**CHAPTER TWO**

**LITERATURE REVIEW**

**2.0 INTRODUCTION**Literature review is the assessment of previous work carried out on the same or related projects and extracts relevant points to serve as milestone in the project at hand. Various works were studied and analyzed. The review and the lessons learnt forthwith, informed the choice of technologies employed for the development of this research outputs which is contained in the theoretical background.

**2.1 THEORETICAL BACKGROUND**

The theoretical background gives an overview of the technologies used in developing the system “Student Result Monitoring System” and the general concept of the research related topic as seen by the other researchers. The technology is chosen in other to present a more user friendly system. The system developed comprises of various component written in C#, Transact-SQL (T-SQL), MSQL server along with its Management System used in the development of the database of the system.

**2.2 RESULT MANAGEMENT AND MONITORING**

Emmanuel and Choji (2012) stated that the introduction of computer into information technology has massively improved the information need of organizations. Anigbogu (2000) defined a computer as an electronic device capable of accepting data and instructions, processing the data based on the instructions to generate results or output in such a manner that is yet to be equaled by any other known machine to mankind. Since the computer is capable of accepting data and instructions, store this data and often time this instructions and then generate an output, it is therefore necessary that we harness this quality and use it to our advantage. The process of school administration and in this case result processing is as already stated a time consuming and strenuous exercise prone to errors, if done manually, hence the need for us to seek out ways to lift this burden from the individuals involved in this exercise. Obiniyi and Ezugwu (2010) observed that Student enrolment in tertiary institutions is increasing at a very alarming rate. The increase in students’ population over the years has made the work of administrative officer in charge of processing students’ result a very tiresome exercise to deal with. The rise in the number of students in schools today has made it imperative that we continue to seek out the best and most efficient ways to handle schools and school administration. Mohini and Amar (2011) indicated that Publication of student’s results in the manual system takes a very long time owing to which students remain idle for months together. Sometimes the delay in declaration of result cause heavy losses to the students as generally they cannot join further studies or appear in competitive exams or join jobs because of the non‐availability of examination result in time. Nmaju *et al*. (2013) also observed that many higher institutions in Nigeria still adopt the manual method of managing students’ data which is time consuming and demanding, and are often prone to a variety of errors and disasters. Hence, it brings to the fore the need to properly address how these shortcomings could be resolved and improved. They further stated that the solution to these shortcomings lies in an efficient information management system, or simply, information system. Okonigene et al. (2008) further stated that, with the use of computers for information processing, the following are possible: instant access to students’ personal and course information, instant student information updating, automatic computation of the Grade Point Average (GPA), generation of the graduating students list, monitoring of failed courses, keeping an up-to-date record of the entire student body in the University, storing course information such as course code, course description, course unit, and scores for the purpose of GPA computation, and producing user friendly data entry screens for ease of use. There exist several programming Languages, programming packages and database management systems that can be used to develop a result processing software. Java is a programming language used to build programs that can work on stand-alone computers and on the internet, its primary features are that it is object-oriented and a cross platform language. By cross platform, I means that the programs can run across several platforms such as Microsoft Windows, Apple Macintosh, and Linux etc. Oracle’s Procedural Language/Structured Query Language (PL/SQL) is the choice of programing language used in the development of this application, because it’s Oracle Corporation's procedural extension language for SQL and the Oracle relational database. PL/SQL's general syntax resembles that of Pascal. It is also one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java. PL/SQL supports variables, conditions, loops and exceptions. Arrays are also supported, though PL/SQL collections. Implementations from version 8 of Oracle Database onwards have included features associated with object-orientation. Once the program units have been stored into the database, they become available for execution at a later time.

Some previous work has actually been carried out in this area. Two of such are the work by Moses Ekpenyong and that by Ukem and Onoyom-Ita. In his work, Ekpenyong employs Microsoft Excel spreadsheet program to build an Intelligent Knowledge-Based System (IKBS), making use of various programming facilities provided by that application (Excel). The central issue here is that the programming is hard coded into the cells, and cell referencing is used to monitor and track students performance (cumulative points, etc.). The system has been reported to be working fine. However, it appears to be rather restrictive, and calls for substantial expertise in programming. In the work by Ukem and Onoyom-Ita, Adobe Dreamweaver, an Integrated Development Environment, is used to create the Graphic User Interface and to write the codes. MYSQL Server, a Relational Database Management System, is used to create the database tables and data. Personal Home Page Pre-Processor (PHP) is used to communicate with and manipulate the database. This application, though tested and found to be working as expected, has however not been put to use widely. Apart from these two, there are undoubtedly several other similar applications in existence. There is, however, always room for improvement. This new application is intended to have reduced complexity and greater ease of use, in order to enhance maintainability while still retaining good speed and accuracy.

