AL-HIKMAH UNIVERSITY, ILORIN, NIGERIA
Adeta Road, Adewole Housing Estate, P.M.B. 1601, Ilorin
…..learning for wisdom and morality…..
CENTRE FOR ICT AND DISTANCE LEARNING (IDL)
e-CONTENT DEVELOPMENT (DL) UNIT

e-Note for CMP 102

A. Course Lecturer's Details
Name: AKANNI A.O., OYELAKIN A.M., GBOLAGADE M.
Mobile Number: +2348167813328, +2348062738520, +2347030848418
Official Email Address: seyeakanni@alhikmah.edu.ng, amoyelakin@alhikmah.edu.ng, dammyconsult@gmail.com

B. Faculty Department and Programme
Faculty: Natural and Applied Sciences
Department: Computer Science
Programme: B.Sc. Computer Science
C. Course Title: Problem Solving
D. Course Code: CMP 102
E. Course Credit: 03

F. Course Description

Introduction to problem solving, problem solving methods: algorithm development and flowcharting, stages in System Development Life Cycle, Concepts in BASIC Programming, Designing, coding, debugging and documenting programs using techniques of a good programming language style, Programming examples and solutions and Questions for practice

G. Learning Objectives

At the completion of this e-note, it s expected that learners will achieve the following:

1. Understand the basic concepts of problem solving

2. Learn and master the necessary skills on problem solving tools: Algorithm and Flowchart

3. Exhibit mastery of techniques for writing good programs in BASIC (VB 6) Language

4. Understand and solving some real life problem solving scenarios

Notes on Topics to be taught

SECTION ONE

COMPUTERS AND ITS AREAS OF APPLICATION

A computer is an electronic machine that receives data as input, processes and stores the data, and then provides output in a specified format. On every computer system, there is some set of programs running. A program is a sequence of instructions that can be executed by a computer to solve some problem or perform a specified task.

Computers are pervasive these days. For this reason, every profession requires some level of literacy in computing. Information and Communication Technology has advanced greatly and it is important that as undergraduates you are well equipped with ICT know how that will enable you perform better on your job, career and day-to-day living. Our emphasis in this course is on problem solving in relation to computing. We want to remind every undergraduate taking this course to be well aware that in the course of your study and career after graduation, you will require to use problem solving skills in various situations. For this reason, pay close attention to the concepts that will be taught therein.

Computer and its Characteristics

Before we go deeply, it is important to remind the learners of some basic characteristics of computers. This will serve as a reminder on why computers are becoming very popular as a tool in every profession for solving problems.

Some of the main characteristics of computers include:

i.   Speed
ii. Accuracy
iii. Versatility
iv. Electronic in nature
v.   Repetitiveness

Areas of Application of Computers

There are uncountable application areas of computers. Different disciplines and sectors have a wide range of applications in which they use computer systems for, on daily basis. The list included herein does not mean the exhaustive. Some of them are as listed and discussed below:

Businesses

Computers are widely used in businesses. For instance, Businessmen make bar graphs and pie charts from tedious figures to convey information with far more impact than numbers alone can covey. Furthermore, computers help businesses to predict their future sales, profits, costs etc. making companies more accurate in their accounts. Thus, computers are used for decision making in businesses.

Buildings and Construction Engineering

Architects use computer animated graphics to experiment with possible exteriors and to give clients a visual walk-through of their proposed buildings. The computers provide architects numerous amount of facilities to create different buildings with greater accuracy, better designing and editing tools, and work done at the fastest speed possible.

Education

Computers and ICT have been fully integrated into schools at all levels for learning purposes. Most schools in the world have computers available for use in the classroom. It has been proved that learning with computers has been more successful and this is why numerous forms of new teaching methods have been introduced. This enhances the knowledge of the students at a much faster pace than the old traditional teaching methods.

Oil and Gas Sector

Energy companies use computers to locate oil, coal, natural gas and uranium. With the use of these technological machines, these companies can figure out the site of a natural resource, its concentration and other related figures. Electric companies use computers to monitor vast power networks. In addition, meter readers use hand held computers to record how much energy is used each month in homes and offices.

Law Enforcement

Computers have been widely used in Law and Justice sector. Recent innovation in computerised law enforcement include national fingerprint files, a national file on the mode of operation of serial killers, and computer modeling of DNA, which can be used to match traces from an alleged criminal's body, such as blood at a crime scene. Computers have been widely used to store and manage people with criminal records.

