

**Name:**

**Shoaib Akhtar**

**Roll no:**

**20p-0147**

**Section:**

**BSCS-6B**

## **Assignment # 01**

### **Evolution of Programming languages**

- o Mention the form of input data, such a punching cards**
- o Also mention the level of language such as high level language, low level language or machine level language**
- o If a language is designed for a particular task such as web-designing, research, engineering, write the name of the language with the associated area of field**

Here is a brief overview of the evolution of programming languages, along with the form of input data and the associated area of field, where applicable:

- 1. First-generation language (1GL) or machine language:** Machine language was the first programming language, and it consisted of binary code, which could only be understood by the computer. Input data was typically entered using punch cards.
- 2. Second-generation language (2GL) or assembly language:** Assembly language uses symbolic representation of machine code, making it easier to read and write. Input data was still typically entered using punch cards.
- 3. Third-generation language (3GL) or high-level language:** Third-generation languages are high-level programming languages that are more user-friendly and

are designed to be hardware independent. Examples of 3GLs include COBOL, BASIC, FORTRAN, C, C++, Java, and Python. Input data was typically entered using a keyboard or other input device.

4. Fourth-generation language (4GL): Fourth-generation languages are used for database management and other data-intensive applications. They use a declarative rather than a procedural approach to programming. Examples of 4GLs include SQL and R. Input data was typically entered using a keyboard or other input device.
5. Fifth-generation language (5GL): Fifth-generation languages are still under development and are used in artificial intelligence and expert systems. They rely on inference and deduction to solve problems. Examples of 5GLs include Mercury and Prolog. Input data is typically in the form of natural language or other high-level input formats.

As programming languages have evolved, the form of input data has also changed from punch cards to keyboard input and other input devices. Additionally, programming languages have moved from low-level languages like machine language and assembly language to high-level languages like C, Java, and Python. Finally, some programming languages are designed for specific tasks like web-designing (e.g., HTML, CSS, JavaScript), research (e.g., R), or engineering (e.g., MATLAB).

## **Generations of programming languages**

- o Mention details of all generation (from 1st generation of languages (1GL up to 5GL))**
- o Mention few names of programming language in each generation**
- o Effect of Generations of Programming languages on the performance of computer**

The generations of programming languages refer to the evolution of programming languages over time. They are categorized based on the level of abstraction from machine language to high-level programming languages. Here are the details of each generation:

1. First-generation language (1GL) or machine language: The first generation of programming languages was machine language or binary code, which directly corresponds to the hardware of a computer.

2. Second-generation language (2GL) or assembly language: Assembly language uses symbolic representation of machine code, making it easier to read and write. However, assembly language is still hardware specific.
3. Third-generation language (3GL) or high-level language: Third-generation languages are high-level programming languages that are more user-friendly and are designed to be hardware independent. They use keywords, variables, and other features that make it easier to write programs. Examples of 3GLs include COBOL, BASIC, FORTRAN, C, C++, Java, and Python.
4. Fourth-generation language (4GL): Fourth-generation languages are used for database management, and other data-intensive applications. They use a declarative rather than a procedural approach to programming. Examples of 4GLs include SQL and R.
5. Fifth-generation language (5GL): Fifth-generation languages are still under development and are used in artificial intelligence and expert systems. They rely on inference and deduction to solve problems. Examples of 5GLs include Mercury and Prolog.

The effect of generations of programming languages on the performance of computers is profound. As programming languages have evolved from machine language to high-level languages, the time and effort required to write programs has decreased, and the readability of code has improved. This, in turn, has led to an increase in the complexity of programs that can be written, making it possible to solve more complex problems. With each generation, programs become more efficient and optimized, leading to better computer performance. Additionally, hardware improvements over time have made it possible to handle more complex programs and more data, which has further enhanced computer performance.

### **What are scripting languages? Elaborate with an example.**

A scripting language is a programming language that is designed to be used for tasks that involve automation, customization, and extension of software applications. These languages are used to create scripts that are interpreted rather than compiled, which means that they do not need to be compiled into machine code before they can be executed. Scripting languages are generally high-level languages, which makes them easier to use than low-level programming languages like C and assembly language.

Scripting languages are used for a variety of tasks, including automating repetitive tasks, creating website templates, and processing data. Some of the popular scripting languages include:

1. JavaScript: JavaScript is a popular scripting language that is used for creating interactive web pages. It is used in web development to create interactive user interfaces, animations, and dynamic web content.

2. Python: Python is a high-level scripting language that is used for a wide range of applications, including web development, machine learning, data analysis, and more. Python is known for its simplicity and ease of use, which makes it an ideal choice for beginners.
3. Ruby: Ruby is a scripting language that is used for web development, software development, and system administration. Ruby on Rails, a web development framework that uses Ruby, is used by many web developers around the world.
4. PHP: PHP is a server-side scripting language that is used for web development. It is used to create dynamic web pages and web applications.

Scripting languages are often used in conjunction with other programming languages to accomplish specific tasks. For example, JavaScript is often used with HTML and CSS to create interactive web pages, while Python is often used in combination with data analysis libraries like NumPy and Pandas to process large data sets. The ease of use and versatility of scripting languages make them an essential tool for many software developers and IT professionals.