An automated result processing system that will increase through put and reduce the response time involved in processing students result immediately after they graduate from the institution. The system enables students register courses and in turn, enable lecturers upload students results every semester. Another is the one that examined the inadequacies involved in the manual method of calculating Students CGPA (cumulative grade point average) and proposes a solution by developed a software Application to facilitate the automated processing of the results. The software was developed using PHP (Hypertext processor) scripting language and employing MYSQL Relational Database Management System in designing the database. The developed software was tested and work as expected. With the use of computers for information processing, the following are possible: instant access to students’ personal and course information, instant student information updating, automatic computation of the Grade Point Average (GPA), generation of the graduating students list, monitoring of failed courses, keeping an up-to-date record of the entire student body in the University, storing course information such as course code, course description, course unit, and scores for the purpose of GPA computation.

**2.2.1 GRADE COMPUTATION AND ANALYSIS PROGRAM**

The grading system in use at Nigerian institutions depends on the institution and sometimes on the faculty of the institution.  In addition, grading scales at university level institutions have changed frequently. Grading scales can be 1 to 7, 1 to 5, or A through F, where A is on a 4.0 scale or on a 5.0 scale. The most common scale is now 1 to 5, with 5 being the highest grade obtained. In addition, degrees are awarded in a Class, depending on the grades received. Degrees may be awarded in the First Class, Second Class (Upper Division), Second Class (Lower Division), Third Class and Pass Class. Grading scales for secondary certificates are standard. The West African Examinations Council Senior School Certificate uses a 1 to 9 grading scale. Both grading scales are listed below.

|  |  |  |
| --- | --- | --- |
| **Secondary Grading Scale** | | |
| **Indigenous Grade** | **Description** | **Suggested US Grade** |
| Grade 1 | Excellent | A |
| Grade 2 | Very Good | B+ |
| Grade 3 | Good | B |
| Grade 4,5,6 | Credit | C |
| Grade 7,8 | Pass | D |
| Grade 9 | Fail | F |

|  |  |  |
| --- | --- | --- |
| General Universities Grade Scale | | |
| Indigenous Grade | Division | Suggested US Grade |
| A/AB  70-100 | First Class | A |
| B/BC    60-69 | Second Class, Upper Division | B+ |
| C/CD  50-59 | Second Class, Lower Division | B- |
| D  45-49 | Third Class | C |
| E  40-44 | Pass | C-/D |
| F  0-39 | Fail | F |

|  |  |
| --- | --- |
| **Nigeria Certificate of Education** | |
| **Indigenous Grade** | **Suggested US Grade** |
| Distinction | A |
| Credit | B |
| Merit | C+ |
| Pass | C |
| Pass on Pass/Fail Basis |  |

**2.2.1.1 CREDIT SYSTEM**

Some Nigerian universities have adopted a credit system. This practice has become more typical in recent years, as the transcripts appear to become more like the American transcript.  Within a university, some faculties may use a credit system and others may not. It is not uncommon for the credits earned in one semester to be between 19 and 24 credits.  For this reason, usually a reduction in credits is required to make the units expressed comparable with the number of semester credit hours typically awarded in the US.

**2.2.1.2 TRANSFER CREDIT PRACTICES**

In the past, Nigeria generally had two separate systems: academic university and technical post-secondary. Students were either in the academic or the technical stream.  At the present time, there is some crossover.  Students with the NCE may be admitted to the second or third year of the Bachelor of Education, depending on the grades obtained.  Students with a National Diploma are considered for admission to technical bachelor's degree programs.

**2.2.2 GRADING SYSTEM ANALYSISTop of FormBottom of Form**

At present the students grading system used by all universities in Nigeria is the Five-Point Grading System, established by the Nigerian Universities Commission (NUC) in 1989. The system is also called the “Carry Over” system, because a student is allowed to carry over a course up to two times (or attempt a course a maximum of three times) before a final “Fail” grade is recorded if he/she still does not pass the course. A few exceptions to the rule are the General Studies and Communication Skills (or GSS) courses, which must be passed for a student to graduate. The NUC five-point grading system [1] is shown in Table 1 below.

|  |  |  |  |
| --- | --- | --- | --- |
| SCORE (%) | LETTER GRADE | GRADE VALUE | REMARK |
| 70-100 | A | 5 | Excellent |
| 60-69 | B | 4 | Very Good |
| 50-59 | C | 3 | Good |
| 45-49 | D | 2 | Average |
| 40-44 | E | 1 | Pass |
| 00-39 | F | 0 | Fail |

Table 1: The NUC Five-Point Grading System

The courses offered in a degree programme are allocated a number of credit hours which vary from one course to another, because the courses vary in their needs and scope. Hence some are allocated greater credit hours than others. The measure of performance of a student in any course is given by the grade-points obtained in that course.