Transportation

Computers are used in cars to monitor fluid levels, temperatures and electrical systems. Computers are also used to help run rapid transit systems, load containerships and track railroads cars across the country. An important part is the air control traffic systems, where computers are used to control the flow of traffic between airplanes which needs a lot of precision and accuracy to be dealt with.

Banking Sector

Computers speed up record keeping and allow banks to offer same-day services and even do-it yourself banking over the phone and internet. Computers have helped fuel the cashless economy, enabling the widespread use of credit cards, debit cards and instantaneous credit checks by banks and retailers. There is also a level of greater security when computers are involved in money transactions as there is a better chance of detecting forged cheques and using credit/debit cards illegally etc.

Agriculture

Farmers use computers to help with billing, researches, crop information, and cost per acre, feed combinations, and market price checks. Cattle ranchers can also use computers for information about livestock breeding and performance.

Learning and Entertainment tools at homes

People having computers in their homes justify the fact that it is not only useful and efficient, but it is also revered as a learning system. Personal computers are being used for tasks such as to keep records, write letters and memos, prepare budgets, produce presentations, draw pictures, publish newsletters and most importantly - connect with other in the rest of plant earth.

Health Sector (Medicine)

Computers are helping immensely to monitor thee extremely ill in the intensive care unit and provide cross-sectional views of the body. This eliminates the need for hired nurses to watch the patient twenty-four hours a day, which is greatly tiring and error prone. Doctors use computers to assist them in diagnosing certain diseases of the sort. This type of computer is called the Expert System, which is basically a collection of accumulated expertise in a specific area of field. Furthermore, computers are used greatly in managing patients, doctors, wards and medicine records, as well as deal with making appointments, scheduling surgeries and other likes.

Manufacturing Industries

Computers have made their way towards jobs that were unpleasant or too dangerous for humans to do, such as working hundreds of feet below the earth or opening a package that might contain an explosive device. In other industries, computers are used to control the production of resources very precisely. All robots and machinery are now controlled by various computers, making the production process faster and cheaper. All the stages of manufacturing, from designing to production, can be done with the use of computer technology with greater diversity.

Computers are used in Scientific Research

This is very important for mankind and with the development of computers; scientific research has propelled towards the better a great deal. Because of high-speed characteristics of computer systems, researchers can simulate environments, emulate physical characteristics and allow scientists to proof of their theories in a cost-effective manner.

SECTION TWO - INTRODUCTION TO PROBLEM SOLVING

Problem Solving- An Explanation

Problem solving is an integral part of Computer Science. Computer is used as a problem solving tool in both public and private organizations. This statement is a testimony to the rate at which computer and ICT solutions are used for driving services and operations of government and private establishments in Nigeria and beyond. From different sectors of the economy, computers are being put to use. They are used for manufacturing, educating, researching, distributing, and increasing productivity.

While we want to develop a software application, set up a network, troubleshoot a computer problem (hardware or software) we engage n problem solving. The problem solver is expected to make use of some skills and techniques while solving problems.

More importantly, a good understand of problem solving enables us to use computers in addressing several problems that you will be encountering in the day-to-day discharge of your duties. This is because problems are at the center of what many people do at work every day. Whether you're solving a problem for a client (internal or external), supporting those who are solving problems, or discovering new problems to solve, the problems you face can be large or small, simple or complex, and easy or difficult.

Problem solving is a crucial part of Computer Science and/or computing. It is required that when you intend using any CIT tools or techniques, the problems you intend using them for are important. It is the act of defining a problem; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution.

Problem-solving is necessary on every job. It is important that students develop the skills to resolve problems and have the personal resilience to meet the challenges and pressure that may be the result of a problem. Problem-solving requires a variety of both analytical and creative thinking skills. Which are used depends on the role in the organization and the problem.

What are Problem-Solving Skills?

Problem solving skills may be grouped into soft skill and hard skill. Considered a soft skill (a personal strength, as opposed to a hard skill that is learned through education or training), an aptitude for creative and effective problem-solving is nonetheless one of the most valued attributes employers seek in their job candidates. For example, a cable television technician might be trying to resolve a customer problem with a weak signal. A teacher might need to figure out how to improve the performance of her students on a writing proficiency test. A store manager might be trying to reduce theft of merchandise. A computer specialist might be looking for a way to speed up a slow program.

