Introduction to the basic data types in programming

[Nagendra Singh, Manish Kumar, Vidhan Singh, Dayanand Choudhary]

Abstract: In the era of technology, the need for developing a language to communicate with computers led to the development of different programming languages. Computers cannot understand the commands in the normal human languages so we needed programming languages to communicate with computers and develop different apps, website and software programs. A crucial part for programming languages are data types. Data types are the classifications of data which tells the compiler or interpreter how the developer wants to insert the data.

Keywords: Data types, numeric data types, text data types, Boolean data types, composite data types, special data types, type casting and type coercion, dynamic and static typing.

1. Introduction

Programming languages are a type of medium used to communicate and give command to computers. To use the different types of programming languages effectively and make the compiler understand the task more efficiently one should know how data is classified in languages in the form of data types. Basic data types are the introduction to the most primary classification of data. This review paper gives a brief knowledge of these basic data types.

1. Numeric Data Types
   i) Integer type
   ii) Floating points type
   iii) Complex numbers

2. Text Data Types
   i) Characters
   ii) String

3. Boolean Data Types
   i) Boolean algebra
   ii) Logical operations
   iii) Conditional statements

4. Composite Data Types
   i) Arrays
   ii) Lists
   iii) Tuples

5. Special Data Types
   i) None
   ii) Enumerations
   iii) Custom data types

6. Type Casting and Type Coercion
   i) Implicit and explicit casting
ii) Type coercion in expressions

7. Type Safety and Compatibility
   i) Strongly typed and weakly typed languages
   ii) Type inference

8. Dynamic Typing and Static Typing
   i) Dynamic typing
   ii) Static typing

2. Previous Review
   1) Early Research: The theory of data types has been an elemental for initial programming languages like Fortran and cobol, Integers, floating point numbers, and characters were the form of classification of data. The introduction and defining of the data types and is representation in the systems was the only focus in the past researches.
   2) Type Systems: Static and dynamic were the two systems explored during the past researches. The enforcement of type checking at the moment of compilation in languages like C and Java is known as Static typing, whereas, the inspection of type at runtime in languages like Python and JavaScript is known as dynamic typing. The exchange between the approaches of these systems were being examined by the researchers on the basis of performance and safety.
   3) User-Defined Data Types: The user-defined data types were offered limited support in the initial programming languages. The invention of custom data types with the help of structures, objects and classes through succeeding researches. To advance in this field, Object-oriented programming languages such as C++ and Small-talk play a crucial role.
   4) Type Inference: In the evolution of type inference blueprint, compiler itself deducts the type of a variable being a remarkable research area. This excludes all the explicit type annotations by allowing expressive and compressed code. The significant examples that use type inference are Haskell and ML.
   5) Data Abstraction: Abstract data types (ADT’s) and encapsulation was the main focus in the data abstraction research. Besides, providing a well-defined interface for data manipulation, it hides the elementary implementation details. Barbara Liskov and Stephen Zilles were the two personalities who popularized the concept of adts, in their 1974 paper.
   6) Generic Types: The research on generic programming was escorted through the need for generic data type capable of working with various data while working with type safety. Templates and generics were introduced by languages like C++ and Java, respectively, to address above.
   7) Unicode and Character Encoding: Research centered on the representation and character coding, due to the globalization of software. Unicode became a standard, resulting in the exploration of efficient methods to handle international character sets in coding languages.
   8) Big Data and Data Structures: Efficient data structures and algorithms were being addressed in the researches to handle big data. Substantial study of topics such as balanced trees, hash tables and graph structures helped in optimization of data processing.
   9) Parallel and Concurrent Data Types: In the present world, there has been a center of attention on synchronization for concurrent and parallel programming and data types as compared to multi-core processors and distributed systems. This encompasses lock-free data structures and concurrency control.
   10) Type Safety and Security: Highlighting of type safety, type inference for security, as well as the prevention of programming with the increasing concern about software security, emphasized type safety, type inference for security, and the prevention of common programming helplessness such as buffering overflow and type-related security issues.

3. Working
   Data types are very important for the smooth functioning and representation of the data in programming languages. Data types organize data sets given by the user to the interpreter in different-different
values, these values are then further used and categorized in different variables, these values are then utilized according to different functions and operations being applied on them, each value acquire unique storage space based on their data types and information they hold. Data types works differently for different programming languages but they mainly work on the following principles in programming languages:

A. Type declaration
B. Data storage
C. Operations
D. Type compatibility
E. Type safety
F. User-defined data types
G. Data type conversions

4. Future Scope

Operations performed among several variables in programming languages, was made efficient and easy with the help of data types. Every data type has a specific and defined calculation as well as transformation, which are used with different variables of similar data type, to be executed without any difficulty. To reduce all the runtime errors, occurring during the operation of multiple variables, type system is used.

References

Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2021, pp. 305-311