The grade-points obtained by a student in any course are determined by multiplying the value of the grade (numeric grade) by the credit hours of the course. The total grade-points are obtained by summing up the grade-points of all the courses offered. The Grade-Point Average (GPA) is computed by dividing total grade-points by the sum of credit hours of all the courses offered in that period. Thus:

Total Grade-Points of the courses offered in the year

GPA = -----------------------------------------------------------------

Summation of the credit hours of the courses in that year

The table below (Table 2) illustrates how a student’s GPA may be calculated manually. If a student obtains the grades as shown at the end of, say, Year One, the computation of GPA is as shown.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course Code | Credit Hours | Scores | Grade | Grade-Points |
| GSS 1101 | 2 | 42 | E | 1 x 2 = 02 |
| GSS 1131 | 2 | 60 | B | 4 x 2 = 08 |
| PHY 1111 | 3 | 46 | D | 2 x 3 = 06 |
| PHY 1141 | 3 | 30 | F | 0 x 3 = 00 |
| PHY 1180 | 2 | 75 | A | 5 x 2 = 10 |
| MTH 1111 | 3 | 50 | C | 3 x 3 = 09 |
| CHM 1011 | 3 | 70 | A | 5 x 3 = 15 |
| GSS 1102 | 2 | 58 | C | 3 x 2 = 06 |
| GSS 1112 | 2 | 45 | D | 2 x 2 = 04 |
| GSS 1122 | 2 | 41 | E | 1 x 2 = 02 |
| PHY 1152 | 3 | 68 | B | 4 x 3 = 12 |
| MTH 1352 | 3 | 80 | A | 5 x 3 = 15 |
| CHM 1022 | 3 | 67 | B | 4 x 3 = 12 |
|  | **33** |  |  | **101** |

Table 2: Calculation of GPA

Total Grade-Points = 101

Total Credit Hours = 33

Grade Point Average (GPA) = 101/33

= 3.06

The Cumulative Grade-Point Average (CGPA) in any year is obtained by dividing the cumulative sum of the total grade-points over the years by the cumulative sum of the credit hours over the same period. Thus,

Cumulative sum of Total Grade-Points of the courses from Year One to the Present Year

CGPA = ------------------------------------------------------------------------------------------------------

Cumulative sum of the Credit Hours of the courses from Year One to the Present Year

The CGPA is a very important measure, as this is what determines whether a student can move on to the next level or be made to repeat a year or even withdraw totally from the programme. The final CGPA determines the class of degree awarded to the candidate on eventual completion of the programme. Obviously, for the first year, the CGPA is equal to the GPA.

Manually calculating and tracking CGPA for each student is rather laborious, especially with increasing number of students, and is prone to error. With a computer the task becomes much easier, faster, and more accurate. All that is needed is a suitable software application to run on the hardware, and a solution is established. With the appropriate software in place, the system would simply capture raw scores as entered by individual lecturers for various students in the different courses, and then process accordingly.

**2.2.3 ADVANTAGES AND DISADVANTAGE OF GRADING SYSTEM**

**ADVANTAGE**

According to many teachers and students, grading system is a good initiative and providing valuable advantages such as:

* **Reduced score pressure:** The grading system has reduced the scoring pressure of students. According to grading method, the actual marks of a subject do not get mentioned on transcripts but only the grades. Which means that now students do not have to score specific marks for good impression but only target a grade that can be achieved easily.
* **Advanced Grading Pattern:** The grading system is based on an advanced marking pattern. Unlike old times, now students get assessed on the basis of monthly assignments, objective and subjective tests, presentations, quizzes and final term paper. That one paper marking system is long gone and now students can perform a little in each task and score overall a good grade.
* **Identification of Weakness and Strengths:** Grading system let the students identify their weaknesses and strengths. Teachers grade children’s skills of writing, reading and listening. Which means that students can easily know about their lacking abilities through the assigned grades and work on them to improve overall performance.
* **Easier Studies:** The grading system has made studies easier in many ways. Those who only want to pass can simply make lesser efforts to achieve passing grade. And those who want to score higher can divide the effort per task and achieve the overall targeted grade easily.

**DISADVANTAGES**

Just like advantages, grading system has also some disadvantages such as:

* **Decreased Performance:** Due to grading system, children tend to perform less. They know that they will achieve the targeted grade even by making a few mistakes. For instance, if a student targets the grade A that involves the range from 90% to 100%. He will know that even if he scores 90, grade A will be assigned hence he doesn’t require to score 95 or 100.
* **Demotivation:** Grading system demotivates the students who perform higher because they stand equal to those making less efforts. For instance, grade A will be assigned to all those scoring from 90 to 100. So students who made no mistakes and those who made a few, all will stand equally at one grade.
* **Increased Lethargy:** As grading system has divided the marks among different tasks such as assignments, presentations and final exams, the students become lethargic due to it. They score enough in assignments and projects and become lesser active in final exams.

