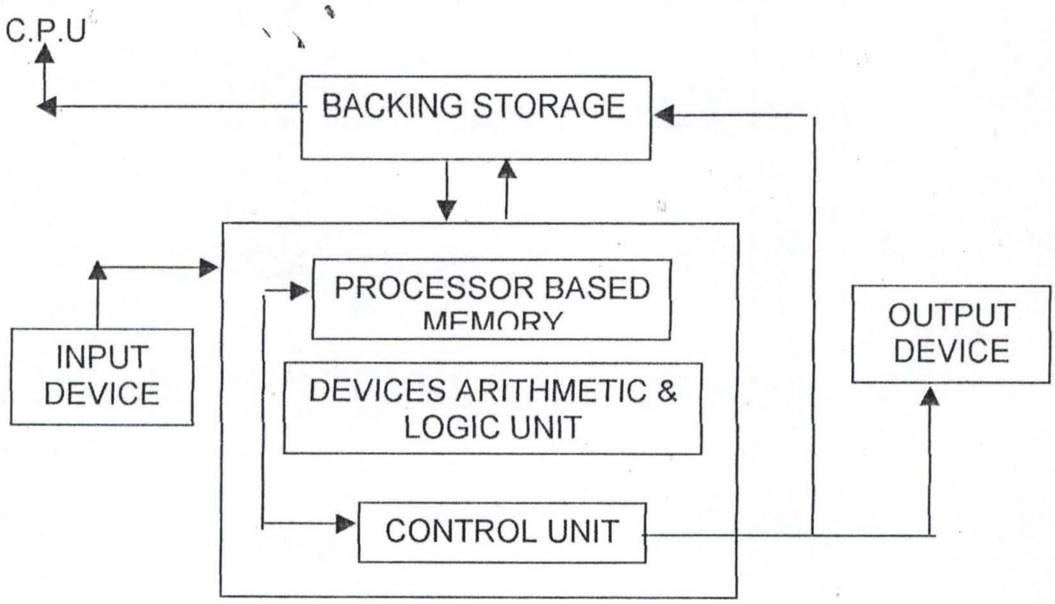
1.1 DEFINITION:

A computer is an electronic device capable of repeatedly performing controlled sequence of operations, accepting data from its environment and processing it to produce results.

A computer can also be defined as a system (or set of machines), which accepts data from an input device, performs operations on the data by means of the processor and transfers the results to an output device. While program is a meaningful sequence of instruction or statement, processing an implicit or explicit order of execution.

1.1.1 BASIC ELEMENTS OF COMPUTER SYSTEM

A computer system consists of five major components, which include: Input, Storage (memory), Processing, Control, Logic and Output



Note that control signals come from the control unit to all parts of the computer system.

Input: An input unit, accepts the necessary input data and instructions. Examples are keyboard, Mouse, card reader etc.

Storage: Storage consists of two types: Main and Auxiliary where computer instructions, data and intermediate calculations are stored.

Processing: Data are entered from input devices as files into the computer memory. They are ready to be manipulated when the computer performs the appropriate arithmetic and Logic operations.

Arithmetic and Logic unit (A.L.U): This performs arithmetic operations and Logical functions in the CPU. It consists of an adder/substractor, one or more working registers to store data being processed and accumulators for storing the results of calculations.

Control: Control is necessary within a computer system to supervise and co-ordinate it's many operations. The control unit in the CPU instructs the various components of the computer (i.e. the input – output devices also referred to as I/O, the ALU, primary and secondary storage) on what should be done and in what sequence. In effect, the control unit is the brain of the computer components in a computer system.

Output: once the data has been processed, it is necessary to communicate this information to the user in the form of output. The type of output device employed will depend on the information desired. Example include printer, monitor e.t.c.

1.1.2 COMPUTER SOFTWARE

In contrast to computer hardware, the computer software consists of the application or program that allow you work on the computer.

Instructions (Program) are fed into the computer in the form of a programming Language that the computer obeys sequentially. These sets of instructions are referred to as software. A good example is the operating system. They are programs that stand between the user and the machine.

Software is classified into two, the system software and the application software.

(a) System software which can also be referred to as operating system (O.S) can be defined as an organized collection of software extensions of hardware, consisting of control routines for operating a computer and for providing an environment for execution of programs.

Operating systems functions are activated through the special operating commands; they are generally supplied by the computer manufacture. Their main functions also include guiding the working of application programs. Another example of system software is FIRMWARE. These are special software whose instructions are hardwired to computer devices.

(b) Application software is, in simple terms, a program or collection of programs that carry out a given task for the user. A few typical software program include, word processor, spreadsheets, databases, games and of course the designed Computerised PC troubleshooting and repairs.

1.2 BRIEF HISTORY OF PERSONAL COMPUTER

1.2.1 HOW COMPUTER DEVELOPED

To think that computers have suddenly arrived on the scene would be wrong, although it is true that their number and use have greatly increased during resent years. Calculators have been in use for very long time, and even in the days of a Navigators and Astronomers; there was a need for some sort of calculating instrument to relieve the human brain of workload.

Blaise Pascal produced the first mechanical calculator in 1862. Others tried to improve on it but until the nineteenth century no real progress was made. In 1901, a Frenchman named Jacquard invented punched card system for controlling the thread on his weaving looms.

Chalse Babbage followed in 1933 with his analytical engine, using punched cards, but the calculating machinery was operated by electromagnetic means. It was introduced in 1889 and was generally used in a highly developed form, right up to wide spread introduction of electronic computers in 1950s

ENIAC (Electronic Numerical Integration and Calculator) was first used at Cambridge University. This brought about the modern electronic computer.

1.2.2 PERSONAL COMPUTER (PC) DEVELOPMENT

The personal computer revolution began in the early 1970"s with the advent of APPLE Personal Computer. The little machine put comparative huge amount of computer power into individual's desk at affordable prices.

With the success of APPLE PC, the big computer manufacturers like IBM and ICL, entered into this rapidly expanding market.

The IBM machines use INTEL 8000 series computer chips and CPUs while the Apple series of machines use the Motorola 6800 family of CPU's and computer chips. The two types of CPU's are not compatible with the might of IBM behind the INTEL based PC, the INTEL chip is the industry standard for personal computing.

We are now in the fifth generation of personal computers and the machine as mentioned earlier are based upon the INTEL 80 x 86 series chip which over the years has been improved upon both in speed and capabilities. The current computer processor chip is called the Pentium. The processor will be called p6. These machines can match mini computer. Frankly, there exists probably no demarcating line between a PC and a Mini computer. This is because of the power with which PC's are now manufactured.

1.3 AIMS AND OBJECTIVES

a. Placing computer in place of an expert so that, by following a set of rules (computational organisations), they could arrive at a decision or recommendation (intelligent) based on the circumstances surrounding the environment.

b. Giving any person (Layman) the skill to diagnose and repair his/her PC and to make intelligent repair and upgrade decisions about the computer.

c. Saving the technicians run memorizing all the faults and their cause with suggested solutions. It aids the technicians in performing their job more easily.

d. Minimizing errors due to wrong prescription/advice being given by technicians.

e. Saving users from spending much on the repair of their computer.

1.4 PROBLEM DEFINITION

As modern industrialized society becomes more specialised, the need to have "expert device" increases rapidly. This necessity is visible in all forms of life. For instance, a consumer wishing to buy a new washing machine may base his decision on a wide variety of technical and aesthetic factors before making his choice. In most cases he will ask a sales assistance for "expert advice" in making these decisions.

There are a number of reasons for doing this: firstly the knowledge of an expert who is going to retire soon can be stored so that his knowledge is not lost to the company forever.

Secondly, for implementing expert systems is to lighten the load on the specialists. If an expert can be created to solve the easier problems in a particular domain then the specialist will be left with the more demanding and rewarding problems to solve.

Hence, there is need for an expert system to replace man based on the following basics and inherent problem areas discovered.

a. Repairing process takes a very long time due to indecision caused by unavailable data / information for the technician

b. Wastage and unnecessary spending do occur due to wrong diagnosis/ prescription/advice by technicians for instance, in a situation where a user gets an assorted NMI (Non-Makable Interrupt) errors, and memory allocation error message. A technician may say that the problem is bad software and reloads his hard – disk. Another technician may say that it is memory problem and may replace nine chips charging ₦20,000. While the computer may only have power supply problem.

1.5 DEFINITION OF TERMS

RAM ----------- Random Access Memory

ROM ----------- Read only memory

BIOS ----------- Basic Input and Output System

DOS ------------ Disk Operating System

CPU ----------- Central Processing Unit

PC ------------ Personal Computer

FAT ----------- File Allocation Table

SCSI ----------- Small Computer System Interface

NMI ----------- Non-Maskable Interrupt

DMA --------- Direct Memory Access

POST -------- Power On Self-Test

CRC --------- Cyclic Redundancy Code

ECC -------- Error Correction Code

MFM ------- Modified Frequency Modulation

RLL --------- Run Length Limited

ESDI -------- Enhance Small Device Interface

1.6 SCOPE AND LIMITATION OF STUDY

The project should be able to suggest the reason behind the fault (area of problem) and advice on what to do in order to repair the device. The areas of PC problem to be treated are:

a. No boot (Computer fails to start up)
b. Power supply
c. No video (No Visual display)
d. Video quality
e. Drive (Floppy/Hard disk)
f. Intermittent weird failure
g. Mouse/Trackball
h. Undefined problem
i. Error Codes/Messages
j. Error beeps

1.7 COMPUTER REPAIR ECONOMICS

When an electronic part in a computer goes bad, one mighty think it is a simple matter to identify the offending chip, replace it, and go on about one's business.

The problem is that this method takes time. A lot of time. The cost for the repairperson's time often exceeds the cost for a whole new part.

The average component repair, as opposed to replacement, takes about three hours. The time includes determining which part is bad, repairing it, replacing it, and testing it to make sure the repair was made correctly.

It is far more cost-effective, quicker and simpler to replace an offending board.

1.8 THE REPAIR PROCESS

The most important tool needed in repairing is the ability to think logically and to follow a step-by-step process to isolate the faulty part.

Figuring out which part to replace can take at least an hour sometime much longer. For example, the symptom may be a screen message, "Abort, Retry, Ignore, Reading Drive C:," This can be caused by any of the following:

- A bad track, which the manufacture located and printed on the drive label, was not entered as bad when the drive was formatted. Now Date has been placed on this bad track.
- The heads of a driver have drifted out of alignment with the tracks, causing some sports to be unreadable
- The hard drive cable has lost connection or has somehow been damaged.
- The hard drive is dying and must be replaced.
- The 12-Volt line on the power supply is too high / too low, or unstable, and the power supply must be changed.
- The DMA controller on the main board is bad. The Main board must be replaced

This project will tell you what to do if you have the screen message "Abort, Retry, Ignore, Reading Drive C:" it helps you determine which problem is the one you are facing and how best to solved it.

1.8.1 WORKSHOP BASICS

In order to make a sensible diagnosis of a fault, one must have together a list of all the symptoms of the problem.

➢ What does the equipment do whilst the error is present?

For example:

- What visual indications are there?

- Are there any audio indications?
- Is anything loose?

➢ When dismantling a piece of equipment, you must remember how it came apart so that you will know how to put it back together again. Note things down on paper to help you. For example:

- The types, lengths and colours of screws.
- Whether that cable went to left or right of that component
- The colours of the cables you remove and their order, say from left to right or from top to bottom.

1.9 THE NEEDS FOR THE AUTOMATION

Since computer has almost taken over manual processing in most organizations like banking, industries, ministries, and Research centres, space mission etc. then the use and importance of computers can not be over emphasized in our society,

We can then imagine what would happen if such important tool (computer) breaks down then there would undoubtedly be many jobs left undone.

Therefore, the need to recover this faulty machine brought about this project. To give a computerized guide/rules or ways of rectifying or repairing the faulty computer.

CHAPTER TWO

2.0 THEORY AND ANALYSIS OF PC TROUBLESHOOTING AND REPAIR

2.1 GENERAL TROUBLE SHOOTING RULES

Some of the rules governing troubleshooting and repair of a pc are as follow:

There is always the need to develop a sort of guideline for every step to be taken in any assignment. This guideline has been outlined as a rule while embarking on PC troubleshooting and repairs.

The following rules have kept some people out of trouble for a long time and will be of use to many more.

(1) **Be confidant " I will win": -** As a trouble-shooter one has to be confident of himself. Take whatever problem may occur as not much to the machines (PCs) this confidence propels one to do the right thing at the right time.

(ii) **Write everything down: -** While carrying out the process of troubleshooting, it is always advisable to write down the observation and if possible jot down some vital points on the way forward. This will help the trouble-shooter in giving the right prescription to the observed problem outside the field. The aim of writing down is to avoid forgetting the fault and its probable cause.

(iii) **Do the Easy Thing First: -**The proverbial saying "known to unknown" holds in this case. One should start from the easiest guess to the cumbersome one. For instance, if it is the video problem, and it is not software, then there are four things to suspect or swap: The motherboard, the video board, the cable, or the monitor. What gets swapped first is the easiest thing among the four that is the cable. Reducing it from the known to the unknown can solve so many complicated faults or problem.

