This paper examines the general characteristics of PASCAL, BASIC and C programming languages. It then enumerates important factors that determine the choice of a programming language. An assessment of PASCAL, BASIC and C is then made using some fundamental attributes of programming languages to discriminate their choice.

1. **INTRODUCTION**

PASCAL is an imperative computer programming language developed in 1970 by Niklaus Wirth as a language particularly suitable for structured programming\[1,2\]. It is based on the ALGOL programming language and named in honour of Mathematician and Philosopher, Blaise Pascal. C programming language on the other hand is a general-purpose programming language developed in 1972 by Dennies Ritchie at Bell Telephone Laboratories for use with the Unix Operating System\[4\]. It is predominantly used for system software\[3\] and has greatly influenced many other popular languages\[5,6\] especially C++ which was designed as an enhancement of C. The BASIC programming language, an acronym for Beginner’s All-Purpose Symbolic Instruction Code refers to a family of high level programming languages which was originally designed in 1963 by John George Kemeny and Thomas Eugene Kurtz to provide access to computer for non-science students\[7\]. Certain features are common to all high level programming languages. These features are explained as follows:

Features of Pascal, Basic and C programming languages.

1. **Vocabulary:** These are set of characters and words from which programs are written.
2. **Control Structure:** All programming languages has a way of transferring control from one section of the program to another using selection, sequencing and iteration.
3. **Character Set:** These are set of valid characters used in a program written in a language such as letters, decimal digits with differences in special characters.
4. **Reserved Words:** These are words that have pre-defined meaning for special operation and cannot be used for any other purposes.
5. **Operation:** It is through here that data items and data are manipulated with rules for order of procedure for operation.
6. **Data:** Data item could be a variable which can change throughout the execution of a program or consent in which case its value does not change throughout the execution of the program. The variable names are called identifiers. Programming language such as PASCAL wants you to declare your variable in the early part of the program before they are being used in any part of it. All programming languages has data types such as real integer, constant, character, Boolean etc.
7. **Structure:** The way program are written or constructed is called its structure using set of rules called syntax.

8. **Input/Output:** These are ways of transferring data into or out of the computer.

### Choice of Programming Language
The choice of a particular programming language to use to solve a problem depends on the following:

(i) **Type of problem:** The type of problem to solve determines the choice of programming language to use. C is a good system implementation language, and can easily be computed in a straight forward manner using a relatively simple compiler, provide low level access to memory, generate only a few machine languages instructions for each of its core language elements and does not require extensive run-time support. It is therefore suitable for many applications that had traditionally been implemented in Assembly language. PASCAL programming languages is distinguished from C by being much more Algol-like. Pascal has ‘`and`, `or` and `mod` where C uses `$\&\&$, `!!` and `%`. In PASCAL also, semicolon separates individual statement within a compound statement whereas in C, they are syntactically part of the statement itself (transforming an expression into a statement) i.e. there can never be a semicolon directly before `else` in PASCAL, wherein it is mandatory in C (unless a block statement is used), and the last statement before an `end` is not required to be followed by a semicolon. Some programming may however put a semicolon on the last line before `end` thereby formally inserting an empty statement. This is however discouraged by some educators as it may confuse students’ perception of the formal role of the semicolon. BASCIC on the other hand was designed to provide access for non-science students as well as beginners, as it shield the users from the operating system, provide clear and friendly error message, very interactive and general-purpose programming language.

(ii) **Likely uses of the program:** Is the program going to be run frequently or occasionally compiled languages are appropriate for frequently run program while interpreted languages for occasionally run programs.

(iii) **Number of people involved in a program development:** A program with subprogram facility will be preferred when people are many and are on the same project. Carefully consideration should be taken to select right caliber of people to work as a group should include Data Processing Manager, Programmers, System Analyst/Designers, and Computer Operators among others. The people involved in the programming development should adhere strictly to budget availability and time-schedule.

### 2. METHOD

**Assessment of Pascal, Basic and C Programming Language**
Apart from the general characteristics of PASCAL, BASIC and C programming languages, the following criteria were used to assess the fitness, appropriateness and choice of a programming language.
(a) **Portability:** This refers to ease of program, transportability to other machines, i.e. it must be less machine-dependent. E.g. while well-defined languages like C is very portable, BASIC is not.

(b) **Efficiency:** This is the speed with which a program can be executed and compiled. Execution efficiency is necessary for large production of program which will be executed many times and compilation time is necessary if the program is to be used for teaching purposes because students’ programs are compiled many times.

(c) **Data types/structure:** This refers to the ability of a language to support a variety of data values such as integers, real, string etc and non-elementary collection of these such as arrays and records. It also includes dynamic data structure such as linked list, queues, stacks and tress.

(d) **Acceptability:** This refers to a language’s usefulness in a wide range of programming application.

(e) **Pedagogy:** This refers to the ease with which a program can be taught or learnt.

(f) **Input/Output facilities:** This is a measure of a language’s support for sequential, indexed, random access files, database as well as information retrieval functions.

(g) **Modularity (ease of language extension):** This is the easiness with which a language extends its primitive operations and data types. Subprograms (procedure, functions and subroutines) are usually provided by most languages for extending their primitive operations.

### 3. RESULTS AND DISCUSSION
The table below is a summary of the results got after test running and running PASCAL, BASIC and C programming languages with many exercises in program writing in the computer laboratory for many years of our research and teaching work.

<table>
<thead>
<tr>
<th>S/N</th>
<th>Criteria</th>
<th>Pascal</th>
<th>BASIC</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Portability</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>Efficiency</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>Data types/structure</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>Acceptability</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>Pedagogy</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>Input/Output facilities</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
7. Modularity

<table>
<thead>
<tr>
<th>Ratings:</th>
<th>5</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>-</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Very good</td>
<td>-</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>-</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Weak</td>
<td>-</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Poor</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Analysis of results
From table 1 above, it is seen that C programming language seem to score high (5) in portability, efficiently and modularity, while BASIC programming score high (5) in pedagogy alone and PASCAL score high (5) in data types/structure and modularity. PASCAL also has (4) for portability and (4) for input/output facilities, but fairly good in efficiency, acceptability and pedagogy, with no weak or poor characteristic remarks. BASIC programming language however, has weak portability and fairly good in efficiency, data types/structures, acceptability and modularity. The C programming language is fairly good in acceptability, pedagogy and input/output facilities with no poor or weak remarks.

4. CONCLUSION
It can be seen from the above analysis that each programming languages of PASCAL, BASIC and C has its own good part, average part and in some cases below or above average part, hence, the choice of program to use depend largely on the type of problem to solve.

REFERENCES