**2.3 PACKAGES USED FOR DESIGNING OF RESULTS PROCESSING**

Different Programming Language, Programming Packages and Database management system can be use to develop result processing software for computing students GPA (Grade Point Average) and CGPA (Cumulative Grade Point Average). Microsoft Excel spreadsheet program can be used to build an Intelligent Knowledge-Based System (IKBS), making use of various programming facilities provided by that application (Excel). The programming is hard coded into the cells, and cell referencing which could be applied to monitor and track students’ performance such as cumulative points (Ekpenyong, 2008). Personal Home Page Pre-Processor (PHP) is used to communicate with and manipulate the database.

Adobe Dreamweaver, an Integrated Development Environment, is used to create the Graphic User Interface and to write the codes. MYSQL Server, a Relational Database Management System, is used to create the database tables and data. This application, though tested and found to be working as expected, has however not been put to use widely. (Ukem and Onoyom-Ita, 2011). Java is a programming language use to build programs that can work on stand-alone computers and on the internet, its primary features are that it is object-oriented and a cross platform language. By cross platform, it means that the  
programs can run across several platforms such as Microsoft Windows, Apple Macintosh, and Linux. MYSQL, a Relational Database Management System (RDBMS) is used to create database tables and data. MySQL is very fast, reliable, and easy to use, and its connectivity, speed, and security make it highly suited for accessing databases.  
(Ukem, and Ofoegbu, 2012).

**2.4 MODES OF RESULT PROCESSING SYSTEM**

Basically, there are two modes of result processing namely, the manual mode system and electronic mode system. Result can be processed manually with the use of primitive tools such as tabs and calculators. The processing of result by machines in general in such a way as to reduce to a minimum, the need for manual processing is referred to as automatic data processing. Result processing is mainly done by computers, and the methodology is referred to as electronic data processing.

Generally the result processing methods are:

* Manual System
* Electronic mode system

**2.4.1 MANUAL SYSTEM**

This is the case where the result processing and generation for the departments are carried manually. This manual system of result processing, as observed in this study has many problems, among which are:

* Delays in carrying out these activities due to tediousness.
* Missing results due to carelessness or misplacement.
* Miscalculation of student‟s results.

Due to the manual system, the desk officers both in Exams and in Records and Statistics who are in charge of result processing find it difficult to edit documented results and cancel out some errors such as omission or wrong entries. Even though these corrections are made, the paper work will look mutilated (Mohini and Amar 2011). Grey (2010) usually manual processing or manual mode generating of results usually ends with correction of errors and when they are given out to students, they look untidy.

**2.4.2 AUTOMATION OF ACTIVITIES**

This is where steps to take when automating a system more especially when it has to do with the result processing is discussed. Examination results of students are one of the most important elements of an institution. Processing of this data must be under effective management, to ensure:

1. Substantial saving of time and decreasing cost.
2. Efficiency and speedy output when results are processed

There are many other steps to take when automating a process. Automation offers many advantages which include an improved process while decreasing cost. When automating, for outside help if indeed such need arises, to ensure the system runs smoothly. Such expertise can ensure the following:

1. Successful automation can improve quality, increase throughput, and reduce costs. A capable integrator recognizes what to automate and what not to automate and make sure that simple parts meet necessary tolerance for automation.
2. Provide the primary goals you want to accomplish with automation (reducing cost, increasing throughput, improving quality and efficiency).

**2.4.3 RESULT PROCESSING TECHNIQUES**

**2.4.3.1 BATCH PROCESSING**

It is a technique by which items to be processed must be coded and collected into groups or batches prior to processing. A batch consists of either a convenient number of records or a collection of records relating to a given period, such as daily, weekly, monthly, etc, accumulated until a sufficient number are present to justify mass updating of a master file.

Each batch is used to update a master file using a file maintenance program and an output produced. This type of processing is suitable for semester results and business applications, in which output is produced according to a predetermined processing cycle. The processing for this type of system is ordinarily defined by the frequency with which the master file must be updated.

The average cost of processing result is low. No additional hardware, such as terminals or direct access storage device is needed. The system lends itself to control, as trailer records can be inputted which contain totals of important fields. The methods are secure as updating by copying can be used. It does, however, impose rigidity on the school. Weekly or monthly deadlines have to be met and peaks of work occur as results are only produced once a period.

**2.4.3.2 REAL TIME PROCESSING**

This is the processing of result in a sufficiently rapid manner so that the results of this processing are available in time as at when needed. The processing goes on, it is applied to the master file and result is obtained from the system on demand.