(iv) **Reboot and Try Again: -** There are cases when PCs encounter minor problems that can be solved by mere rebooting via the system tools checker (scandisk). So for one to go into serious PC troubleshooting, one needs to go through rebooting to find out if the problem is minor or major.

(v) **Simplifying Your Configuration: -** The average PC has several component parts, each parts having its own functional responsibility. When there is a fault, one should be able to go through the component configuration one after the other to enable one narrow down where the fault is coming from. This attempt implies simplifying the configuration that is removing one component after another.

(vi) **Draw a Picture, Separate into Components And Test: -** When there is PC problem, there is always the need to draw a diagram of the areas suspected. Doing this divides the problem into various sub-components and each component can then be tested directly with test tools or indirectly by swapping with working one since system test tools are not easy to come by. Sketching the system and following the step-by-step approach helps to take a stand on whether the fault is software or hardware based. It is advisable to first take on software problem test before going for the hardware aspect.

(vii) **Never assume something is Good:-** It is far too easy to assume that some components are blameless. One may wonder how can it be this or that when I just bought them? A good trouble-shooter should endeavour to be patient and calm while carrying out the act of troubleshooting. He should subject everything to scrutiny, including the documentation.

(viii) **Be prepared to believe that sometimes Documentation lie:-** There are situations where it will be documented that the motherboard for instance has one or two jumpers for setting things like VGA, CMOS, etc whereas there is more or less the stated number.

In this case, one should be ready to go through the rigours of reading the manual and observing directly the sketches attached and comparing it with the original/real board. If done this way, problem won't occur

2.2 STEPS FOR TROUBLE SHOOTING

Having gone through the rules of general PC troubleshooting, it is pertinent to consider the steps involved in the troubleshooting proper.

Troubleshooting can be described as the act of finding fault existent in a system. This fault as earlier described can be from user, software, or hardware. Whichever is the case, to trace out a fault in a PC, one should be able to first: -

(I) Check for external signs: - If the computer has indication lights, what do they signify. ? Are all the lights showing on the modem? Does the printer indicate "ready"? Is the hard disk drive squealing or grinding? Does the monitor image look bent? Your drives and other peripheral produce hums, whirs and clicks. After a while, these noises become familiar and any variation in them signals a problem.

One should pay attention to these signs. Since the first step in successful troubleshooting is isolating the problem component, these signs can point the way and there should be documentation of the lights that are ON and OFF and the switch positions.

(ii) Run Diagnostic Programs: - Many PC type machines come with diagnostic programs, which can help pinpoint a problem (assuming, of course, that the computer is well enough to run them in the first place). Other PCs, like some "no names" clones, do not.

There are various good third-party diagnostic programs that can pinpoint problems, repair it or proffer suggestions on how to solve some problems.

(iii) Interpreting PC Error Codes: - Many PCs complain that there is a problem with in English. You will see messages like "floppy controller failure, Drive unavailable" etc. a good troubleshooter should be able to decode some of this basic error codes and messages and apply this knowledge to correct an error/problem.

(iv) Power On Self-Test (POST): - A short diagnostic routine runs on a desktop computer any time the computer is powered. It does a check to see that the basic important hardware exists and performs memory test. This check is called the power on self-test (POST) or the BIOS listing, power on Diagnostic (POD).

The above tests are preliminary in nature but can solve some minor problems like some users or software problems. The following steps can be applied if the problem persists:

(a) Check the nuts behind the system.

(b) Check to ensure that everything is plugged in: power, monitor, mouse, keyboard, printer, etc, depending on the type of fault.

(c) Check the software.

(d) Ask: "what am I doing differently? "What's new? "

(e) Check external signs and make notes of them

(f) Run better diagnostic disks and when the last six steps fail,

(g) Disassemble the machine, push the socketed chips back into their sockets, clean the connectors and put the machine back together and test again. . Having gone through the rules of general PC troubleshooting, it is pertinent to consider the steps involved in the troubleshooting proper.

2.3 BASIC TROUBLESHOOTING GUIDELINES

The basic and general troubleshooting procedures and guidelines are listed below.

Before starting any system troubleshooting, there are few basic steps that should be performed to ensure a consistent starting point and to how isolating the failed components.

1. Turn off the system and any peripheral devices.

Disconnect all external peripherals from the system, except for the keyboard and the video display.

2 Make sure the system is plugged in to a properly grounded power outlet.

3. Make sure the keyboard and video displays are connected to the system. Turn on the video display and turn up the brightness and contrast control to at least two-thirds of the maximum. Some displays have on screen controls that may not be instinctive. Consult the display documentation for more information on how to adjust these settings.

4. To allow the system to boot from a hard disk, make sure there is no floppy disk in the floppy drive or put a known good bootable floppy with DOS or diagnostics on it in the floppy drive for testing.

5. Turn on the system. Observe the power supply and chassis fan (if any), and the lights on either the system front panel or power supply. If the fans don't spin and the lights don't light, the power supply or motherboard might be defective.

6. Observe the power on self test (POST). If no errors are detected, the system beeps once and boot up. Errors that display on the screen (non-fatal errors) and which do not lock up the system display a text message that varies according to BIOS type and version.

7. Confirm that the operating system loads successfully.

2.4 DIAGNOSTIC ERROR CODES AND DESCRIPTION

A. SYSTEM BOARD ERRORS

Code	description
101	interrupt failures
102	timer failures
103	timer interrupt failure
104	protected mode failure

B. MAIN MEMORY (RAM) ERRORS

CODE	DESCRIPTION
201	memory test failed
202	memory address error
203	memory address error

C. DISKETTE DRIVE/CONTROLLER FAILURES

CODE	DESCRIPTION
601	Adapter or drive failed POST
602	Diskette test failed: boot record not valid
606	Diskette verify function failed
607	Write – protected diskettes
608	Bad command diskette status returned

DIAGNOSTIC SIGNAL AND PROBABLE CAUSE

TOPIC	DESCRIPTION
No beep, nothings happens	Power supply bad or not plugged in
Continuous beep	power supply bad or keyboard stuck
Repeating short beep	power supply bad
1 long beep, 2 short beep 2	system board failure
1 short beep, no boot	Floppy drive adapter cable
1 short beep, blank screen	Failure or lack of display adapter cable
Short beeps, (ps/2)	Configuration error
Error beeps	Ram, wrong connection
No boot	power supply, VGA, RAM
Error codes / message	Corrupt software, incomplete installation
System Hanging	Incomplete software, incomplete Installation
Cursor not blinking	mouse not detected
Power On but no windows	CMOS setting not correct.

2.5 MEASURES TO ADOPT FOR A HEALTHY PC

One of the best measures to adopt is to make the PC environment friendly. This involves doing the following: -

Ø Check power consideration: There should be no other heating element like heater, iron etc in the same outlet as a PC. There should also not be large electric motors (refrigerator, air conditioners) on the same line or any kind of power noise protection.

Ø Check temperature ranges: temperature range in the PC environment should be between 65 degrees F (18 degrees C) maximum.

Ø Prevent dust builds up: power supplies with filtered fan that suck air in through the back should be preferred rather than the one that pull through the front.

Ø Make sure that there is no vibration source like impact printer on the same level/take the hard disk.

Ø Be familiar with the following:

(a) Packing hard disks

(b) Leaving the machine on all the time.

(c) Keeping cables screwed in and out of the way

(d) Basic "don't do this " on things in DOS, like formatting the hard disk.

Ø Protect against static electricity

Another important measure to adopt for a healthy PC is to sample a preventive maintenance program. Preventive maintenance (PM) implies taking the machine off a person's desk at regular intervals, perhaps as often as every six months, and move it to a support person's "shop" to give it a good going-over so that you will anticipate problems.

The following PM procedures are considered important for a continued healthy condition of a PC.

(1) Pickup the PC at its worksite. Yes, this takes more time than having it delivered to your work place, but you will learn a lot. Examine: -

Ø Are the connectors screwed in?

Ø Have screws disappeared from the back of the machine?

Ø What else is plugged into the PC outlet? No heater, pressing iron? Etc

Ø Is the PC near a window? Is it in a location that can get direct sun at some points in the day?

(2) Ask if the machine is doing anything strange.

(3) Ensure that the hard disk is backed up.

(4) Pack the hard disk and take the machine to a shop.

(5) Run the machines diagnostics. It is a good idea to run scandisk to see what percent of the users drive consist of "lost cluster"

(6) Examine Auto EXEC. BAT and configure system for any obvious problems-

Lack of BUFFERS command, for example. If it is a windows machine, look at the INIs for obvious tempering. If running windows 95/98 or NT, just run the computer a bit to ensure that it is not obviously misconfigured

(7) Remark the head, if the drive is not self- packing.

(8) Disassemble the PC

(9) Clean the edge connectors with connector cleaner and a hint free cloth or a hard white artist's eraser

(10) Push the chips back into their sockets.

(11) Use canned air to remove dust from circuit board - do not forget the circuit board under the hard disk.

(12) Reassemble the PC. Ensure that all of the cables are securely in place.

(13) Rerun the diagnostics

(14) Ensure that all screws are present. If they are not, add screws.

(15) If the drive supports a low -level format, then low - level format the hard disk with a non- destructive - reformatter program like Disk Technician (prime solution), SPINRITE (Gibson Research), OPTUNE, or the like.

2.6 COMPUTER SYSTEM COMPONENTS

A modern PC is both simple and complicated .It is simple in the sense that over the years Components used to construct a system have become integrated with other components into fewer and fewer actual parts. It is complicated in the sense that each part in a modern system performs many more functions than did the same types of parts in older systems.

Below are the components needed to assemble a basic modern PC system:

- Motherboard (system / main board)

- Processor
- Memory (RAM)
- Case (Chassis)
- Power supply
- Floppy drive
- Hard disk
- CD- ROM drive

- Mouse
- Video Card
- Monitor (display)
- Sound Card
- Speakers
-Keyboard

2.6.1 SYSTEM CASE (CHASSIS)

The case is the frame or chassis that houses the motherboard, power supply, disk drives adapter cards, and any other physical components in the systems. There are several different styles of cases available, from small or slim version that horizontally on a desktop to huge tower types that stand vertically on the floor, and even some that are designed to be rack mounted for industrial use. In addition to be physical styles, different cases are designed to accept from factor motherboards and power supplies. Some cases have features that make installing or removing component easy, such as a screw less design that requires no tools to disassemble, side open panels or trays that allow easy motherboard access, removable cages or brackets that gives access to disk drives, and so on. Some cases include additional cooling fan for heavy-duty systems, and some are even available with air filters that ensure that the interior will remain clean and dust free. Most cases include a power supply, but you can also purchase bare cases and power supplies separately.

2.6.2 MOTHERBOARD

The motherboard is the core of the system. It is really the pc. Everything else is connected to it, and it controls everything in the system. Motherboards are available in several different, shapes or form factors. The motherboard usually contains the following component:

- Processor socket (slot)

- Processor Voltage regulators
- Motherboard chipset
- Level 2 cache (Normally found in the CPU today)
- Memory SIMMs or Dimm sockets
- Bu slots
- Rom BIOS
- Clock / Cmos battery
- Super I/O chip

The chipset contains all the primary circuitry that makes up the motherboard; in essence, the chipset is the motherboard. The chipset control the CPU or processor bus, the L2 cache and main memory the PCI (Peripheral component interconnected) bus, the ISA (Industrial standard Architecture) bus, System resources, and more. If the processor represents the engine of your system, the chipset represents the chassis in which the engine is installed. As such, the chipset dictates the primary features and specifications of your motherboard, including what type of processors, memory, expansion cards, disk, and so on the system supports.

Note that most newer (Pentium celeron / II / III class) systems include the L2 cache inside the processor rather than on the motherboard.

The chipset plays a big role in determining what sorts of features a system can support. For example, which processor you can use, which type and how much memory you can install, at what speeds can you run the machine, and what types of system buses can your system can support? Are all tied into the motherboard chipset. The ROM BIOS contains the initial POST (Power – on Self - Test) program,

Bootstrap loader (which loads the operating system), drivers for items that are built into the board (the actual BIOS code), and usually a system setup program (often called cmos setup) for configuring the system.

MICROPROCESSOR

The processor is often thought of as the "engine" of the computers. Also called CPU (central processing unit), it is the single most important chip in the system because it is the primary circuit that carries out the program instructions of whatever software is being run. Modem processors contain literally millions of transistors, etched onto a tiny square of silicon called a die, which is about the size of your thumbnail. The processor has the distinction of being one of the most expensive parts of most computers, even though it is also of the smallest parts.

Microprocessors can be implemented in one to ten IC package using large scale Integrated (LSI) Technology.

They are variable in various word sizes varying from four to sixteen bits.

BIOS ROM (SYSTEM ROM)

A Rom is a read only memory chip i.e. it is another kind of memory exists that cannot be altered but only read. This is a memory some (usually the manufacturer) Loads just once with a special device such as a Prom blaster, an EPROM programmes, or the like. You can read information from ROMS, but you cant write new information into it. Rom has the virtue of not loosing its memory when you turn the machine off.

