



**FULL NAME:**

Diego Monroy Minero

**STUDEN ID:**

2109110

**QUARTER:**

1° (January – April)

**GROUP:**

1-A

**MAJOR:**

Data

**UNIT:**

First Unit

# Index

Cover Page	1
Index	2
Introduction	3
Stages of program compilation	4
Levels of programming	5
Conclusion	6
Bibliographic References	7

# Introduction

In this research work we will notice the stages of program compilation in a more detailed way, starting with lexical analysis until reaching the optimization of the code.

We are also going to review the levels of programming, starting with the characteristics of them, differences between them and finishing with some examples of the levels.

# Stages of program compilation

1. **Lexical Analysis:** This is the first stage of compilation process, where the source code is broken into atomic data, that is the smallest meaningful item of data, and it is used to create tokens. This mean that all the spaces and comments that we, as programmers, use so the code can be more understandable are removed because this is not executable. They also need to follow a set of rules, for example to not start with a special character, such as a number.
2. **Syntax analysis:** In this stage, the compiler makes two principal things: check the syntax of the code and create an abstract syntax tree (AST).  
In order to create the AST, the code needs to be without semicolons, brackets, etc., So the first subprocess is to check is the syntax. If there is any syntax error, messages are produced. Tokens also are checked to see if they match the rules of the programming language. Once the syntax is checked, the AST is created, and this maps the structure of the program.
3. **Code Generation:** In this stage the machine code is generated, which means that the tokens are used to create a low-level machine code, and this is useful for the translator to break the high-level language down into simple low-level language. Also, another program is created, which contains the binary equivalent of the source code so it can be executable.
4. **Optimization:** This stage is important because the optimizer identifies redundant, repeated, and never called parts of the code to remove them and make the program more efficient.

For example:

```
z = 12
x = 10
y = x
print x
```

This stage would do the following

```
x = 10
print x
```

# Levels of programming

- **High-level Language:** The programming languages that have been developed to make it easier for the programmer to read and write programs are called high-level languages. The reason is because code is more likely to human languages with some features like: selection and iteration constructs; Boolean operators; identifiers using unlimited number of alphabetic and special characters, data structures.

Some advantages of using high-level programming languages are: they're relatively easy to learn; they allow you to use mathematical operators making it easy to read, understand, debug and maintain; shows how the algorithm will solve a problem in a clearer and more straightforward way.

- **Low-level Language:** This kind of language is not user-friendly because it different uses than high-level languages: it is used in computer forensics, cyber-attacks and debugging but in a very specific way. Since this kind of language is more likely to the machine code, only people with very specialist knowledge can use it and it has limited number of programming constructs. An example of a low-level code is Assembly, that is a language used to write code to a processor. The assembler is used to write the code into machine code.
- **Differences between Low-level and High-level Language:** While a low-level language is machine-friendly, a high-level language is more user-friendly. Both need a program to convert the code into machine code, but the low-level language needs an assembler, and the high-level language needs a compiler to make this task. High-level language is better at debugging and portability, while a low-level language is better in memory efficiency.
- **Examples of programming languages in the levels of programming:**
  - High-level Languages: Python, Java, Visual Basic, C++, etc.,
  - Low-level Languages: Assembly, Machine Code.

# Conclusion

As a conclusion from this research work, we can infer that programming is not simply writing code, it takes a lot more (at least for the computer) to execute a program. Besides, now we can understand more about why sometimes it takes a lot to just run a program, and it is because the computer needs to change the code to binary code so it can understand all the task that the program is supposed to do.

On the other hand, we also learned about the levels of programing and that the programming language level is not defined on how many functions it has, it is simply reduced on how it is written, for humans or for computers to understand it.

# Bibliographic References

- Stages of compilation - Program construction - Eduqas - GCSE Computer Science Revision - Eduqas. (s. f.). BBC Bitesize. Recuperado 7 de enero de 2022, de <https://www.bbc.co.uk/bitesize/guides/zmthsrd/revision/3>
- The National Centre for Computing Education. (s. f.). Stages of compilation. isaac computer science. Recuperado 7 de enero de 2022, de [https://isaacomputerscience.org/concepts/sys\\_trans\\_stages?examBoard=all&stage=all](https://isaacomputerscience.org/concepts/sys_trans_stages?examBoard=all&stage=all)
- TeachAllAboutIT. (2020, 29 febrero). Stages of Compilation | TeachAllAboutIT. TeachAllAboutIT | Computer Science Tuition, Resources, & Live Lessons. Recuperado 7 de enero de 2022, de <https://teachallaboutit.school/stages-of-compilation/>
- 3.2.9 Classification of Programming Languages. (s. f.). Bournetocode. Recuperado 7 de enero de 2022, de [https://bournetocode.com/projects/GCSE\\_Computing\\_Fundamentals/pages/3-2-9-class\\_prog\\_langs.html](https://bournetocode.com/projects/GCSE_Computing_Fundamentals/pages/3-2-9-class_prog_langs.html)
- Classification of Programming Languages: Low-Level and High-Level - javatpoint. (s. f.). Www.Javatpoint.Com. Recuperado 7 de enero de 2022, de <https://www.javatpoint.com/classification-of-programming-languages>