**2.4.3.3 ONLINE TRANSACTION PROCESSING**

It concerns the facilitation and management of transaction oriented applications (such as result generation) typically for data entry and retrieval. Automated transaction processing has been used to refer to processing in which the system responds immediately to users request.

**2.4.3.4 TIME-SHARING PROCESSING**

The system interacts with many users, giving each of them fast individual attention on a time slice basis.

Result TransmissionResults can be fed into the computer in any of the three forms:

* Simplex: A simplex line allows results to be transmitted in one direction and one direction only. These simplex lines may be used to connect results from collection terminals to distant computer systems, thus speeding the flow of results from these online input terminals.
* Half duplex: A half-duplex line is one in which results can be transmitted in either direction, but in only one direction at a time, the flow of result in one direction must stop before results may flow in the opposite direction. Such lines may be used in time sharing operations.
* Full duplex: a duplex line is one in which results can be transmitted simultaneously in both directions. Inputs/outputs (I/O) operations can be completed simultaneously. Such lines serve best in real time system handling large volume of I/O transmission thereby, encouraging rapid distribution of results.

**2.5 THEORETICAL TECHNOLOGY FOR DEVELOPMENT**

**2.5.1 C#, Programming Language**

C# [(/si: ʃɑːrp/)](https://en.wikipedia.org/wiki/Help:IPA/English) is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) encompassing [strong typing](https://en.wikipedia.org/wiki/Strong_typing), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [declarative](https://en.wikipedia.org/wiki/Declarative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), [generic](https://en.wikipedia.org/wiki/Generic_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) ([class](https://en.wikipedia.org/wiki/Class_(computer_science))-based), and [component-oriented](https://en.wikipedia.org/wiki/Component-based_software_engineering) programming disciplines. It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) within its [.NET](https://en.wikipedia.org/wiki/.NET_Framework) initiative and later approved as a standard by [Ecma](https://en.wikipedia.org/wiki/Ecma_International) (ECMA-334) and [ISO](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO/IEC 23270:2006). C# is one of the programming languages designed for the [Common Language Infrastructure](https://en.wikipedia.org/wiki/Common_Language_Infrastructure).

C# is a general-purpose, object-oriented programming language. Its development team is led by [Anders Hejlsberg](https://en.wikipedia.org/wiki/Anders_Hejlsberg). The most recent version is C# 7.2, which was released in 2017 along with [Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) 2017 version 15.5.

**2.5.1.1 DESIGN GOAL**

The ECMA standard lists these design goals for C#:

* The language is intended to be a simple, modern, general-purpose, [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) language.
* The language, and implementations thereof, should provide support for software engineering principles such as [strong type](https://en.wikipedia.org/wiki/Strong_type)checking, array [bounds checking](https://en.wikipedia.org/wiki/Bounds_checking), detection of attempts to use [uninitialized variables](https://en.wikipedia.org/wiki/Uninitialized_variable), and automatic [garbage collection](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). Software robustness, durability, and programmer productivity are important.
* The language is intended for use in developing [software components](https://en.wikipedia.org/wiki/Software_components) suitable for deployment in distributed environments.
* Portability is very important for source code and programmers, especially those already familiar with [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B).
* Support for [internationalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) is very important.
* C# is intended to be suitable for writing applications for both hosted and [embedded systems](https://en.wikipedia.org/wiki/Embedded_system), ranging from the very large that use sophisticated [operating systems](https://en.wikipedia.org/wiki/Operating_system), down to the very small having dedicated functions.

**2.5.1.2 HISTORY**

During the development of the .NET Framework, the [class libraries](https://en.wikipedia.org/wiki/Base_Class_Library) were originally written using a [managed code](https://en.wikipedia.org/wiki/Managed_code) compiler system called *Simple Managed C* (SMC). In January 1999, [Anders Hejlsberg](https://en.wikipedia.org/wiki/Anders_Hejlsberg) formed a team to build a new language at the time called Cool, which stood for "[C-like](https://en.wikipedia.org/wiki/C-like) Object Oriented Language".[[19]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-computerworld2008-21) Microsoft had considered keeping the name "Cool" as the final name of the language, but chose not to do so for trademark reasons. By the time the .NET project was publicly announced at the July 2000 [Professional Developers Conference](https://en.wikipedia.org/wiki/Professional_Developers_Conference), the language had been renamed C#, and the class libraries and [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET) runtime had been ported to C#.

Hejlsberg is C#'s principal designer and lead architect at Microsoft, and was previously involved with the design of [Turbo Pascal](https://en.wikipedia.org/wiki/Turbo_Pascal), [Embarcadero Delphi](https://en.wikipedia.org/wiki/Embarcadero_Delphi) (formerly CodeGear Delphi, Inprise Delphi and Borland Delphi), and [Visual J++](https://en.wikipedia.org/wiki/Visual_J%2B%2B). In interviews and technical papers he has stated that flaws[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] in most major programming languages (e.g. [C++](https://en.wikipedia.org/wiki/C%2B%2B), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [Delphi](https://en.wikipedia.org/wiki/Embarcadero_Delphi), and [Smalltalk](https://en.wikipedia.org/wiki/Smalltalk)) drove the fundamentals of the [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), which, in turn, drove the design of the C# language itself.