In essence, you can say that ROM contains the software that tells the system how to use a circuit board. ROM is found on expansion board like Ems, LAN or EGA cards, also on the system board. The ROM on the system boards contains a piece of software called BIOS (Basic input/output system). DOS doesn't communicate directly with your hardware but issue command through BIOS. That is why BIOS is so important. It determines in large measure how compatible your pc is. IBM's BIOS is the standard of compatibility.

MEMORY (RAM)

The system memory is often called RAM (Random Access Memory). This is the primary memory, which holds all the programs and data the processor is using at a given time. RAM requires power to maintain storage, so when you turn off the computer everything in RAM is cleared; when you turn it back on, the memory must reload with programs for the processor to run. The initial programs for the processor come from a special type of memory called ROM (Read Only Memory), which is not erased when the power to the system is turned off.

The ROM contains instructions to get the system to load or boot an operating system and others programs from one of the disk drives into the main RAM memory so that the system can run normally and perform useful work.
Memory is normally purchased and installed in a modern system in SIMM (single inline memory module) or DIMM (dual inline memory module) form.

HOW THE MOTHERBOARD WORKS:

To understand how the main board components interact, consider what happens, Step-by-step, when we power a computer. The initial steps are into the circuitry.

The power supply comes up to speed, and then sends a "power good" signal to the clock-reset chip on the main board. The clock reset, in turn sends a hard reset to the microprocessor (CPU.) The CPU resets and initialised (runs a self-test and prepares the bus for use).

The final, hard-wired step in the microprocessor-reset procedure involves jumping to a very high address to look for further instructions. These is a redirect pointer at that high address, which sends the microprocessor to the starting address of the POST (power –on-self test) and initialisation routine in the BIOS ROM.

The ROM POST /initialisation routines creates a computer from a di coordinated pile of circuit.

During this procedure, the Rom BIOS writes its names, date and copyright signature on the screen; checks the keyboard (Caps lock and Num lock lights flash on and off); read the CMOS to learn what equipment the CMOS believe is installed; Counts and test the system memory; and puts a system memory message on the screen.

The Rom next tries to boot from a disk in A, look at track 0,sector 1,the DOS boot sector. If no disk is present in drive C, the hard disk, again checking only track 0,sector 1.

HOW TO TEST THE MAINBOARD:

Every view of the Mainboard components is individually. The ROM and RAM chips can be replaced, battery can also be replaced.

A faulty mainboard gives error like (Non-Maskable Interrupt).

The NMI occurs most often when the memory parity (a test for memory recording accuracy) shows a memory error. This interruption is called Non-Maskable Interrupt because the hardware cannot mask (selectively ignore) this interrupt while processing another task. When an NMI occurs, the NMI error message goes up on the screen and everything is deadlocked until you address the NMI error matter what else is going on.

There are four major causes of parity error:

(i) A bad RAM chip

(ii) A drop in voltage

(iii) Malfunctioning address Logic to the memory chips that causes one RAM chip to be mistaken for another.

(iv) A bad parity Logic chip that reports a parity problem when there is more.

Causes (i) and (ii) (a bad chip or low power) are the most likely to occur. While causes (iii) and (iv) are symptoms of bad (and non replaceable) chips or circuits on the mainboard.

2.6.3 POWER SUPPLY

The power supply converts power coming into your computer from the wall outlet to the forms usable by the electronics in the system. It changes incoming alternating current (Ac) at 120 or 240 volts (v) to direct current (DC) at 5 and +/- 12. A good power supply does more than this: appliances and other electrical devices create noise in the power at your wall outlet, as do lightning strikes and other effects further away. If that noise gets through the power supply into the electronics in the computer, it causes trouble ranging from erratic operation to complete shut down. A quality power supply will be more resistant to these problems giving more reliable operation from your computer.

HOW THE POWER SUPPLY WORKS:

The power supply has a very simple jobs to provide deem smooth power at +12 volts Dc, -12volts Dc,+5 volts Dc, and 5volts Dc.

It monitors its own power output on start-up and send a "power good" signal to the main board when the voltage have stabilised at their required levels. The microprocessor resets and boot up starts when the mainboard receives this signal.

HOW TO TEST THE POWER SUPPLY:

You can easily test the voltage produced by your power supply with a hand held voltmeters. To do this, you must hook the power supply up to 240 volts and to a load of some kind of disk drive or the mainboard will do .All the four-wire connector are identical, so you can use any one of them to test your power supply while it is hooked into your computer. This is probably the best way to test the power supply since it should perform properly when fully loaded with all add-on boards, drive, e.t.c

2.6.4 MONITORS (DISPLAY)

In most systems, the monitor is housed in its own protective case, separate from the system case and chassis.

In portable systems and some Low-cost pcs, however, the monitor is built into the system case. Monitors are generally classified by three major criteria: Diagonal size in inches, resolution in pixels, and refresh rate in hertz (HZ).
Desktop monitors usually range from 14"to 21" diagonal measure (the actual viewable area is smaller than the advertised measure). LCD monitors in portable systems range from 11"to 14".

2.6.5 FLOPPY DISKS DRIVE:

The floppy drive is a simple, not expensive, low capacity removable, magnetic storage device. For many years, floppy disks were the primary medium for software distribution and system backup. However, with the advent of CD-ROM an DVD-ROM discs as the primary method of installing or loading new software in a system, and with affordable high capacity tape drives for backup, the floppy drive is not used very often in most modern systems, except perhaps by a system builder, installer or technician. Because the floppy drive is the first device from which a PC attempts to boot, it is still the primary method that is used for loading initial operating systems' start up software and core hardware diagnostics. Recent advancements in technology have created new types of floppy drives with up to 120 MB or more of storage, making the device much more usable for temporary backups or for moving files from system to system.

2.6.6 KEYBOARD:

The Keyboard is the primary device on a PC that is used by a human being to communicate with and control a system. Keyboards are available in a large number of different languages, layouts, sizes, shapes, and with numerous special features or characteristics. One of the best features of the PC as designed by IBM is that it was one of the first personal computers to use a detached Keyboard. Most systems prior to the PC had the keyboard as an integral part of the system chassis, which severely limited flexibility. Because the PC uses a detached Keyboard with a standardized connector and signal design, in most cases, it is possible to connect any PC compatible Keyboard you want to your system, which gives you the freedom to choose the one that suit you best.

2.6.7 MOUSE:

With the advent of computer operating systems that used a Graphical User Interface (GUI), it became necessary to have a device that enabled user to point at or select items that were shown on the screen. Although, there are many different types of pointing device in the market today, the first and most popular device for this purpose is the mouse. By moving the mouse across a desk or table top, a corresponding pointer can be moved across the computer screen, allowing items to be more easily selected or manipulated than they can with a keyboard alone. Standard mice, as used on PCs, have two buttons: one for selecting items under the pointer, and the other for activating menus. Mice are also available with a third button, a wheel, or a stick, which can be used to scroll the display or for the special function.

2.6.8 CD ROM DRIVE:

CD (compact Disc) and DVD (Digital Versatile Disc) ROM (Read Only Memory) drives are relatively high capacity removable optical drives. They are primarily a read- only – medium, which means the drivers can only supply information, and the data on the discs can not be altered or re-written. Writeable or re-writeable versions of the discs and derives are available, but they are much more expensive than their read – only, and are therefore not included in most PCs. CD-ROM and DVD-ROM are the most popular media for distributing software or large amounts of data. They are affordable because they are produced in large quantity and they can hold a large quality of information.

2.6.9 HARD – DISK DRIVE:

The hard disk is the primary archival storage memory for the system. It contains copies of all programs and data that are not currently active in main memory. A hard disk drive is so named because it consists of spinning platters of aluminium or ceramic that is coated with magnetic medium. Hard drives can be created with many different storage capacities, depending on the density, size, number of platters.

Most desktop systems today use drives with 3.5 – inch platters, whereas most laptop or notebooks computers use 2.5 – inch platter drives.

HOW A HARD DISK WORKS:

Hard disk is similar to floppy, disk but the technology is superior in every way. Hard disks spin faster, hold more data per inch, move from track to track more quickly, and accommodate multiple platters in each unit.

Hard disks come in half –height and full – height models and in 5 – inch and 3 – inch styles. All hard disks share the same basic design. One or more polished aluminium disks, called platters spin inside the hard drive.

The platters are coated with oxides of magnesium, chromium, and / or iron. On each side of each platter, a read / write head guides on a cushion of air very close to the platter, but not quite touching it. The read / write head record data as magnetic masks on the platters.

HOW TO TEST A HARD DISK:

To being with, one can test to see if the hard disk platters are spinning when the power is on. While the computer is on, but not doing any work unplug the wire, hard disk power connector, and then plug it back in. You should hear and feel the hard disk slow down, stop spinning, then start up again as power is re-applied. If the hard disks fail this test, check the power connector with a voltmeter (+ 5 volts and +12 volts) before condemning the hard disk. If the power connector voltage is okay, replace the hard disk.

HARD DISK CONTROLLER:

The hard disk controller mediates between DOS and one or two hard drives. DOS requests read or write action at a particular sector and track on the hard disk. The controller translates this instruction into electrical signal to the steeper motor, which moves the heads inside the hard disk drive. 286 and 386 computers generally use combination cards, with both floppy and hard drive located on the same card.

HOW A HARD DISK CONTROLLER WORKS:

Three things happen when the microprocessor send an I / O (Input / output) write signal down the bus (the major internal information path) along with the address of the hard disk controller.

First, it addressed logic chips on the controller, which are responsible for monitoring the bus and recognising the controller's code, the controller pays attention to the next bus signals.

Next, DOS requests a read/ write operation by track and sector number. it tells the hard disk where (to what track) to move the assembly and what sector on the track should be used to store the upcoming data.

Finally, DOS sends the data it wants to record. The hard disk controller is responsible for packing (encoding) the data before the hard disk records it. When the data is read back, everything recorded on the hard disk is sent to the disk controller. Data separation circuits locate on the controller, sort out this raw information, discard the house keeping bits, and send a clean stream of stored data back to the microprocessor through the bus.

There are two popular ways to package information: MFM (Modified Frequency Modulation) and RLL (Run Length Limited) encoding. MFM encoding uses 17 sectors spaced relatively for apart. RLL data encoding uses 26 sectors per track and packs many more magnetic marks on each track, so an extremely precise hard disk is required.

HOW TO TEST HARD DISK CONTROLLER:

Neither a hard disk nor a hard disk controller can be tested alone. These devices are tested by seeing how they do their jobs, and they should / must be installed to any work at all. It is very difficulty, however, to know which is bad without substituting a known – good hard disk or a known –good controller and then observing if the symptom is gone.

CHAPTER THREE

3.0 DESIGN OF NEW SYSTEM

3.1 GENERAL REQUIREMENT FOR SOFTWARE SYSTEMS.

As end-user requirements become more exact, collecting of programs grow in complexity and size. The following requirements were taken into consideration in the development of the program.

1. **Correctness and Reliability:** These outweigh all the other requirements. Faults in the program were taken into consideration and corrected so that Logical errors can be avoided, as the computer cannot detect it.

2. **Flexible and Reusability:** The program was designed in such a way that it can be modified.

3. **Efficiency:** As a software system grows in size and complexity, it also tends to run slower, slow response time may be quite time consuming and strenuous but efficiency was not compromised at the expense of correctness and only in extreme cases at the expense of flexibility. The program is quite efficient as it minimises time and maximises efficiency.

3.2 SYSTEM DESIGN, SPECIFICATION AND PROCEDURE.

System design is the use of a programmer's creative ability in organising a logical and feasible procedure for a computerised system, the design stage of the **TRAINING GUIDE** Software would produce some outputs as follows:

(a) To list all the necessary files that would be used.

(b) To keep log-book of all successive operation on computers.

(c) To keep status of the user and the nature of the work done both on site and the office on the system

(d) To provide exit key at any level in the program

(e) To develop procedures and programs

(f) To identify how all programs will work together as a system.

The approach to be used to achieve this is structured programming. Structured programming means, techniques concerned with improving the programming process the better organisation of programs and better programming notation to facilitate correct and clear description of data and control structure.

3.3 PHYSICAL AND LOGICAL DESIGN OF THE SYSTEM

The physical and logical design of the system deals with the physical construction of the logical design of the proposed system. It has to do with program specification of output, input file and processing into computer software.

The design of the computer software is important to ensure that the actual program produced performs the task intended and to allow for future modification to be performed efficiently and with minimum destruction to the design of the system.

3.4 SYSTEM LIMITATION

This project is limited on IBM PC types and the compatibles.

3.5 CHOICE OF PROGRAMMING LANGUAGE.

From the previous analysis, it is pertinent to say that the proposed system is going to be used to store large number of data/information, and time- to-time retrieval of record. Due to this fact, the choice of the programming language chosen for the development of the system is Database Management System (DBMS) package with special preference for Visual FoxPro.

3.6 FEATURES OF THE PROGRAMMING LANGUAGE CHOSEN.

The choice of Visual FoxPro arose because of the following reasons and features posed by the application software.

(i) It is easy to write an interactive user interface program and also simple to understand.