Required Skills for a problem solver

Some of the required skills that a problem solver is expected to have include:

i. Active Listening
ii. Data Gathering
iii. Data Analysis
iv. Fact Finding
v. Historical Analysis
vi. Causal Analysis
vii. Process Analysis
viii. Needs Identification

Some good examples of real life problems that can be attended to using problem solving skills are: Diagnosing Illnesses, Identifying the Causes for Social Problems, Interpreting Data to Determine the Scope of Problems, Pinpointing Behaviors Contributing to Marital Distress, Recognizing Invalid Research Models

Problem Solving and Decision Making in work place

Problem-solving and decision-making do go pari-pasu. This is because, a problem solver is providing a solution to a particular problem in his domain with a view to making useful decision. Ask anyone in the workplace if these activities are part of their day and they answer 'Yes!' But how many of us have had training in problem-solving? We know it's a critical element of our work, but do we know how to do it effectively?

People tend to do three things when faced with a problem: they get afraid or uncomfortable and wish it would go away; they feel that they have to come up with an answer and it has to be the right answer; and they look for someone to blame. Being faced with a problem becomes a problem. And that's a problem because, in fact, there are always going to be problems!

There are two important things to remember about problems and conflicts: they happen all the time and they are opportunities to improve the system and the relationships. They are actually providing us with information that we can use to fix what needs fixing and do a better job. Looked at in this way, we can almost begin to welcome problems!

Seven-Steps for an Effective Problem-Solving Process

Below are some of the key steps that can be taken in every problem solving process.

1. Identify the issues.

- Be clear about what the problem is.
- Remember that different people might have different views of what the issues are.
- Separate the listing of issues from the identification of interests (that's the next step!).

2. Understand everyone's interests.

- This is a critical step that is usually missing.

7

- Interests are the needs that you want satisfied by any given solution. We often ignore our true interests as we become attached to one particular solution.
- The best solution is the one that satisfies everyone's interests.
- This is the time for active listening. Put down your differences for awhile and listen to each other with the intention to understand.
- Separate the naming of interests from the listing of solutions.

3. List the possible solutions

- This is the time to do some brainstorming. There may be lots of room for creativity.
- Separate the listing of options from the evaluation of the options.

4. Evaluate the options.

- What are the pluses and minuses?
- Separate the evaluation of options from the selection of options.

5. Select an option or options.

- What's the best option, in the balance?
- Is there a way to "bundle" a number of options together for a more satisfactory solution?

6. Document the agreement(s).

- Don't rely on memory.
- Writing it down will help you think through all the details and implications.

7. Agree on contingencies, monitoring, and evaluation.

- Conditions may change. Make contingency agreements about foreseeable future circumstances (If-then!).
- How will you monitor compliance and follow-through?
- Create opportunities to evaluate the agreements and their implementation.

PROBLEM SOLVING TOOLS IN SOFTWARE DEVELOPMENT

In the field of Computer Science, there are various problem solving tools that are widely used. In the area of modeling and developing an application (desktop, mobile or server-based), we make use of problem solving skills and tools. This is because such a task requires some approaches. Some of the tools used include: Algorithm, Flowchart, Unified Modeling Language (UML), Decision tables and so on.

Note: In the context of this course, only Algorithm and Flowchart will be covered at this introductory level

While developing software, there are stages that the software product (program) must pass through. The stages are generally called Software Development Life Cycle (SDLC).

The phases of Software Development Life Cycle are:

Problem Definition/User Requirement Analysis

Program Design

Program coding/System coding

Program Testing and Debugging

Program/System Implementation

System Evaluation and Maintenance

Algorithm/Flowchart as Problem Solving Tools

Understanding these stages are vey important. Moreover, while providing solutions to a wide range of IT problems, problem solving tools can be used. Algorithm and Flowchart are being introduced herein as problem solving tools that can help drive varying solutions.

ALGORITHM

An algorithm can be defined as the step-by-step procedure for solving a given problem. It clearly spells out how a solution to a particular problem is achieved.

Characteristics of a good Algorithm

A good algorithm must be exact in the process of providing solutions to the problem

A good algorithm must not be ambiguous. That is, the algorithm must be clear enough.

A good algorithm must be finite. This statement means that a good algorithm must have a beginning and an end. It must not run indefinitely.

Input- a good algorithm must have a given set of input to be worked upon

Output- A good algorithm must have a given set of output to be generated

A problem solver can make use of algorithm as a tool. This is because the problem solver thinks critically and is able to list out the various steps he will use to accomplish a given task or solve a give problem. A problem solver should be able to think critically on problem he intends solving. He is expected to think out of the box in the process of getting a solution provided to an issue. It is expected that a problem solve able to identify the possible approaches for providing solution to the problem at hand.