Since the release of C# 2.0 in November 2005, the C# and Java languages have evolved on increasingly divergent trajectories, becoming two very different languages. One of the first major departures came with the addition of [generics](https://en.wikipedia.org/wiki/Generic_programming) to both languages, with vastly different implementations. C# makes use of [reification](https://en.wikipedia.org/wiki/Reification_(computer_science)) to provide "first-class" generic objects that can be used like any other class, with code generation performed at class-load time.[[25]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-27) Furthermore, C# has added several major features to accommodate functional-style programming, culminating in the [LINQ](https://en.wikipedia.org/wiki/Language_Integrated_Query) extensions released with C# 3.0 and its supporting framework of [lambda expressions](https://en.wikipedia.org/wiki/Lambda_expressions), [extension methods](https://en.wikipedia.org/wiki/Extension_method), and [anonymous types](https://en.wikipedia.org/wiki/Anonymous_type).

These features enable C# programmers to use functional programming techniques, such as [closures](https://en.wikipedia.org/wiki/Closure_(computer_science)), when it is advantageous to their application. The LINQ extensions and the functional imports help developers reduce the amount of [boilerplate code](https://en.wikipedia.org/wiki/Boilerplate_code) that is included in common tasks like querying a database, parsing an xml file, or searching through a data structure, shifting the emphasis onto the actual program logic to help improve readability and maintainability.

**2.5.1.3 NAME**

The name "C sharp" was inspired by musical notation where a [sharp](https://en.wikipedia.org/wiki/Sharp_(music)) indicates that the written note should be made a [semitone](https://en.wikipedia.org/wiki/Semitone) higher in [pitch](https://en.wikipedia.org/wiki/Pitch_(music)). This is similar to the language name of [C++](https://en.wikipedia.org/wiki/C%2B%2B), where "++" indicates that a variable should be incremented by 1. The sharp symbol also resembles a [ligature](https://en.wikipedia.org/wiki/Typographic_ligature) of four "+" symbols (in a two-by-two grid), further implying that the language is an increment of C++.

**2.5.1.4 SYNTAX**

The core syntax of C# language is similar to that of other C-style languages such as C, C++ and Java. In particular:

* Semicolons are used to denote the end of a statement.
* [Curly brackets](https://en.wikipedia.org/wiki/Curly_brackets) are used to group statements. Statements are commonly grouped into methods (functions), methods into classes, and classes into [namespaces](https://en.wikipedia.org/wiki/Namespaces).
* Variables are assigned using an [equals sign](https://en.wikipedia.org/wiki/Equals_sign), but compared using [two consecutive equals signs](https://en.wikipedia.org/wiki/%3D%3D).
* [Square brackets](https://en.wikipedia.org/wiki/Square_brackets) are used with [arrays](https://en.wikipedia.org/wiki/Array_data_structure), both to declare them and to get a value at a given index in one of them.

**2.5.1.5 EXAMPLES**

The following is a very simple C# program, a version of the classic "[Hello world](https://en.wikipedia.org/wiki/Hello_world)" example:

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello, world!");

}

}

What will display on the program is:

Hello, world!

Each line has a purpose:

**Using System;**

The above line of code tells the compiler to use System as a candidate prefix for types used in the source code. In this case, when the compiler sees use of the Console type later in the source code, it tries to find a type named Console, first in the current assembly, followed by all referenced assemblies. In this case the compiler fails to find such a type, since the name of the type is actually System.Console. The compiler then attempts to find a type named System.Console by using the System prefix from the using statement, and this time it succeeds. The using statement allows the programmer to state all candidate prefixes to use during compilation instead of always using full type names.

**Class Program**

Above is a [class](https://en.wikipedia.org/wiki/Class_(computer_science)) definition. Everything between the following pair of braces describes Program.

**static void main(string[] args)**

This declares the class member method where the program begins execution. The .NET runtime calls the Main method. (Note: Main may also be called from elsewhere, like any other method, e.g. from another method of Program.) The [static keyword](https://en.wikipedia.org/wiki/Method_(computer_programming)#Static_methods) makes the method accessible without an instance of Program. Each console application's Main entry point must be declared static. Otherwise, the program would require an instance, but any instance would require a program. To avoid that irresolvable [circular dependency](https://en.wikipedia.org/wiki/Circular_dependency), C# compilers processing [console applications](https://en.wikipedia.org/wiki/Console_application) (like that above) report an error, if there is no static Main method. The void keyword declares that Main has no [return value](https://en.wikipedia.org/wiki/Return_value).