(ii) It is users' friendly

(iii) It reduces data redundancy

(iv) Data integrity can be maintained

(v) Provides easy and greater access to information

(vi) Individual database file can be designed to meet specification requirement of a particular functional unit of an organisation

(vii) The Visual FoxPro allows the source program to be compiled to an executable file thereby allowing the program to be run independent of the application software that was used in coding the program, for example, Visual FoxPro.

(viii) It has quality graphical users' interface

(ix) It is easy to maintain and to debug; it has debugging tools.

CHAPTER FOUR

4.0 SOFTWARE DEVELOPMENT/ IMPLEMENTATION,

4.1 PROGRAM DEVELOPMENT

The process of software development is a co-operation effort of the user of the software and system engineer. While the engineers are the professionals that deals with the technology and its application to project information processing needs, the user of the software on the other side have on in depth familiarity with the respective manual functional areas.

The skill and knowledge of these two groups complement each other and can be combined to create any type of information systems during the course of the software development process.

However, because software development is a team effort," system /software development methodology " that provides a frame work for cooperation and successful development of the system. This step-by-step system development procedures is as illustrated below in a tabular form.

STEPS	ACTIVITIES	MEASURABLE OUTPUT
Analysis and specification	Appraisal of existing solution Identification of users requirement	
Design	Design the overall program Structure design the detailed processor processing logic	Process logic specification, using pseudocode or flow chart techniques
Programming	Writing of code in the	Hand written program

	appropriate processor language Entry of written code into the computer	code. Computer printed out of code
Testing and debugging	Removal of syntax and logic errors. Final testing of program	Error free Program execution
Installation And Maintenance	Error free program installation. User training	Program documentation and upgrade procedures

4.2 SYSTEM IMPLEMENTATION

The design of a new system is never complete until it has been tested, implemented and finally documented.

The implementation phase is basically meant for the user of the system and it entails translating the ideas and details plans developed in the system study into actual practice. It is also the occasion for effecting the end view for the system, to see it functioning and to monitor its procedures.

Efforts have been made to ensure that the program is working as desired and that the system will function as processed.

The system package has been developed in an easy and interactive way to make the implementation of the system comprehensive. All the necessary information within each module has been taken into full consideration, thus providing a menu driving for the operation to be performed on each menu option.

4.2.1 IMPLEMENTATION TOOLS

The system is an integrated user friendly and menu driven package. It has been developed as a troubleshooting and repair guide for any user.

The application package is written in visual FoxPro programming language. It is designed to run on one standard operating system window '98o r 2000'.

4.3 SYSTEM DOCUMENTATION

Documentation is virtually important if a program is to be understood, not only by others, but also by the original programmer at a later stage. So any program should be documented while being written. That is, the program itself should contain comments or remarks stating exactly what various parts of the program are to do. This technical documentation is of utmost importance to the programmer (or other programmers who are with the program) when the time comes to debug, modify or update the program. As a result of this, the program was well documented. The following are also part of the system documentation.

1. System/program flowchart (appendix A)
2. Program input and output design (appendix B)
3. Hardware and software specification (section 3)
4. Program Listing (appendix C)

4.4 HARDWARE AND SOFTWARE REQUIREMENT

To make maximum utilization of the developed system, certain hardware and software needs to be installed.

4.4.1 HARDWARE REQUIREMENT

This comprises of all the physical component of the computer and its accessories. Therefore the choice of the computer requirement is done to suit both the current and the future needs of the organization with respect to the volume and types of data to be processed. In summary, a computer system with the following minimum requirement is required.

4.4.2 SOFTWARE REQUIREMENTS

Software requirement are the basic and other relevant application software that is needed to be installed on the computer in order to make maximum utilization of the computer system and the developed Accounting software. It is because of this fact that the following software must be installed on the system.

However, other application software apart from the ones mentioned above may be installed as may be required by the staff and organization concerned.

4.5 PROJECT MODULES AND TASK PERFORMED

S/No	Project modules	Task performed
1	Components	This is used to list and describe all the components of the computer system

2	Gen-trouble-Rule	This state and discuss the general rules used ion troubleshooting and repair of a pc.
3	Trouble steps	This is used to enumerate various steps taking when troubleshooting of pc.
4	Diagnostic signals	State and explain diagnostic signals and their probable causes
5	Error codes	This enumerates error codes of different components/parts of computer system. Together with the description of the error codes.
6	Adoptable measure	This gives an export measure to adopt for a healthy pc.
7	Entry	This takes the status and details complaint fault, customer and operation to be performed on the machine.
5	Report	This is used to display the operation performed on the machine.

USERS MANUAL

The software system is user-friendly. Software is to said to be user friendly if its human users find it easy to use. The use of user interface such as window is an important component of user friendliness enhances this.

However, the user still needs a written manual that guides him/her through the software system so that appropriate input should be supplied to the software system for the system to work effetely.

Below is the guideline to the software system

USING THE SYSTEM

The software system could be run on the hard disk by installing the software on the hard disk.

4.6 PROGRAM INSTALLATION

This is the process of transferring the developed programme/software from the floppy disk to a permanent storage device called the hard disk.

However, due to vast improvement in recent operating systems, the installation of the software is a very simple operation. The procedure is as follows:

STEPS	PROCEDURE	RESULT
1. Go to start	Click	Start popup menu is displayed.
2. Run submenu	Click	Run dialog box displayed
3. Insert TR – GUIDE diskette into A:\	-	-
4. Type the source drive (A:\)	Browse	Content of A:\ displayed
5. Select setup	Double – click	Installation begins
6. Follow the instruction that Follows	Click	Installation in progress
7. Destination	-	Software installed C:\ successfully

BRINGING THE SOFTWARE FROM C:\ TO WINDOW PROGRAM SUBMENU

STEPS	PROCEDURE	RESULT
1. Go to start	Click	Start popup menu displayed
2. Setting submenu	Click	-
3. Choose taskbar & start menu	Click	Taskbar dialog box
4. Select start menu program	click	-
5. Browse to select TR – GUIDE software	Double - click	-
6. Select folder (program)	Click	-
7. Destination	click	TR – GUIDE software copied into program submenu successfully.

At the end of the installation, the floppy diskette becomes a backup and should be well safeguarded against any damages for future use.

To start the system after successful installation of the TR – GUIDE software, all one need to do is to start menu and run the folder name you have created in the program submenu.

This will invoke introductory screen, which will ask the user to enter "password" after which a screen introducing the title of the program has been displayed and left after sometime. If this is done, there appears the main menu, which has 7 items.

(i) Components
(ii) Troubleshooting
(iii) Diagnostic signals
(iv) Error codes
(v) Adoptable measure document
(vi) Quit

The user is allowed to move up and down arrow keys to select any of the items. The functions of each item are stated below:

(i) Components: This is the list that describes all the component of computer system.
(ii) Troubleshooting: this has two submenu which are – a) General troubleshooting rules used in troubleshooting and repair of a computer system. b) Steps for troubleshooting, which state the various steps taking when troubleshooting a PC.
(iii) Diagnostics signals and probable causes: Which explain and state diagnostics signals and the probable causes of these signals of PC.

(iv) Error codes: This enumerate error codes of different components / parts of computer system together with the description of the error code.

(v) Adoptable measure: This gives an expert measure to adopt for a healthy PC.

(vi) Documentation: This has submenu viz: a), Entry – This takes the status and details of and how the work is done, complains, nature of fault, customer and success operation performed on machine form from user. b), Report – This displays the list of machine that has undergone troubleshooting. It displays on the screen the complaint' name, machine number, date the computer was brought in for repair, date released and final status of the machine.

It also has printing as icon, which is used to print the report by sending it to printer.

The main menu of the TR – GUIDE as described above is in form of rectangular square containing pop up or pulls down menu option. Below is the structure of the main menu.

4.7 PROGRAM LAYOUTS AND DESIGN:

The general system design includes the flowchart (procedure), a written explanation (pseudocode). For this study, coding (programming) is used to depict the representation of the project. This is available in the project Appendix C, while the system flowchart is at A, and the output forms are in Appendix B.

4.7.1 INPUT DESIGN:

Input design, which is also known as file design, is the denominator of any system. It contains the raw material (Data) necessary to produce output.

In manufacturing, for example, it is only when the product to be made is decided upon that the raw material could be specified and ordered. In the process of developing an Information system, output requirement are first decided upon before data are specified. In a sense, output requirement can be thought of as an input database design.

Therefore, the data file design includes the entire database files used throughout the system for the proper storage of the course processing data.

Basic input to be used in the system are as follows:

FIELD	FIELD NAME	FIELD TYPE	WIDTH	**DECIMAL PLACES**
1.	TOPIC	Character	50	-
2.	Prob_Desc	Memo	4	-
3.	Who	Character	25	-
4.	When	Date	8	-
5.	How	Character	10	-
6.	Related	Logical	1	-
7.	Complaint	Memo	4	-
8.	Where	Character	20	-
9.	Periodic	Logical	1	-
10.	Components	Memo	4	-
11.	Observation	Memo	4	-
12.	Site action	Memo	4	-
13.	Date_in	Date	8	-
14.	Result	Memo	4	-
15.	Date_out	Date	8	-
16.	Prob_type	Character	10	-
17.	Code	Numeric	4	-
18.	Shop_action	Memo	4	-

4.7.2 OUTPUT DESIGN:

The output/report generated by the system includes:

(a) List of all successful operations on the computer

(b) Suggested reasons for the problem of the machine together with the observation

(c) The different components of a PC system

(d) The description of problem as reported by the client

(e) The fault as diagnosed by the engineer

(f) Report on general troubleshooting guidelines.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 RECOMMENDATION

What is the essential difference between existing computer science and what is called **ARTIFICIAL INTELLIGENCE?**

What is the basis for the widespread interest in the subject today. These are different question, which may be asked by anyone considering an investment of resources in developing or incorporating A. I technology.

A primary feature of most A. I application is an attempt to utilize formal and informal human knowledge in a structure way.

The expected potential of a technology is based on increase. The availability and usefulness of expert knowledge. It is this potential which will affect many diverse areas; Education, Commerce, Government, Medicine, Manufacturing, Engineering.

Additionally, this technology is being used to enhance the understanding of human through process and is facilitating the generation of new knowledge.

Currently, research into A I have found no based in Nigeria. No known move to start one has been initiated.

Expert system are presently being used along side most human expert in developed countries performing different industrial operations. Research into expert system and A. I. in general should be initiated immediately.

It is therefore recommended that the instruction and expert advice given in this system TR—GUIDE should be adhered to for better performance.

5.2 CONCLUSION

Like MYCIN, PROSPECTOR e.t.c. Expert system still undergoes some process of system overhaul, knowledge elicitation reconstruction, further and analysis and refinement.

TR—GUIDE would also undergo these processes, as further enhancement could still be tenable. Several surprising tends may occur in the types of Artificial intelligence technology that will be developed in the near future. There will be a rapid growing use of small and imbedded. A.I based system unconventional software programs.

This system will improve the utility of large conventional programs and reduce the training requirements of their use.

Conclusively, the PC troubleshooting and repair guide has been computerized. It has successfully been designed, programmed, tested and debugged. This system has many features that make it a useful tool.

The features are;

[1] Ease to use.

[2] It is flexible—it accommodate modifications

[3] Speed of processing

The high level of operating efficiency, which has been incorporated into the new system, has helped immensely to induce quick repair decision and it can be upgraded to accommodate more error messages.

REFERENCES

1.Anne Taussig (Editor) Microsoft Visual FoxPro 6.0, programmer's Guide; Microsoft press, one Microsoft way, Redmond, Washington, 1998.

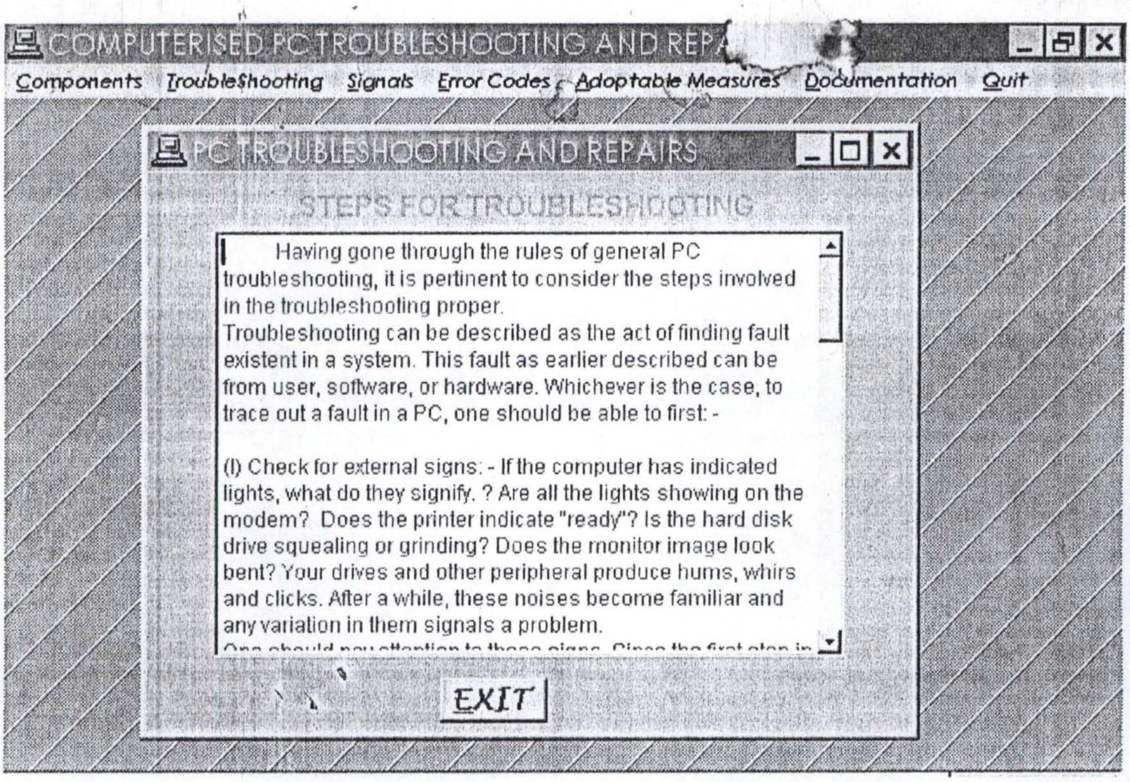
2.Badmus Artificial Intelligent, Lecture notes Federal University of Technology, Minna (Unpublished).