Example of Algorithm as a problem solving tool
//This is an algorithm to compute the area of a circle
Step I:Start
Step II:Initialise Pi=3.142
Step III:Enter radius of the circle
Step IV: Compute Area=Pie * square of radius
Step V: Print Area
Step VI: Stop

Another Example of an Algorithm to enter surname, firstname, middlename, faculty, department, gender and age of a student is as shown below:

//An algorithm to accept and display details of a student
Step I: Start
Step II: Input surname
Step III: Input firstname
Step IV: Input middlename
Step V: Input faculty
Step VI: Input department

Step VII: Input gender

Step VIII: Input age

Step IX: Print surname, firstname, middlename,faculty,department,gender,age

Step X; Stop

Note: Different problems require the use of appropriate approaches while designing algorithms. For instance, some problems may require you to test for conditions and thus you will need to use control structures such as IF statement, IF THEN ELSE or IF THEN ELSE IF statements.

During lectures, efforts will be made to explain these further.


FLOWCHART

A flowchart is the pictorial representation of an algorithm. That is a flowchart refers to as the diagrammatic representation of steps followed in solving a given problem. A problem solver should be able to sketch an acceptable flowchart for a problem, the different flowchart symbols must be understood.

**Benefits/Advantages of Flowcharting in Problem Solving**

 i. Visual Clarity. One of the biggest benefits of a flowchart is the tool's ability to visualize multiple progresses and their sequence into a single document.

 ii. Instant Communication

 iii. Effective Coordination

 iv. Efficiency Increase

 v. Effective Analysis

 vi. Problem Solving

 vii. Proper Documentation

To be able to draw a flowchart for a given problem, flowchart symbols are used. Examples of these flowchart symbols include:
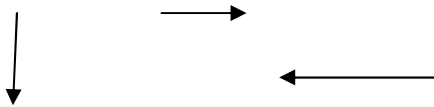

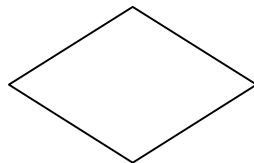**Terminal Symbol**-This is the symbol that is used to indicate the beginning and ending of a flowchart

Start
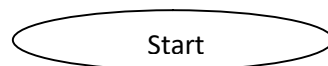
Processing Symbol-This is the symbol that is used to denote processing and computational activities.

```
┌─────────────────────────────────────┐
│            Calculate Total          │
│                                     │
└─────────────────────────────────────┘
```
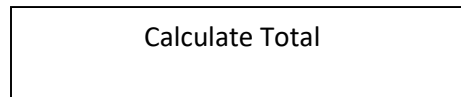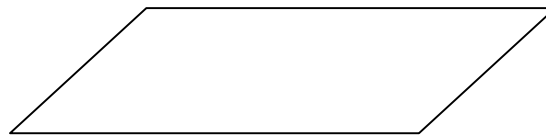
Flow Arrow-This is the symbol that is used to indicate the direction in which a particular operations is going.

Decision Symbol-This is the flowchart symbol that is used while making decision.

Input and Output Symbol-This is the kind of symbol that is used for feeding the computers with the set of needed inputs. The symbol is equally used for generating outputs.
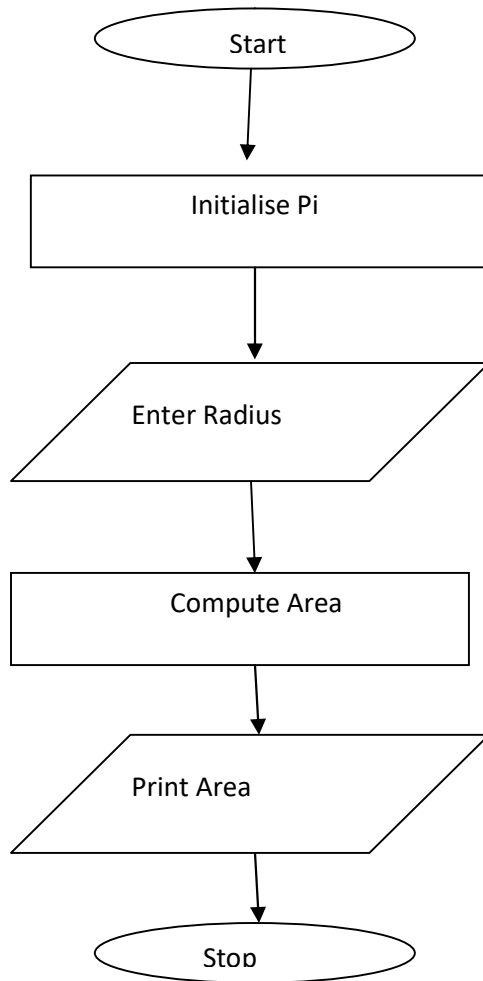
Connector-This is used to join a flowchart diagram from one page to another. It should be properly labelled so as to avoid confusing the user of the diagram.