**Console.WriteLine("Hello, world!");**

This line writes the output. Console is a static class in the System namespace. It provides an interface to the standard input, output, and error streams for console applications. The program calls the Console method WriteLine, which displays on the console a line with the argument, the string "Hello, world!".

**2.5.1.6 C# Version**

**C# Version 1**

C# version 1 really did look an awful lot like Java.  As [part of its stated design goals for ECMA](http://feeldotneteasy.blogspot.com/2011/01/c-design-goals.html), it sought to be a “simple, modern, general purpose object-oriented language.”  At the time, it could have done worse thank looking like Java in order to achieve those goals.

But if you looked back on C# 1.0 now, you’d find yourself a little dizzy.  It lacked the built in async capabilities and some of the slick functionality around generics that we take for granted.  As a matter of fact, it lacked generics altogether.  And [Linq](https://msdn.microsoft.com/en-us/library/bb308959.aspx)?  Nope.  That would take some years to come out.

**C# Version 2**

Now things start to get interesting.  Let’s take a look at some major features of C# 2.0, released in 2005, along with Visual Studio 2005.  (Check out [the book by NDepend creator Patrick Smacchia](http://www.ndepend.com/practicalbook) about .NET 2.0.)

* [Generics](https://www.tutorialspoint.com/csharp/csharp_generics.htm)
* [Partial types](https://www.dotnetperls.com/partial)
* [Anonymous methods](https://www.tutorialspoint.com/csharp/csharp_anonymous_methods.htm)
* [Nullable types](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/nullable-types/)
* [Iterators](https://msdn.microsoft.com/en-us/library/65zzykke(v=vs.100).aspx)
* [Covariance and contravariance](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/covariance-contravariance/)

While Microsoft may have started with a pretty generic object-oriented language, C# Version 2 changed that in a hurry.  Once they had their feet under them, they went after some serious developer pain points.  And they went after them in a big way.

With generics, you have types and methods that can operate on an arbitrary type while still retaining type safety.  So, for instance, having a List<T> lets you have List<string> or List<int>  and perform type safe operations on those strings or ints while you iterate through them.  This certainly beats creating ListInt inheritors or casting from Object for every operation.

**C# Version 3**

C# Version 3 came in late 2007, along with Visual Studio 2008, though the full boat of language features would actually come with C# Version 3.5.  And what a version this proved to be.  I would go so far as to say that this established C# as a truly formidable programming language.  Let’s take a look at some major features in this version.

* [Auto implemented properties](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/auto-implemented-properties)
* [Anonymous types](http://www.c-sharpcorner.com/UploadFile/ff2f08/anonymous-types-in-C-Sharp/)
* [Query expressions](https://docs.microsoft.com/en-us/dotnet/csharp/linq/query-expression-basics)
* [Lambda expression](http://www.daedtech.com/introduction-to-c-lambda-expressions/)
* [Expression trees](https://blogs.msdn.microsoft.com/charlie/2008/01/31/expression-tree-basics/)
* [Extension methods](https://www.codeproject.com/Tips/709310/Extension-Method-In-Csharp)

**C# Version 4**

C# Version 4 would have had a difficult time living up to the groundbreaking status of version 3.  With version 3, Microsoft had moved the language firmly out from the shadow of Java and into prominence.  The language was quickly becoming elegant.

The next version did introduce some cool stuff, though.

* [Dynamic binding](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/dynamic)
* [Named/optional arguments](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/named-and-optional-arguments)
* [Generic covariant and contravariant](https://docs.microsoft.com/en-us/dotnet/standard/generics/covariance-and-contravariance)
* [Embedded interop types](https://stackoverflow.com/questions/20514240/whats-the-difference-setting-embed-interop-types-true-and-false-in-visual-studi)

**C# Version 5**

With C# Version 5, Microsoft released a very focused version of the language.  They put nearly all of their effort for that version into another pretty groundbreaking language concept.  Here is the major features list.

* [Asynchronous members](https://msdn.microsoft.com/library/hh191443(vs.110).aspx)

**C# Version 6**

With versions 3 and 5, Microsoft had done some pretty impressive stuff in an OO language.  (Version 2 did as well, but they were fast following Java with those language features.)  With version 6, they would go away from doing a dominant killer feature and instead release a lot of features that delighted users of the language.  Here are some of them.