3.Barry press DIG PC Upgrade and Repair Bible, profession Edition, Books worldwide inc., An International Data Group Company. 1998

4.Bramer Max (Editor) Practical Experience in Building Expert system, chich ester; John Wiley and sons, 1990

5.Ozichi Nweke Practical Expert system on minerals/oil Exploration (Min-OEES); federal University of technology, Minna, 1997

6.Scott Mueller Upgrading and repairing PC, 11th Edition, Que, 201 west 103rd street, Indiana 46290, 1999.


COMPUTERISED PC TROUBLESHOOTING AND REP
Components TroubleShooting Signals Error Codes Adoptable Measures Documentation Quit
PC TROUBLESHOOTING AND REPAIRS
EXIT

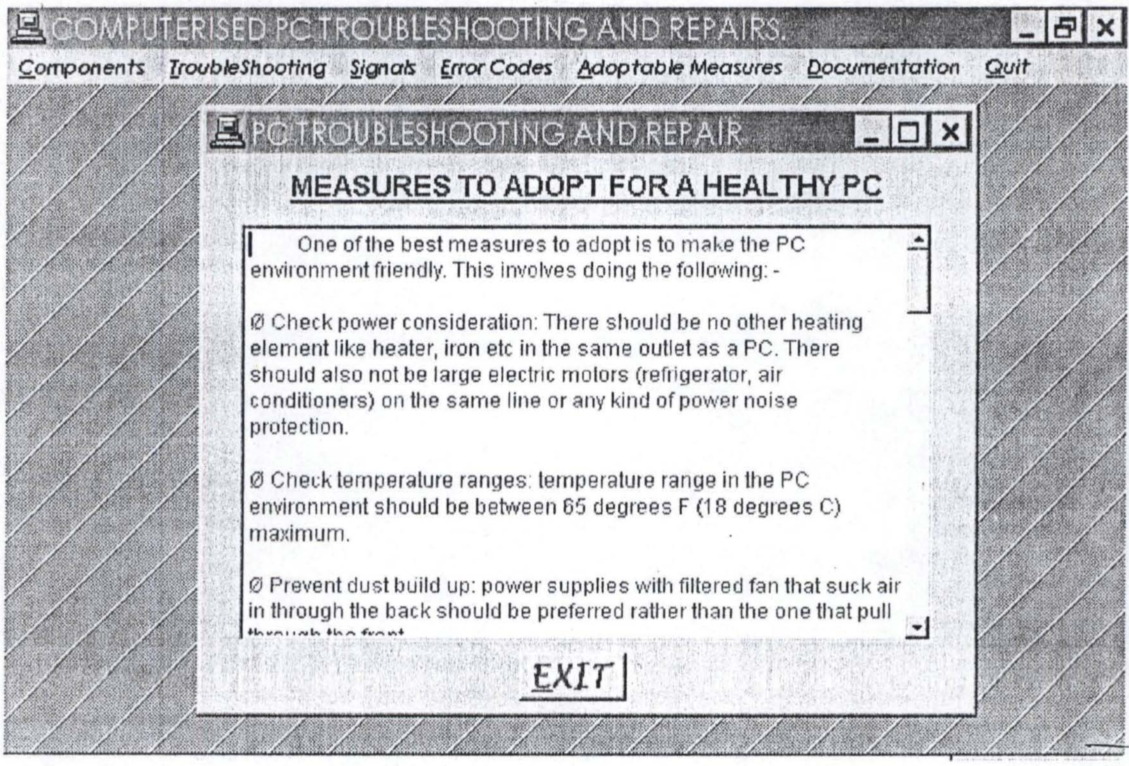
STEPS FOR TROUBLESHOOTING

Having gone through the rules of general PC troubleshooting, it is pertinent to consider the steps involved in the troubleshooting proper.

Troubleshooting can be described as the act of finding fault existent in a system. This fault as earlier described can be from user, software, or hardware. Whichever is the case, to trace out a fault in a PC, one should be able to first: -

(I) Check for external signs: - If the computer has indicated lights, what do they signify. ? Are all the lights showing on the modem? Does the printer indicate "ready"? Is the hard disk drive squealing or grinding? Does the monitor image look bent? Your drives and other peripheral produce hums, whirs and clicks. After a while, these noises become familiar and any variation in them signals a problem.

One should pay attention to these signs. Since the first step in


COMPUTERISED PC TROUBLESHOOTING AND REPAIRS.
Components TroubleShooting Signals Error Codes Adoptable Measures Documentation Quit
PC TROUBLESHOOTING AND REPAIR
EXIT

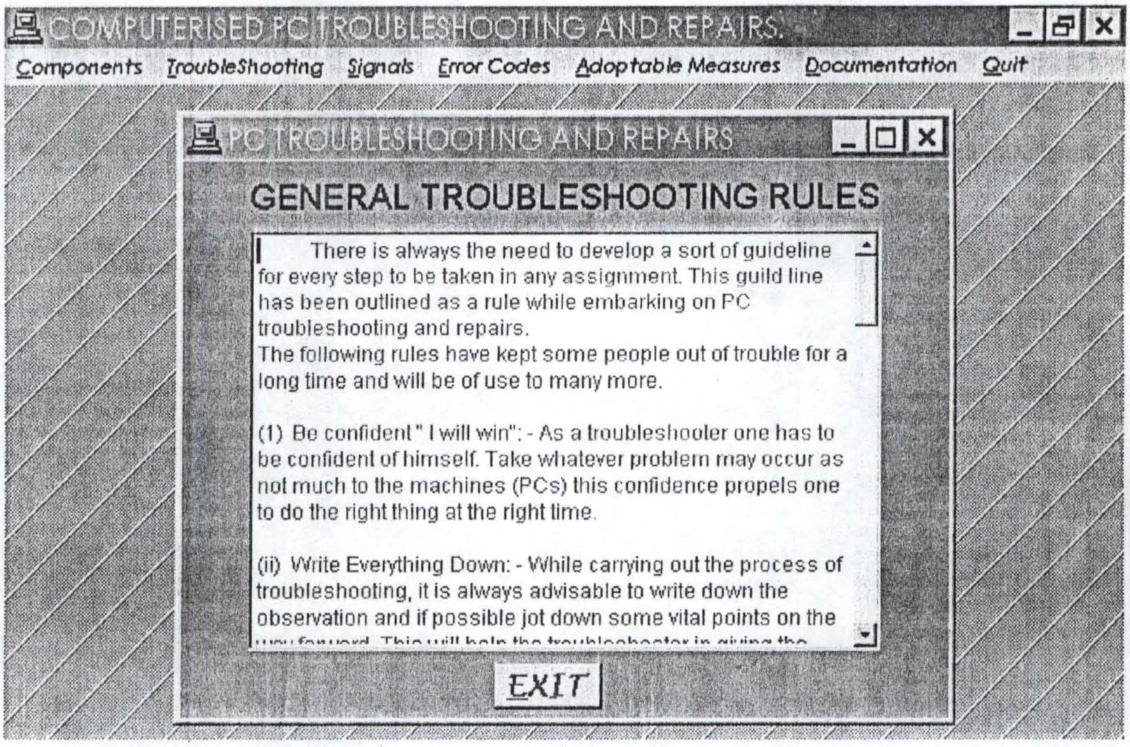
MEASURES TO ADOPT FOR A HEALTHY PC

One of the best measures to adopt is to make the PC environment friendly. This involves doing the following: -

Ø Check power consideration: There should be no other heating element like heater, iron etc in the same outlet as a PC. There should also not be large electric motors (refrigerator, air conditioners) on the same line or any kind of power noise protection.

Ø Check temperature ranges: temperature range in the PC environment should be between 65 degrees F (18 degrees C) maximum.

Ø Prevent dust build up: power supplies with filtered fan that suck air in through the back should be preferred rather than the one that pull through the front

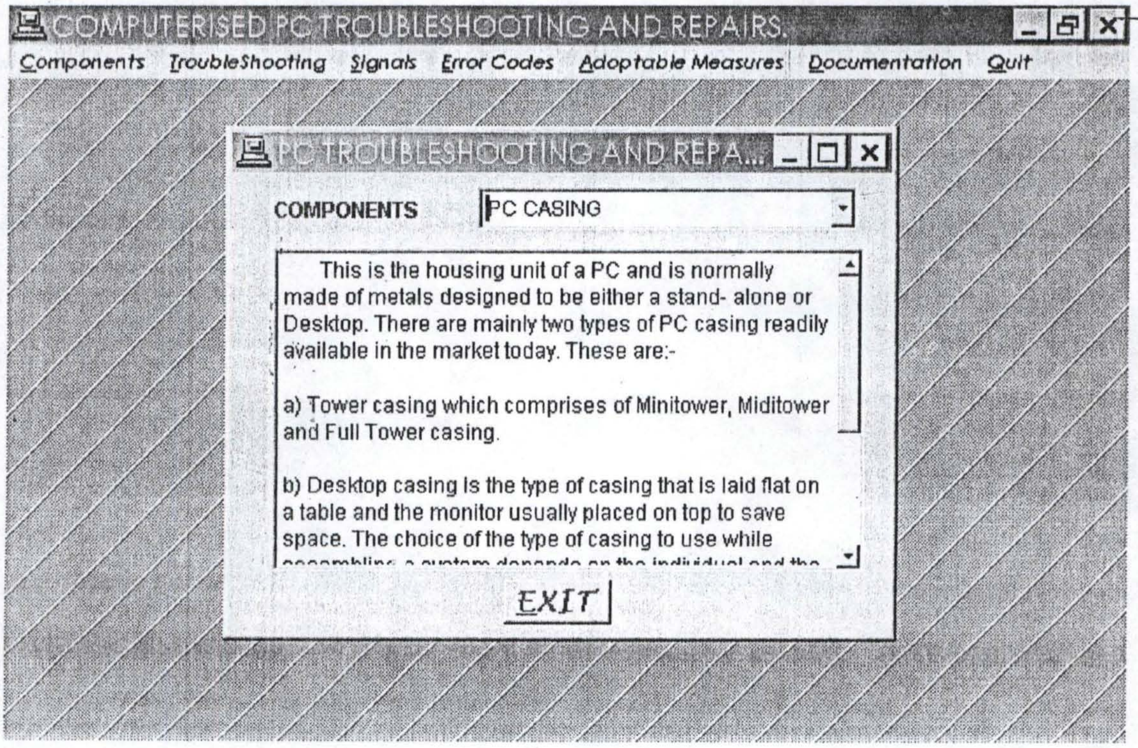


GENERAL TROUBLESHOOTING RULES

There is always the need to develop a sort of guideline for every step to be taken in any assignment. This guild line has been outlined as a rule while embarking on PC troubleshooting and repairs.
The following rules have kept some people out of trouble for a long time and will be of use to many more.

(1) Be confident " I will win": - As a troubleshooter one has to be confident of himself. Take whatever problem may occur as not much to the machines (PCs) this confidence propels one to do the right thing at the right time.