In every problem that requires the drawing of a flowchart, the problem solver must take his time to study the logic involved. This will enable him/.her to know which of the symbols will be used for the different stages in getting the problem solved.

Flowchart for the computation of area of a circle

```
        ┌─────────────┐
        (    Start     )
        └──────┬──────┘
               │
               ▼
      ┌────────────────────┐
      │    Initialise Pi    │
      └─────────┬──────────┘
                │
                ▼
      ╱────────────────────╱
     ╱     Enter Radius    ╱
    ╱────────────────────╱
                │
                ▼
      ┌────────────────────┐
      │    Compute Area     │
      └─────────┬──────────┘
                │
                ▼
      ╱────────────────────╱
     ╱      Print Area     ╱
    ╱────────────────────╱
                │
                ▼
        ┌─────────────┐
        (     Stop     )
        └─────────────┘
```

Introduction Programming Language Concepts

As part of solving problems, programming languages are used in computing. There are tens of thousands of programming languages. It is important for a problem solver to be farmiliar with some concepts that will help them use the languages in providing programming solutions or developing programs.

To understand programming languages better, it may be necessary to have mastery of some of the basic concepts. The list of programming language concepts is numerous. Some of them are:

Program

Syntax

Syntax Error

Translator (Compiler, Interpreter and Assembler)

Logical Error

Loop

Data Types

Looping

Identifier-this can be a variable name, a function name and so on.

Constant

Control Structures

Program logic

Functions (In-built function, user-defined function) and many others.

Lets briefly explain few of these concepts. Others will be discussed during lectures.

Program- A program is the set of instructions that tells the computer system how to perform a given task.

Syntax-This simply refers to as the set of rules which have to be followed while writing a valid program statement in the chosen programming language. For instance, the language we are using in this course is BASIC (Visual BASIC 6). As a way to be able to use the language to develop applications, the syntax of the language must be mastered.

If you intend entering a values in VB 6 environment, you make use of InputBox function and so on.

Syntax Error- This is defined as the type of errors that occurs when the syntax of a programming language is violated.

Translator- A translate is defined as the type of software that is used to convert a program written in high level or low level language to machine language. This translator is of three types, namely: Assembler, Interpreter and Compiler.

Low Level Language (Assembly Language) uses Assembler. That is an assembler converts low level language source codes to machine code.

Interpreter-This is the type of translator that converts program written in high level language to machine code , statement by statement (or one-by-one)

Compiler-This is the type of program translator that converts program written in High Level Language to machine code all at once. It is more faster when compared to the interpreter.

Note: Some of the terms listed above and others will be defined in the course of treating topics under which they are relevant. The language we are introducing you to in this course is a good example of High Level Language.

Computer Programming is the act of coding instructions for computer to carry out. Providing solution to a problem is achieved with the use of suitable programming language. The choice of a programming language for a given problem depends on a number of factors. These factors include:

the programming needs,

the domain of the problem (education, enterprise, hospitality, banking, e.t.c)

the ease of use of the programming language

the level of education/technical expertise of the programmer

among others.

There are three levels of programming languages. They are:

Machine Language;

Low Level Language (LLL) or Assembly Language and;

High Level language (HLL).

Examples of High Level Languages are: BASIC, FORTRAN, Scala, C++, C. PASCAL, Ada, Python, Java, Smalltalk, Delphi, and so on.

More so, programming languages can be classified into Imperative, Functional, Structured, Object Oriented programming languages and so on. Object Oriented Programming is a programming method that combines data and instructions for processing that data into a self-sufficient 'object' that can be used within a program or in other programs. In OOP, Objects are modeled on real world entities. This enables modeling complex systems of real world into manageable software solutions.

These languages are written using human language and have their well defined syntaxes. A programmer in whatever level has to obey the syntax of the chosen language he/she wants to use to develop the targeted applications/softwares.