* [Static imports (a la Java)](http://geekswithblogs.net/BlackRabbitCoder/archive/2015/04/16/c.net-little-wonders-static-using-statements-in-c-6.aspx)
* [Exception filters](https://www.thomaslevesque.com/2015/06/21/exception-filters-in-c-6/)
* [Property initializers](http://geekswithblogs.net/WinAZ/archive/2015/06/30/whatrsquos-new-in-c-6.0-auto-property-initializers.aspx)
* [Expression bodied members](https://lostechies.com/jimmybogard/2015/12/17/c-6-feature-review-expression-bodied-function-members/)
* [Null propagator](https://davefancher.com/2014/08/14/c-6-0-null-propagation-operator/)
* [String interpolation](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/interpolated-strings)
* [nameof operator](https://stackoverflow.com/questions/31695900/what-is-the-purpose-of-nameof)
* [Dictionary initializer](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/how-to-initialize-a-dictionary-with-a-collection-initializer)

**C# Version 7**

Finally, we arrive at C# version 7.  That’s the current version as of the writing of this post.  This has some evolutionary and cool stuff in the vein of C# 6, but without the compiler as a service.  Here are some of the new features.

* [Out variables](http://www.c-sharpcorner.com/article/out-variables-in-c-sharp-7-0/)
* [Tuples and deconstruction](https://www.thomaslevesque.com/2016/08/23/tuple-deconstruction-in-c-7/)
* [Pattern matching](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-7#pattern-matching)
* [Local functions](http://www.infoworld.com/article/3182416/application-development/c-7-in-depth-exploring-local-functions.html)
* [Expanded expression bodied members](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-7#more-expression-bodied-members)
* [Ref locals and returns](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-7#ref-locals-and-returns)

**2.5.2 TRANSACT-SQL** (**T-SQL**)

It is [Microsoft](https://en.wikipedia.org/wiki/Microsoft)'s and [Sybase](https://en.wikipedia.org/wiki/Sybase)'s proprietary extension to the [SQL](https://en.wikipedia.org/wiki/SQL) (Structured Query Language) used to interact with [relational databases](https://en.wikipedia.org/wiki/Relational_database). T-SQL expands on the SQL standard to include [procedural](https://en.wikipedia.org/wiki/Procedural_programming) programming, [local variables](https://en.wikipedia.org/wiki/Local_variable), various support functions for string processing, date processing, mathematics, etc. and changes to the [DELETE](https://en.wikipedia.org/wiki/Delete_(SQL)) and [UPDATE](https://en.wikipedia.org/wiki/Update_(SQL)) statements.

Transact-SQL is central to using [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server). All applications that communicate with an instance of SQL Server do so by sending Transact-SQL statements to the server, regardless of the user interface of the application. [Stored procedures](https://en.wikipedia.org/wiki/Stored_procedure) in SQL Server are executable server-side routines. The advantage of stored procedures is the ability to pass parameters.

**2.5.3 MICROSOFT SQL SERVER**

Microsoft SQL Server is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application)—which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](https://en.wikipedia.org/wiki/Concurrent_user).

The history of Microsoft SQL Server begins with the first Microsoft SQL Server product – SQL Server 1.0, a 16-bit server for the [OS/2](https://en.wikipedia.org/wiki/OS/2) operating system in 1989 - and extends to the current day.

Milestones

* MS SQL Server for OS/2 began as a project to port [Sybase SQL Server](https://en.wikipedia.org/wiki/Sybase_SQL_Server) onto OS/2 in 1989, by [Sybase](https://en.wikipedia.org/wiki/Sybase), [Ashton-Tate](https://en.wikipedia.org/wiki/Ashton-Tate), and [Microsoft](https://en.wikipedia.org/wiki/Microsoft).
* SQL Server 4.2 for NT is released in 1993, marking the entry onto [Windows NT](https://en.wikipedia.org/wiki/Windows_NT).
* SQL Server 6.0 is released in 1995, marking the end of collaboration with [Sybase](https://en.wikipedia.org/wiki/Sybase); Sybase would continue developing their own variant of *SQL Server*, Sybase [Adaptive Server Enterprise](https://en.wikipedia.org/wiki/Adaptive_Server_Enterprise), independently of Microsoft.
* SQL Server 7.0 is released in 1998, marking the conversion of the source code from C to C++.
* SQL Server 2005, released in 2005, finishes the complete revision of the old Sybase code into Microsoft code.
* SQL Server 2017, released in 2017, adds Linux support for these Linux platforms: [Red Hat Enterprise Linux](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux), [SUSE Linux Enterprise Server](https://en.wikipedia.org/wiki/SUSE_Linux_Enterprise_Server), [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) & [Docker Engine](https://en.wikipedia.org/wiki/Docker_(software)).[[3]](https://en.wikipedia.org/wiki/Microsoft_SQL_Server#cite_note-3)

Currently

As of October 2017 the following versions are supported by Microsoft:

* SQL Server 2008
* SQL Server 2008 R2
* SQL Server 2012
* SQL Server 2014
* SQL Server 2016
* SQL Server 2017