(ii) Write Everything Down: - While carrying out the process of troubleshooting, it is always advisable to write down the observation and if possible jot down some vital points on the way forward. This will help the troubleshooter in giving the

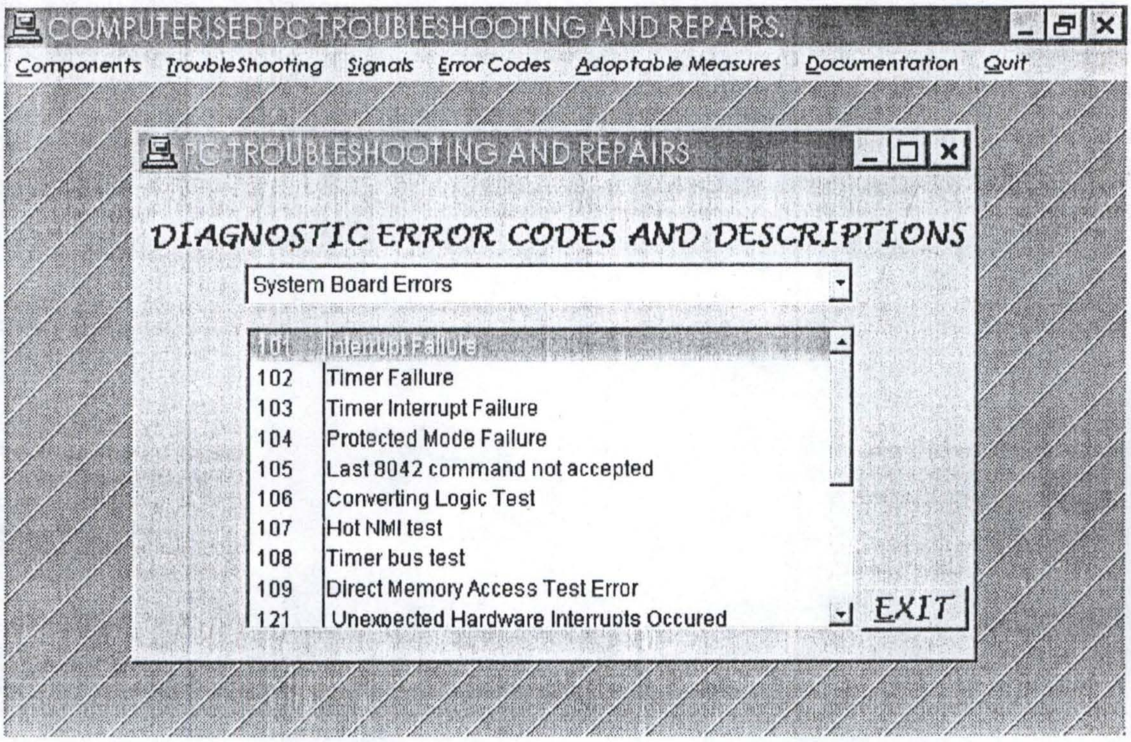


COMPONENTS PC CASING

This is the housing unit of a PC and is normally made of metals designed to be either a stand- alone or Desktop. There are mainly two types of PC casing readily available in the market today. These are:-

a) Tower casing which comprises of Minitower, Miditower and Full Tower casing.

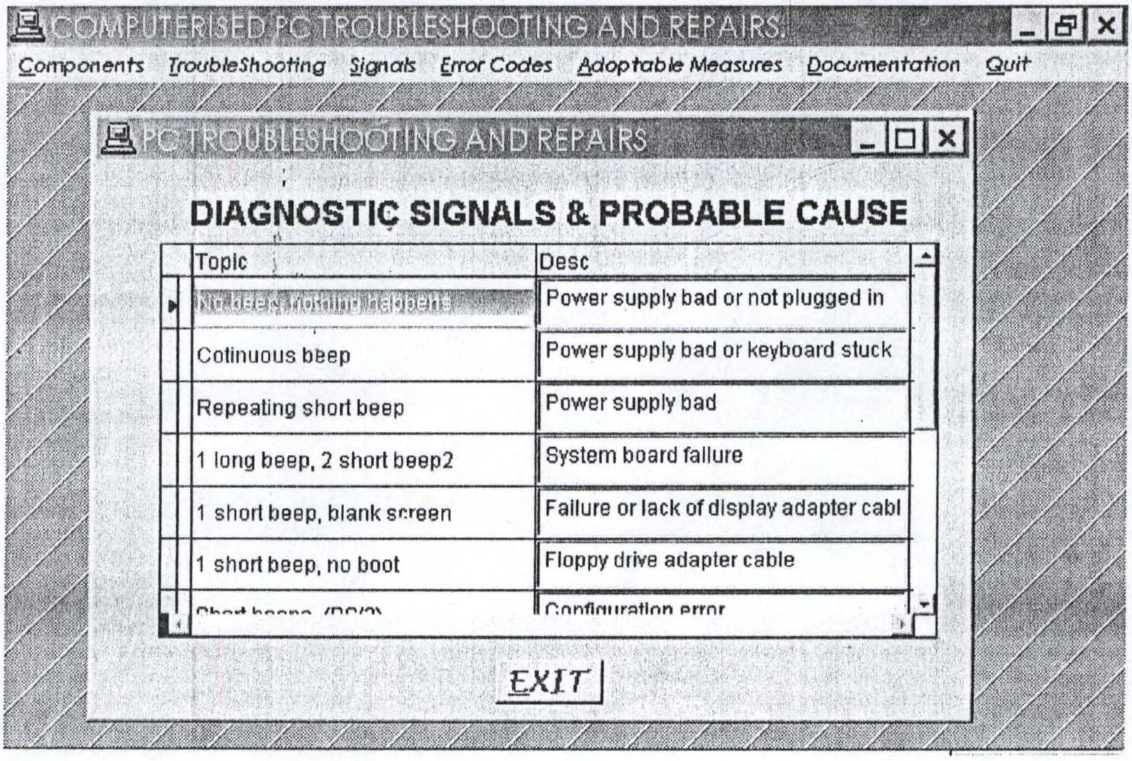
b) Desktop casing is the type of casing that is laid flat on a table and the monitor usually placed on top to save space. The choice of the type of casing to use while assembling a system depends on the individual and the



DIAGNOSTIC ERROR CODES AND DESCRIPTIONS

System Board Errors

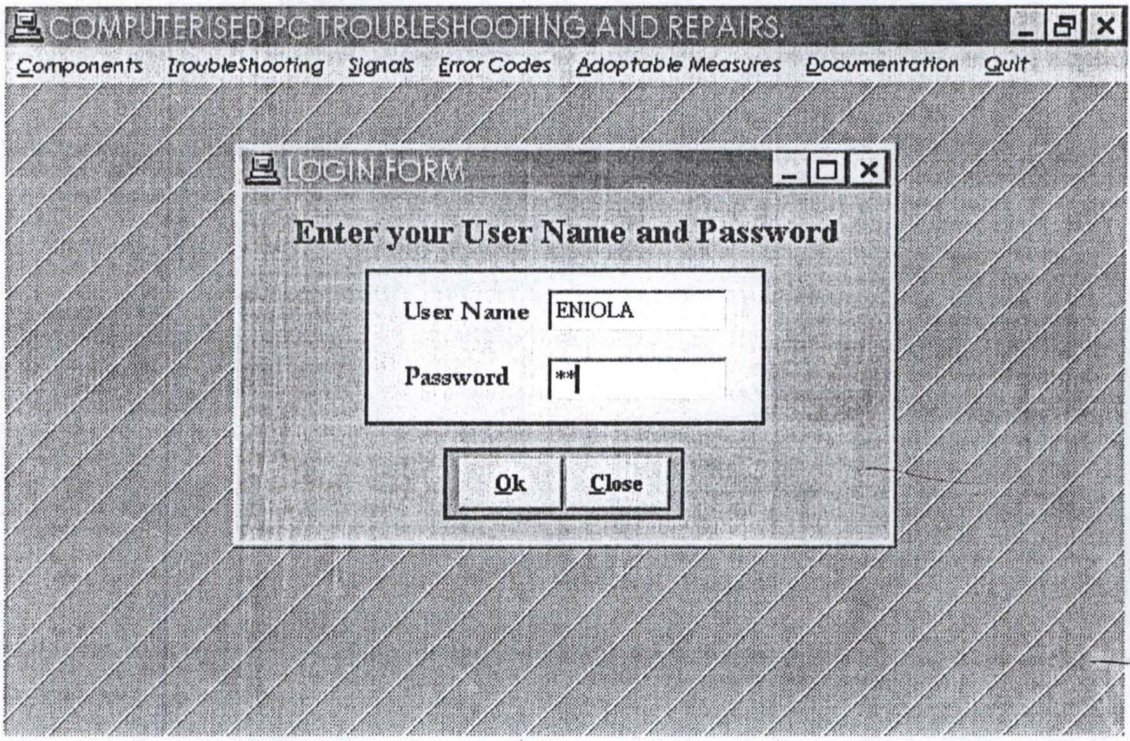
101	Interrupt Failure
102	Timer Failure
103	Timer Interrupt Failure
104	Protected Mode Failure
105	Last 8042 command not accepted
106	Converting Logic Test
107	Hot NMI test
108	Timer bus test
109	Direct Memory Access Test Error
121	Unexpected Hardware Interrupts Occured



DIAGNOSTIC SIGNALS & PROBABLE CAUSE

Topic	Desc
No beep, nothing happens	Power supply bad or not plugged in
Cotinuous beep	Power supply bad or keyboard stuck
Repeating short beep	Power supply bad
1 long beep, 2 short beep2	System board failure
1 short beep, blank screen	Failure or lack of display adapter cabl
1 short beep, no boot	Floppy drive adapter cable
Short beeps (PS/2)	Configuration error

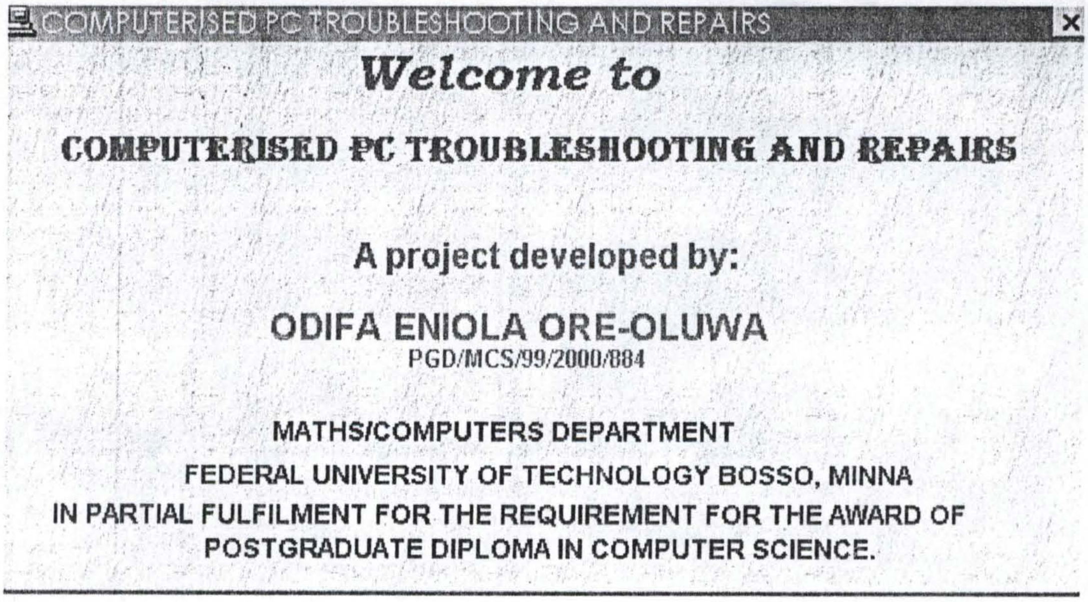
SRUCTURE OF THE MAIN MENU & THE INTRODUCTORY SCREEN.


COMPUTERISED PC TROUBLESHOOTING AND REPAIRS.
Components TroubleShooting Signals Error Codes Adoptable Measures Documentation Quit
LOGIN FORM
Ok
Close

Enter your User Name and Password

User Name ENIOLA

Password **



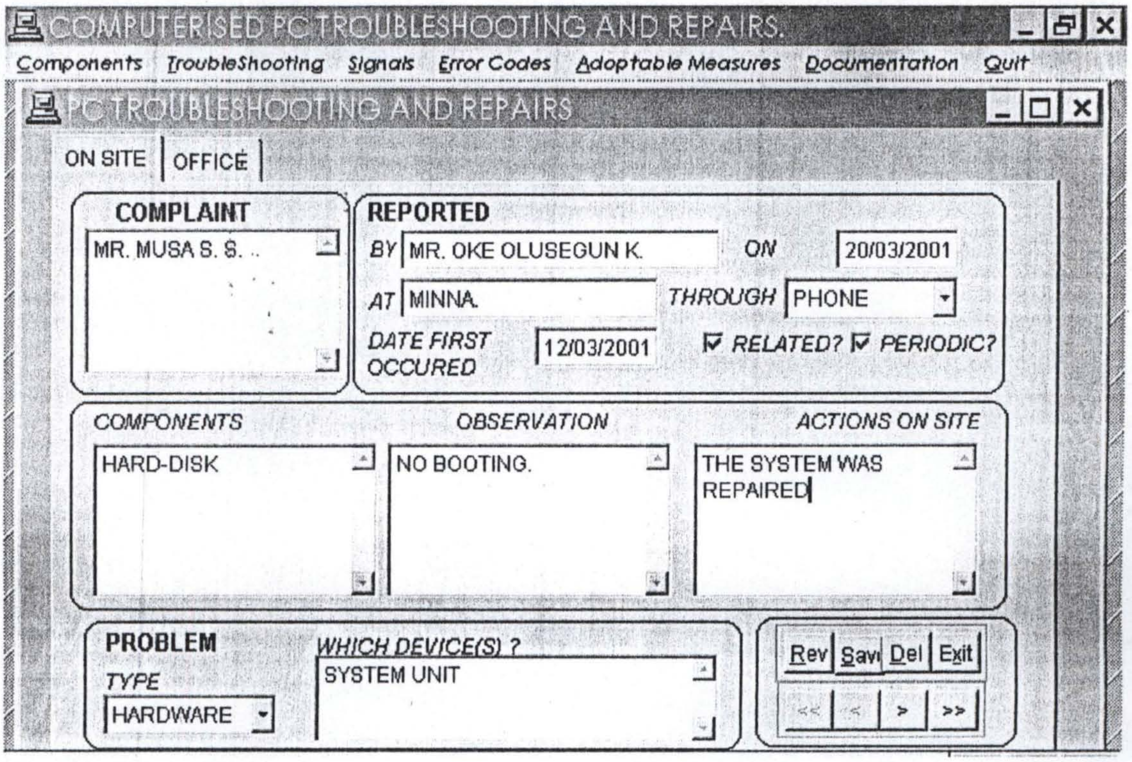
Welcome to

COMPUTERISED PC TROUBLESHOOTING AND REPAIRS

A project developed by:

ODIFA ENIOLA ORE-OLUWA
PGD/MCS/99/2000/884

MATHS/COMPUTERS DEPARTMENT
FEDERAL UNIVERSITY OF TECHNOLOGY BOSSO, MINNA
IN PARTIAL FULFILMENT FOR THE REQUIREMENT FOR THE AWARD OF
POSTGRADUATE DIPLOMA IN COMPUTER SCIENCE.