SECTION FOUR

INTRODUCTORY PROGRAMMING IN (VISUAL) BASIC

i. BASIC is a High level programming language,

ii. BASIC is a procedural programming language

iii. It produces efficient programs

iv. BASIC is Easy to learn and use for programming problems

v. It is a good example of interpreted programming language. That is, it makes use of an interpreter

Note: While writing programs in BASIC or any other language, efforts should be made to have a good understanding of the syntaxes of the language. Syntax simply means the set of rules that have to be followed while writing valid program statement in the language. When the syntax of the language is violated, we have what is called syntax error.

BASIC has several versions and has evolved over years. Examples of versions of BASIC that we have include: Standard BASIC, Quick Basic, GW BASIC, Visual BASIC (6,.NET. 2010,2013 and so on). The version of BASIC programming language that we recommend for practicing the art of problem solving in this is Visual BASIC. It is a third-generation event-driven programming language released by Microsoft Corporation for its Component Object Model programming model first released in 1991 and is still being used to introduce students to the art of visual programming.

Visual Basic Environment (VB 6.0)

Visual Basic is a version of BASIC Programming language that is used for building applications of various types and functionalities. The programming language uses interpreter, meaning that the program source codes are converted into object codes, statement by statement. The Visual BASIC working environment is an IDE (Integrated Development Environment) that allows programmers to write, compile and execute program instructions.

Note:

As a language, BASIC support s the use of commenting. The purpose of commenting is for program documentation. The comment can appear at the beginning of the program or any other parts, depending on the message that is intended to be passed. Figures 1,2, and 3 are images of the Visual Basic working environment.
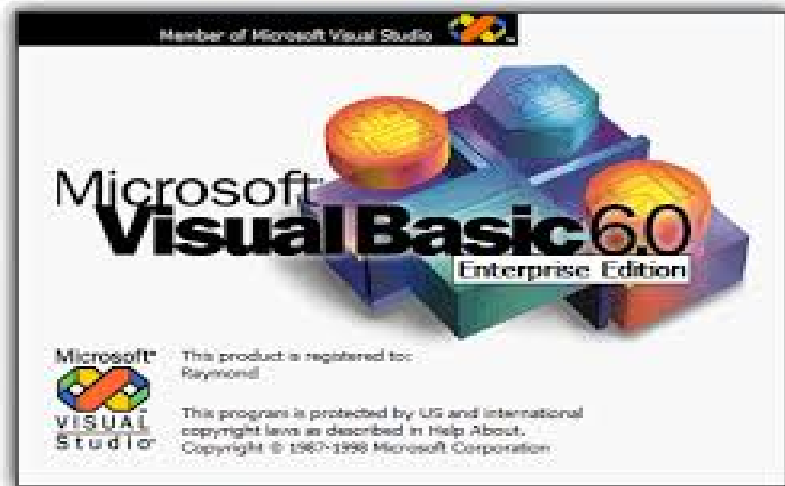

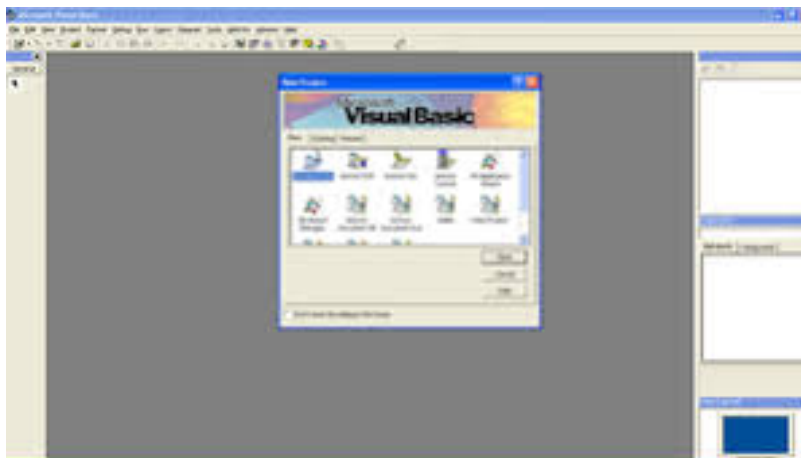Figure 1: Visual Basic 6 Splashcreen at start up