COMPUTERISED PC TROUBLESHOOTING AND REPAIRS.
Components TroubleShooting Signals Error Codes Adoptable Measures Documentation Quit
PC TROUBLESHOOTING AND REPAIRS
ON SITE
OFFICE
Rev Savi Del Exit
<< < > >>

COMPLAINT

MR. MUSA S. S.

REPORTED

BY MR. OKE OLUSEGUN K. ON 20/03/2001

AT MINNA. THROUGH PHONE

DATE FIRST OCCURED 12/03/2001 ☑ RELATED? ☑ PERIODIC?

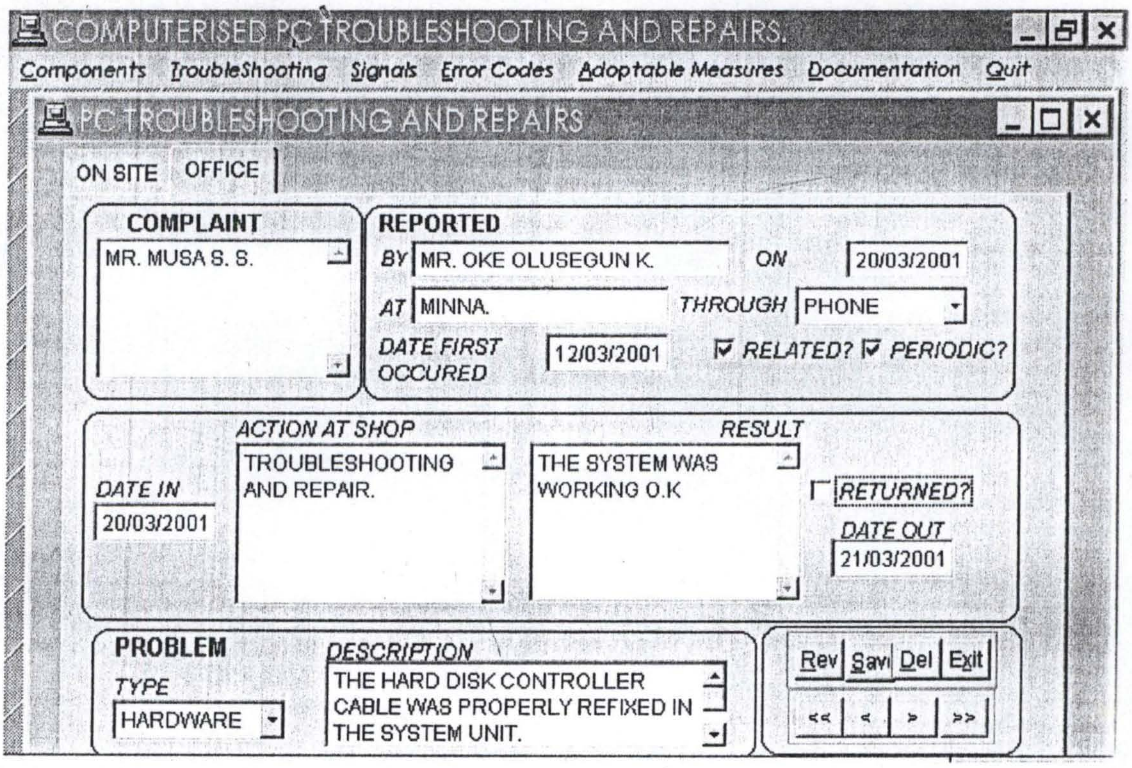
COMPONENTS	OBSERVATION	ACTIONS ON SITE
HARD-DISK	NO BOOTING.	THE SYSTEM WAS REPAIRED

PROBLEM

TYPE HARDWARE

WHICH DEVICE(S) ?

SYSTEM UNIT


COMPUTERISED PC TROUBLESHOOTING AND REPAIRS.
Components TroubleShooting Signals Error Codes Adoptable Measures Documentation Quit
PC TROUBLESHOOTING AND REPAIRS
ON SITE
OFFICE
Rev Savi Del Exit
<< < > >>

COMPLAINT

MR. MUSA S. S.

REPORTED

BY MR. OKE OLUSEGUN K. ON 20/03/2001

AT MINNA. THROUGH PHONE

DATE FIRST OCCURED 12/03/2001 ☑ RELATED? ☑ PERIODIC?

DATE IN 20/03/2001

ACTION AT SHOP

TROUBLESHOOTING AND REPAIR.

RESULT

THE SYSTEM WAS WORKING O.K

☐ RETURNED?

DATE OUT 21/03/2001

PROBLEM

TYPE HARDWARE

DESCRIPTION

THE HARD DISK CONTROLLER CABLE WAS PROPERLY REFIXED IN THE SYSTEM UNIT.

TROUBLE DOCUMENTATION

18/03/2001

ROBLEM TYPE: **HARDWARE**

USTOMER'S COMPLAINT

MR. MUSA S. S.

PROBLEM DESCRIPTION

THE HARD DISK CONTROLLER CABLE WAS PROPERLY REFIXED IN THE SYSTEM UNIT.