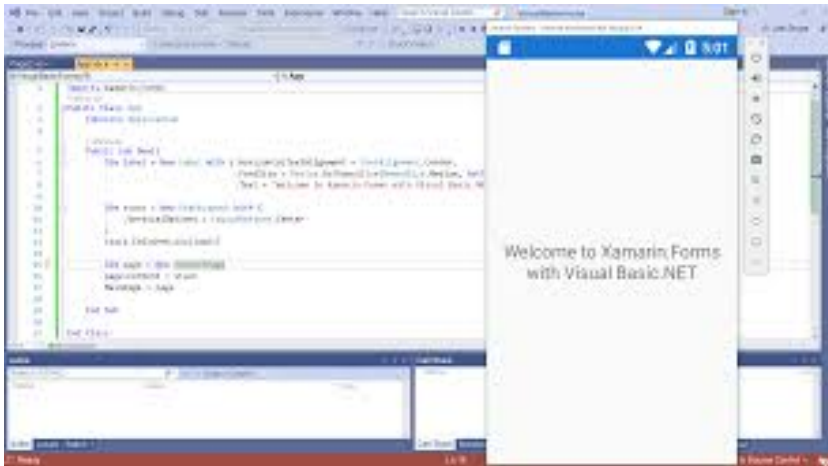Figure 2: Start up page for selecting Project Type

Figure 3: Visual Basic 6 Code Window

Note: It is recommended that students get installation disc of VB 6 and install it on their systems. The department also provides the installation software for the students. You can contact the course handlers for the collection and installation of the software, free of charge.

The Visual Basic 6 environment is an integrated environment with a lot of tools (forms, controls, menus, properties window and so on. During lectures, efforts will be made to show you some of these features and how to use them.

Keywords in Visual BASIC Programming Language

A keyword refers to as the word that has predefined meaning to the BASIC interpreter. Some good examples of keywords in BASIC include: DIM, Inputbox, MsgBox, if, then, else, do, while, for, next, for, switch, end, end if, byte, call, goto, dim, while, do, switch and so on. The list is very long.

Note: For the purpose of this course, you are only being introduced to some of them. To be able to program effectively in any programming language, it is required that the programmer masters the syntaxes of such language and then develop the ability for reasoning based on the required program logic of each problem. Programmer does not need to cram anything as problem varies.

Choosing Variable Names in BASIC Programming Language

In BASIC programming language, a variable or variable name is used as the storage location for data. Depending on the data type that we want to use in a program, we declare the variable as such. To have a valid variable name in BASIC, one is expected to have a mastery of the rules for declaring the variables.

Some of such rules include:

i. A valid variable name in BASIC should start with an alphabet or letter InitialValue,Num,Value

ii. A variable name in BASIC can be a mixture of alphabet or number (but must not start with the number) e.g. Traveller45,T78,A5,Value34,num23,number892

iii. A valid variable name must not be BASIC keyword

iv. A valid variable name must not contain blank space.

v. A valid variable name may contain alphabets, numbers and/or with underscore (_) e.g. sum_score

vi. A variable name can contain up to 255 characters

More Examples (for clarification purposes only, pls. Efforts will be made to explain these during lectures)

For instance the following are valid variable names because they obey one or a combination of the rules mentioned above:

(i)sum (ii) studentscore (iii)Totalsalary23 (iv) Manager_Pay (v) StudentAdd4 e.t.c

The following variable names are not valid

(i)   34Name (ii) Sum total (iii) 7President name* (iv)printf (v) getchar (vi) N89}

What are data types in Visual BASIC 6?

Data Types- a data type describes the kinds of data that can be entered into the computer system at a time.

By default Visual Basic variables are of variant data types. The variant data type can store numeric, date/time or string data. When a variable is declared, a data type is supplied for it that determines the kind of data they can store. The fundamental data types in Visual Basic including variant are integer, long, single, double, string, currency, byte and boolean. Visual Basic supports a vast array of data types. Each data type has limits to the kind of information and the minimum and maximum values it can hold. In addition, some types can interchange with some other types. A list of Visual Basic's simple data types are given below.

1. **Numeric**

Byte	Store integer values in the range of 0 – 255
Integer	Store integer values in the range of (-32,768) - (+ 32,767)
Long	Store integer values in the range of (- 2,147,483,468) - (+ 2,147,483,468)
Single	Store floating point value in the range of (-3.4x10-38) - (+ 3.4x1038)
Double	Store large floating value which exceeding the single data type value
Currency	store monetary values. It supports 4 digits to the right of decimal point and 15 digits to the left

2. **String**

Use to store alphanumeric values. A variable length string can store approximately 4 billion characters

3. **Date**

Use to store date and time values. A variable declared as date type can store both date and time values and it can store date values 01/01/0100 up to 12/31/9999

4. **Boolean**

Boolean data types hold either a true or false value. These are not stored as numeric values and cannot be used as such. Values are internally stored as -1 (True) and 0 (False) and any non-zero value is considered as true.

5. **Variant**

Stores any type of data and is the default Visual Basic data type. In Visual Basic if we declare a variable without any data type by default the data type is assigned as default.

Note: It is expected that a programmer determines and use the most appropriate data type in a particular program before proceeding to writing the codes.

This course being an introductory course considers only simple data types in some of the examples and exercises.

Understanding Looping Structures in BASIC (Visual Basic 6) Language

BASIC, as a programming language, supports a wide range of loop structures. The popular among them are FOR LOOP, DO WHILE and WHILE DO. Emphasis will be on FOR LOOP, DO WHILE LOOP and WHILE DO LOOP. The main purpose of using loops is to perform some program segments in a repeated manner.

Like any other program constructs, these loops have their syntax which must be adequately mastered and put to use for proper practical mastery. Before, we go further; there may be a need to define looping. Let says, the problem solver wants to display or generate some values or statement several times and does not need to that one-by-one, he/she will resort to using loops as may be required.

What is looping?

Looping can be defined as the process of performing certain segment of a program n a repeated manner until the specified condition is met. The loop allows programmer to write program codes that are able to run repeatedly.

Syntax of FOR LOOP in BASIC

This is the simplest and mostly used loop statement for beginners.

It has the general syntax:

The syntax of a FOR LOOP in a C Program is as follows:

> *FOR index variable=initial condition TO Finalvalue*
>
> *Program codes*
>
> *NEXT (index variable)*

Visual BASIC 6 Program Examples:

(i)

Integer k
FOR k=1 To 50
Msgbox(" We are learning Problem Solving at Al-Hikmah University")
NEXT k

This code above runs repeatedly until the specified value of fifty count of the message is reached. The index variable is incremented by 1,one at a time.

(ii)

Integer J;
Integer Sum
LET Sum=0
FOR J= 1 TO 122
LET Sum=Sum+J
Msgbox(Sum)
NEXT
STOP

The lines of Visual BASIC code above add and print sum of integers from 1 to 122

THE WHILE LOOP /WHILE Statement

The while statement continually executes a block of statements while a particular condition is true. Its syntax can be expressed as:

while (expression)
```
 {
    //Program statement(s)
}
```
The while statement evaluates expression, which must return a boolean value. If the expression evaluates to true, the while statement executes the statement(s) in the while block. The while statement continues testing the expression and executing its block until the expression evaluates to false.

I. Study Questions

1. Define each of the following terms: (i) Program (ii) Programming (iii) Problem solving

2. Write a C program to find the perimeter of a rectangle given that P=2(L+B) where L and B are length and Breath respectively.

3. Design and algorithm and a flowchart to find the perimeter of a rectangle

Identify the basic steps in Problem solving

4. Why is there a need for commenting in a BASIC Program?

5. Design an algorithm and a flowchart to compute the value of T given that T=23M-4G/3

Note: M and G are input variables.

6. Write a Visual Basic program code to find the sum of four integers entered from the keyboard.

7. Itemise the stages of Software Development Process/Software Development Life Cycle

8. Design a flowchart to generate integer values from 1250 to 350

9. Write a BASIC program to display the message "Al-Hikmah University is waxing stronger" five hundred times

10. Comment on the importance of problem solving skills in programming as a student programmer

11. Write a Visual BASIC program to find the roots of a quadratic equation given that $ax^2+bx+c=0$ **Hint:** Use Almighty formula for your problem

12. Identify any five flowchart symbols that you know

13. List and explain any seven skills that a problem solver should possess

14. Why is it necessary for a programmer to understand the type of data he wants to handle in a programming problem?

J. References

1. Mastering Visual Basic 6, by Envangelos Petroutsos
2. Tim Hiks (n.d.). Seven Steps for Effective Problem Solving in the Workplace, retrieved from https://www.mediate.com/articles/thicks.cfm
3. Problem Solving Skills: Definition, Steps, and Examples
4. Alison Doyle (2020). https://www.thebalancecareers.com/problem-solving-skills-with-examples-2063764
5. FreeTures.com (n.d.) Visual Basic Tutorials, retrieved from https://www.freetutes.com/learn-vb6/lesson2.html

K. Recommended books/Materials /Web Documents

1. Mastering Visual Basic 6, by Envangelos Petroutsos
2. Tim Hiks (n.d.). Seven Steps for Effective Problem Solving in the Workplace, retrieved from https://www.mediate.com/articles/thicks.cfm