ORTED

BY	MR. OKE OLUSEGUN K.
DATE	20/03/2
THROUGH	PHONE
PERIODIC?	Y

RESULT

THE SYSTEM WAS WORKING O.K

ULTY COMPONENTS

HARD-DISK

OBSERVATION

NO BOOTING.

```
*********************
*Menu Section
*********************
set device to screen
clear
*RUN \projinfo\info.EXE
SET SYSMENU OFF
CLOSE ALL
CLEAR
set safety off
set color to w/b
set exclusive on
set date to brit
set century on
set status off
set talk off


clear
set path to \ENIOLA
PUBLIC MY, myear
store space(9) to myear

_screen.icon = "PC04.ico"
_screen.caption = "COMPUTERISED PC TROUBLESHOOTING AND REPAIRS."
_screen.picture = "wizflax.bmp"


my =0
do form frmlogin

sele 1
use tchap1 again exclusive
sele 2
use tchap21 again exclusive
sele 3
use tchap22 again exclusive
sele 4
use tchap3 again exclusive
sele 5
use tchap32 again exclusive
sele 6
use tchap34 again exclusive
sele 7
use tchap35 again exclusive
sele 1
```

```
do form ffirst
do mainmenu.mpr
read events
return

PUBLIC oform1

oform1=NEWOBJECT("form1")
oform1.Show
RETURN

**************************************************
*-- Form:        form1 (c:\eniola\fchap32.scx)
*-- ParentClass:  form
*-- BaseClass:   form
*
DEFINE CLASS form1 AS form

        Top = 11
        Left = 11
        Height = 372
        Width = 617
        DoCreate = .T.
        Caption = "PC TROUBLESHOOTING AND REPAIRS"
        Icon = "pc04.ico"
        BackColor = RGB(115,185,185)
        Name = "Form1"

        ADD OBJECT pageframe1 AS pageframe WITH ;
                ErasePage = .T., ;
                PageCount = 2, ;
                TabStyle = 1, ;
                Top = 2, ;
                Left = 17, ;
                Width = 576, ;
                Height = 358, ;
                Name = "Pageframe1", ;
                Page1.Caption = "ON SITE", ;
                Page1.Name = "Page1", ;
                Page2.Caption = "OFFICE", ;
                Page2.Name = "Page2"


        ADD OBJECT form1.pageframe1.page1.shape4 AS shape WITH ;
                Top = 122, ;
                Left = 10, ;
        Height = 119, ;
        Width = 535, ;
        Curvature = 20, ;
        Name = "Shape4"

ADD OBJECT form1.pageframe1.page1.shape1 AS shape WITH ;
        Top = 6, ;
        Left = 171, ;
        Height = 112, ;
        Width = 372, ;
        Curvature = 20, ;
        Name = "Shape1"


ADD OBJECT form1.pageframe1.page1.shape2 AS shape WITH ;
        Top = 5, ;
        Left = 10, ;
        Height = 113, ;
        Width = 159, ;
        Curvature = 20, ;
        Name = "Shape2"

ADD OBJECT form1.pageframe1.page1.shape3 AS shape WITH ;
        Top = 247, ;
        Left = 18, ;
        Height = 74, ;
        Width = 376, ;
        Curvature = 20, ;
        Name = "Shape3"

ADD OBJECT form1.pageframe1.page1.edtcomponents AS editbox WITH ;
        Comment = "", ;
        Height = 88, ;
        Left = 23, ;
        TabIndex = 2, ;
        Top = 146, ;
        Width = 163, ;
        ControlSource = "tchap32.components", ;
        Name = "edtComponents"


ADD OBJECT form1.pageframe1.page1.lblcomponents AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "COMPONENTS", ;
        Left = 23, ;
        Top = 126, ;
        Width = 85, ;
        TabIndex = 1, ;
        Name = "lblComponents"


ADD OBJECT form1.pageframe1.page1.edtobservation AS editbox WITH ;
        Comment = "", ;
        Height = 88, ;
        Left = 191, ;
        TabIndex = 4, ;
        Top = 146, ;
        Width = 163, ;
        ControlSource = "tchap32.observation", ;
        Name = "edtObservation"


ADD OBJECT form1.pageframe1.page1.lblobservation AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "OBSERVATION", ;
        Left = 230, ;
        Top = 126, ;
        Width = 87, ;
        TabIndex = 3, ;
        Name = "lblObservation"


ADD OBJECT form1.pageframe1.page1.edtsiteaction AS editbox WITH ;
        Comment = "", ;
        Height = 88, ;
        Left = 366, ;
        TabIndex = 6, ;
        Top = 146, ;
        Width = 163, ;
        ControlSource = "tchap32.siteaction", ;
        Name = "edtSiteaction"


ADD OBJECT form1.pageframe1.page1.lblsiteaction AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "ACTIONS ON SITE", ;
        Height = 17, ;
        Left = 424, ;
        Top = 126, ;
        Width = 105, ;
        TabIndex = 5, ;
        Name = "lblSiteaction"


ADD OBJECT form1.pageframe1.page1.shape5 AS shape WITH ;
        Top = 245, ;
        Left = 402, ;
        Height = 74, ;
        Width = 149, ;
        Curvature = 20, ;
        Name = "Shape5"


ADD OBJECT form1.pageframe1.page2.shape1 AS shape WITH ;
        Top = 6, ;
        Left = 171, ;
        Height = 112, ;
        Width = 372, ;
        Curvature = 20, ;
        Name = "Shape1"


ADD OBJECT form1.pageframe1.page2.shape2 AS shape WITH ;
        Top = 5, ;
        Left = 11, ;
        Height = 113, ;
        Width = 159, ;
        Curvature = 20, ;
        Name = "Shape2"


ADD OBJECT form1.pageframe1.page2.shape3 AS shape WITH ;
        Top = 249, ;
        Left = 17, ;
        Height = 74, ;
        Width = 379, ;
        Curvature = 20, ;
        Name = "Shape3"
```

```
ADD OBJECT form1.pageframe1.page2.lblshopaction AS label WITH ;
        AutoSize = .F., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "ACTION AT SHOP", ;
        Height = 17, ;
        Left = 98, ;
        Top = 127, ;
        Width = 108, ;
        TabIndex = 3, ;
        Name = "lblShopaction"


ADD OBJECT form1.pageframe1.page2.edtresult AS editbox WITH ;
        Comment = "", ;
        Height = 91, ;
        Left = 266, ;
        TabIndex = 6, ;
        Top = 142, ;
        Width = 156, ;
        ControlSource = "tchap32.result", ;
        Name = "edtResult"


ADD OBJECT form1.pageframe1.page2.lblresult AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "RESULT", ;
        Left = 374, ;
        Top = 127, ;
        Width = 48, ;
        TabIndex = 5, ;
        Name = "lblResult"


ADD OBJECT form1.pageframe1.page2.txtdate_out AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.date_out", ;
        Height = 23, ;
        Left = [illegible], ;
        TabIndex = 10, ;
        Top = 196, ;
        Width = 71, ;
        Name = "txtDate_out"


ADD OBJECT form1.pageframe1.page2.lbldate_out AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "DATE OUT", ;
        Height = 17, ;
        Left = 443, ;
        Top = 184, ;
        Width = 60, ;
        TabIndex = 9, ;
        Name = "lblDate_out"


ADD OBJECT form1.pageframe1.page2.chkreturned AS checkbox WITH ;
        Comment = "", ;
        Top = 161, ;
        Left = 426, ;
        Height = 17, ;
        Width = 93, ;
        FontItalic = .T., ;
        Caption = "RETURNED?", ;
        ControlSource = "tchap32.returned", ;
        Name = "chkReturned"


ADD OBJECT form1.pageframe1.page2.shape5 AS shape WITH ;
        Top = 246, ;
        Left = 399, ;
        Height = 74, ;
        Width = 149, ;
        Curvature = 20, ;
        Name = "Shape5"


ADD OBJECT txtwho AS textbox WITH ;
        Comment = "", ;
        ControlSource = "tchap32.who", ;
        Height = 23, ;
        Left = 216, ;
        MaxLength = 25, ;
        TabIndex = 3, ;
        Top = 56, ;
        Width = 183, ;
        Name = "txtWho"


ADD OBJECT lblwho AS label WITH ;
        AutoSize = .T., ;
        FontItalic = .T., ;
        WordWrap = .T., ;
        BackStyle = 0, ;
        Caption = "BY", ;
        Left = 198, ;
        Top = 59, ;
        Width = 16, ;
        TabIndex = 2, ;
        Name = "lblWho"


ADD OBJECT cmdgrpnavigator AS commandgroup WITH ;
        AutoSize = .T., ;
        ButtonCount = 4, ;
        BackStyle = 1, ;
        Value = 1, ;
        Height = 33, ;
        Left = 432, ;
        Top = 312, ;
        Width = 120, ;
        TabIndex = 8, ;
        BackColor = RGB(192,192,192), ;
        Name = "cmdgrpnavigator", ;
        Command1.AutoSize = .F., ;
        Command1.Top = 5, ;
        Command1.Left = 5, ;
        Command1.Height = 23, ;
        Command1.Width = 27, ;
        Command1.Caption = "<<", ;
        Command1.Name = "cmdtop", ;
        Command2.AutoSize = .F., ;
        Command2.Top = 5, ;
        Command2.Left = 32, ;
        Command2.Height = 23, ;
        Command2.Width = 27, ;
        Command2.Caption = "<", ;
        Command2.Name = "cmdprevious", ;
        Command3.AutoSize = .F., ;
        Command3.Top = 5, ;
        Command3.Left = 59, ;
```

```
                Command3.Height = 23, ;
                Command3.Width = 27, ;
                Command3.Caption = ">", ;
                Command3.Name = "cmdnext", ;
                Command4.AutoSize = .F., ;
                Command4.Top = 5, ;
                Command4.Left = 88, ;
                Command4.Height = 23, ;
                Command4.Width = 27, ;
                Command4.Caption = ">>", ;
                Command4.Name = "cmdbottom"


        PROCEDURE Activate
                Thisform.setall("Readonly",.T.,"Textbox")
                Thisform.setall("Readonly",.T.,"Editbox")
                thisform.cmdgrpnavigator.enabled = .T.


                thisform.cmdgrpeditor.cmdadd.visible = .T.
                thisform.cmdgrpeditor.cmdmodify.visible = .T.
                thisform.cmdgrpeditor.cmdrevert.visible = .F.
                thisform.cmdgrpeditor.cmdsave.visible = .F.
                thisform.cmdgrpeditor.cmdsave.enabled = .F.

                thisform.cmdgrpeditor.cmdadd.setfocus
        ENDPROC


        PROCEDURE Load
                PUBLIC modified
                MODIFIED = .f.

                public Mwho ,mwhen ,mwhere,mhow ,mwhenfirst ,mrelated ,mperiodic,;

mprob_type,mprob_desc,mcomplaint,mcomponents,mobservation,;

msiteaction,mdate_in,mdate_out,mshopaction,mresult,mreturned

        ENDPROC


        PROCEDURE chkreturned.Click
                if this.value == .f.
                        thisform.txtdate_out.enabled = .f.
                else
                thisform.txtdate_out.enabled = .t.
        endif
    ENDPROC


    PROCEDURE cboprob_type.InteractiveChange
        thisvalue = allt(this.value)
        if thisvalue == 'HARDWARE'
                Thisform.lblprob_desc.caption = 'WHICH DEVICE(S) ?'
        Endif

        if thisvalue == 'SOFTWARE'
                Thisform.lblprob_desc.caption = 'WHICH
SOFTWARE/PACKAGE(S) ?'
        Endif

        if thisvalue == 'HUMANWARE'
                Thisform.lblprob_desc.caption = 'HOW DID IT HAPPEN ?'
        Endif
    ENDPROC


    PROCEDURE cmdgrpeditor.cmdadd.Click
        *sele proj1
        APPEND BLANK

        modified = .f.
        Thisform.setall("Readonly",.F.,"Textbox")
        Thisform.setall("Readonly",.F.,"combobox")
        Thisform.setall("Readonly",.F.,"EDITbox")
        *thisform.txtyear.readonly = .T.
        thisform.cmdgrpnavigator.enabled = .F.

        thisform.cmdgrpeditor.cmdadd.visible = .F.
        thisform.cmdgrpeditor.cmdmodify.visible = .F.
        thisform.cmdgrpeditor.cmdrevert.visible = .T.
        thisform.cmdgrpeditor.cmdsave.visible = .T.
        thisform.cmdgrpeditor.cmdsave.enabled = .T.

        *
        THISFORM.REFRESH()
    ENDPROC


    PROCEDURE cmdgrpeditor.cmdsave.Click
                        replace who with thisform.txtwho.value
                replace when with thisform.txtwhen.value
            replace where with thisform.txtwhere.value
                replace how with thisform.cbohow.value
                replace whenfirst with thisform.txtwhenfirst.value

                replace related with thisform.chkrelated.value
                replace periodic with thisform.chkperiodic.value
                replace prob_type with thisform.cboprob_type.value
                replace prob_desc with thisform.edtprob_desc.value
            replace complaint with thisform.edtcomplaint.value

            replace components with
thisform.pageframe1.page1.edtcomponents.value
            replace observation with
thisform.pageframe1.page1.edtobservation.value
            replace siteaction with thisform.pageframe1.page1.edtsiteaction.value

            replace date_in with thisform.pageframe1.page2.txtdate_in.value
            replace date_out with thisform.pageframe1.page2.txtdate_out.value
            replace shopaction with
thisform.pageframe1.page2.edtshopaction.value
            replace result with thisform.pageframe1.page2.edtresult.value
            replace returned with thisform.pageframe1.page2.chkreturned.value

        thisform.cmdgrpeditor.cmdadd.visible = .T.
        thisform.cmdgrpeditor.cmdrevert.visible = .F.

        thisform.cmdgrpeditor.cmdmodify.visible = .T.

        thisform.cmdgrpnavigator.enabled = .T.
        Thisform.setall("Readonly",.T.,"Textbox")
        Thisform.setall("Readonly",.T.,"EDITbox")
    ENDPROC


    PROCEDURE cmdgrpeditor.cmddelete.Click
        store 0 to rep1
        *rep1 = messagebox("Are you really sure ?", 36, "Want to Delete ?")
        *if rep1 = 6 then
        DELETE
        PACK
        THISFORM.REFRESH()
        *endif
        if this.parent.cmdadd.visible == .F.
            this.parent.cmdadd.visible = .T.
            this.parent.cmdrevert.visible = .F.
            this.parent.cmdsave.enabled = .F.
        endif
ENDPROC


PROCEDURE cmdgrpeditor.cmdexit.Click
        THISFORM.RELEASE()
ENDPROC


PROCEDURE cmdgrpeditor.cmdrevert.Click
        if modified == .f.
        GO BOTTOM
        DELETE
        PACK
        else
                        replace who with mwho
                        replace when with mwhen
            replace where with mwhere
                        replace how with mhow
                        replace whenfirst with mwhenfirst

                        replace related with mrelated
                        replace periodic with mperiodic
                        replace prob_type with mprob_type
                        replace prob_desc with mprob_desc
            replace complaint with mcomplaint

            replace components with mcomponents
            replace observation with mobservation
            replace siteaction with msiteaction

            replace date_in with mdate_in
            replace date_out with mdate_out
            replace shopaction with mshopaction
            replace result with mresult
            replace returned with mreturned

        endif
        Thisform.setall("Readonly",.T.,"Textbox")
        Thisform.setall("Readonly",.T.,"EDITbox")

        thisform.cmdgrpeditor.cmdadd.visible = .T.
        thisform.cmdgrpeditor.cmdmodify.visible = .T.
        thisform.cmdgrpeditor.cmdrevert.visible = .F.
        thisform.cmdgrpeditor.cmdsave.enabled = .F.
```

```
	thisform.cmdgrpnavigator.enabled = .T.

	THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpeditor.cmdmodify.Click

	Thisform.setall("Readonly",.F.,"Textbox")
	Thisform.setall("Readonly",.F.,"editbox")

	thisform.cmdgrpnavigator.enabled = .F.

	modified = .t.

			Mwho = thisform.txtwho.value
			mwhen = thisform.txtwhen.value
		mwhere = thisform.txtwhere.value
		mhow = thisform.cbohow.value
			mwhenfirst = thisform.txtwhenfirst.value

			mrelated = thisform.chkrelated.value
			mperiodic = thisform.chkperiodic.value
			mprob_type = thisform.cboprob_type.value
			mprob_desc = thisform.edtprob_desc.value
		mcomplaint = thisform.edtcomplaint.value

		mcomponents = thisform.pageframe1.page1.edtcomponents.value
		mobservation = thisform.pageframe1.page1.edtobservation.value
		msiteaction = thisform.pageframe1.page1.edtsiteaction.value

		mdate_in = thisform.pageframe1.page2.txtdate_in.value
		mdate_out = thisform.pageframe1.page2.txtdate_out.value
		mshopaction = thisform.pageframe1.page2.edtshopaction.value
		mresult = thisform.pageframe1.page2.edtresult.value
		mreturned = thisform.pageframe1.page2.chkreturned.value

	thisform.cmdgrpeditor.cmdadd.visible = .F.
	thisform.cmdgrpeditor.cmdmodify.visible = .F.
	thisform.cmdgrpeditor.cmdrevert.visible = .T.
	thisform.cmdgrpeditor.cmdsave.visible = .T.
	thisform.cmdgrpeditor.cmdsave.enabled = .T.
```

```
        *
        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdtop.Click
        GOTO TOP
        THISFORM.REFRESH()


        this.parent.cmdprevious.enabled = .F.
        this.enabled = .F.

        this.parent.cmdnext.enabled = .T.
        this.parent.cmdbottom.enabled = .T.
ENDPROC


PROCEDURE cmdgrpnavigator.cmdprevious.Click
        IF !BOF()
                SKIP -1
                this.parent.cmdbottom.enabled = .T.
                this.parent.cmdnext.enabled = .T.
                IF BOF()
                        GO TOP
                        this.parent.cmdtop.enabled = .F.
                        this.enabled = .F.

                        this.parent.cmdnext.enabled = .T.
                        this.parent.cmdbottom.enabled = .T.
                else

                        this.parent.cmdtop.enabled = .T.
                        this.enabled = .T.
                ENDIF
        ENDIF
        THISFORM.REFRESH()
ENDPROC


PROCEDURE cmdgrpnavigator.cmdnext.Click
        IF !EOF()
                SKIP
                this.parent.cmdtop.enabled = .T.
                this.parent.cmdprevious.enabled = .T.
                IF EOF()
```

```
                        GO BOTTOM
                        this.parent.cmdbottom.enabled = .F.
                        this.enabled = .F.

                        this.parent.cmdprevious.enabled = .T.
                        this.parent.cmdtop.enabled = .T.
                ENDIF
        ENDIF
        THISFORM.REFRESH()
    ENDPROC


    PROCEDURE cmdgrpnavigator.cmdnext.MouseDown
        LPARAMETERS nButton, nShift, nXCoord, nYCoord
    ENDPROC


    PROCEDURE cmdgrpnavigator.cmdbottom.Click
        GO BOTTOM
        THISFORM.REFRESH()


        this.parent.cmdnext.enabled = .F.
        this.enabled = .F.

        this.parent.cmdtop.enabled = .T.
        this.parent.cmdprevious.enabled = .T.
    ENDPROC


ENDDEFINE
*
*-- EndDefine: form1
**************************************************
```