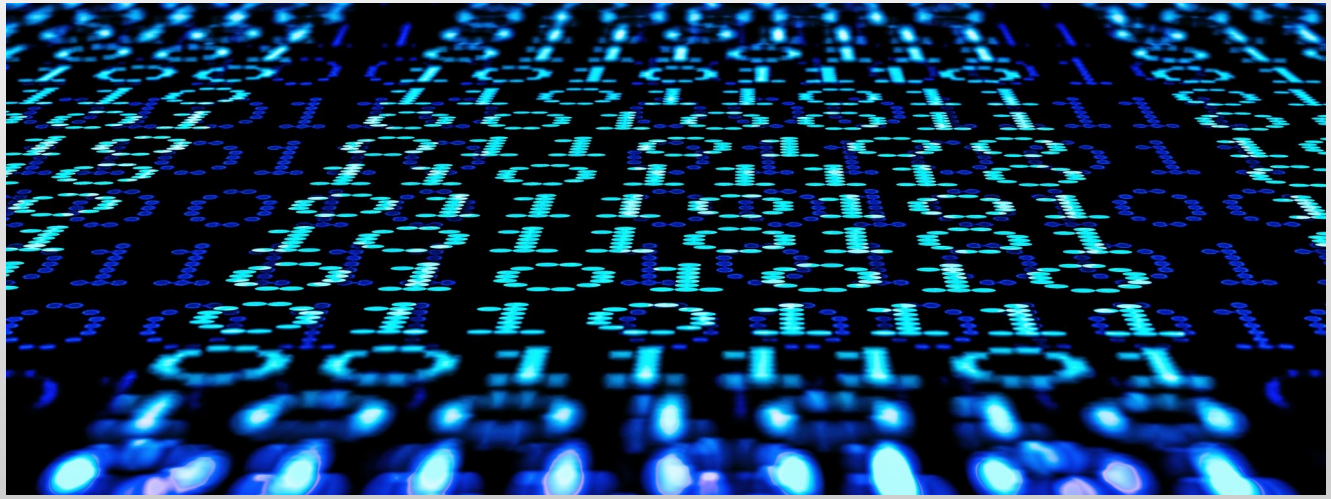
DATA TYPES



WHAT IS A DATA TYPE?

data type

- a classification that specifies which type of value a variable can hold or which type of data an expression can produce
- determines the **domain**, and the ***operations*** that can be performed on the data, and
- the memory representation of that data
- helps enforce constraints on what kind of values can be stored in variables or used in expressions, which contributes to the reliability and correctness of programs

domain

- the set of all possible, valid values that a particular data type can take
- important for understanding the constraints and limitations of a data type

PRIMITIVE DATA TYPES

primitive data type

- a basic data type provided by the programming language itself
- represents a fundamental kind of data
- used to define variables and represent simple values like numbers, characters, and Booleans
- fixed size in memory
- often used as building blocks for more complex data structures

NUMERICAL DATA TYPES

Name	Range	Storage Size
`byte`	-2^7 to $2^7 - 1$ (-128 to 127)	8-bit signed
`short`	-2^{15} to $2^{15} - 1$ (-32768 to 32767)	16-bit signed
`int`	-2^{31} to $2^{31} - 1$ (-2147483648 to 2147483647)	32-bit signed
`long`	-2^{63} to $2^{63} - 1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed
`float`	Negative range: -3.4028235E+38 to -1.4E-45 Positive range: 1.4E-45 to 3.4028235E+38	32-bit IEEE 754
`double`	Negative range: -1.7976931348623157E+308 to -4.9E-324 Positive range: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

NUMERIC OPERATORS

Name	Meaning	Example	Result
+	Addition	34 + 1	35
-	Subtraction	34.0 - 0.1	33.9
*	Multiplication	300 * 30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

THE BOOLEAN TYPE AND OPERATORS

- Domain: `true` or `false`
- Six comparison operators (also known as relational operators) that can be used to compare two values
- The result of the comparison is a Boolean value: true or false.

RELATIONAL OPERATORS

Java Operator	Mathematics Symbol	Name	Example (radius is 5)	Result
`<`	<	less than	`radius < 0`	`false`
`<=`	≤	less than or equal to	`radius <= 0`	`false`
`>`	>	greater than	`radius > 0`	`true`
`>=`	≥	greater than or equal to	`radius >= 0`	`true`
`==`	=	equal to	`radius == 0`	`false`
`!=`	≠	not equal to	`radius != 0`	`true`

CHARACTER DATA TYPE

```
char letter = 'A'; (ASCII)
char numChar = '4'; (ASCII)
char letter = '\u0041'; (Unicode)
char numChar = '\u0034'; (Unicode)
```

Four hexadecimal digits.

- American Standard Code for Information Interchange (ASCII)
- Unicode
- increment and decrement operators can be used

THE FIRST PROGRAMMER

ADA LOVELACE

ADA LOVELACE, THE "FIRST PROGRAMMER"

- Augusta Ada King, Countess of Lovelace, commonly known as Ada Lovelace (1815 – 1852)
- an English mathematician and writer
- Worked on Charles Babbage's early mechanical general-purpose computer, the Analytical Engine
- Translated the Italian mathematician Luigi Menabrea's article on the Analytical Engine
- Added extensive annotations of her own explaining how the machine could be programmed to perform various operations beyond mere number crunching
- Described an algorithm for the Analytical Engine to calculate Bernoulli numbers, which is often considered the first computer program

ADA LOVELACE: ALGORITHM AND PROGRAMMING CONCEPTS

- Groundbreaking insight into the concept of "looping" or "iteration" in programming allowing for the creation of more complex calculations and operations
- Dismissed the idea of artificial intelligence: "*The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform. It can follow analysis; but it has no power of anticipating any analytical relations or truths.*"
- Insight into potential of computing devices: *[The Analytical Engine] might act upon other things besides number, were* ***objects*** *found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine…Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.*

OBJECT-ORIENTED PROGRAMMING (OOP)

OBJECT-ORIENTED PROGRAMMING (OOP)

- A programming paradigm that focuses on designing software using "objects"
- Simula (1960s)
  - The first programming language to introduce explicit support for object-oriented concepts
  - Included classes, objects, inheritance, and other OOP features
- Smalltalk (1970s - 1980s)
  - Played a significant role in popularizing object-oriented programming
  - Introduced many key concepts of OOP, such as classes, objects, and inheritance
  - Also emphasized the importance of a visual user interface through its graphical environment

OBJECT-ORIENTED PROGRAMMING (OOP)

- C++ Emergence (1980s)
  - an extension of the C programming language
  - Introduced object-oriented features while retaining compatibility with existing C code
  - Gained popularity due to its versatility, allowing both low-level programming and high-level OOP constructs
- Java Arrival (1990s)
  - "Write Once, Run Anywhere" capability
  - Emphasis on platform independence and security
  - Embraced OOP principles as a core part of its design

OOP IN SOFTWARE DEVELOPMENT

- Has become the predominant paradigm in software development
- Encouraging modularization, code reusability, and easier maintenance
- Many software development methodologies, like Agile and Scrum, have integrated OOP concepts into their practices
- Evolved beyond its initial concepts to incorporate ideas from various fields,
  - software engineering
  - design patterns, and
  - architecture
- Shaped the way software is designed, written, and maintained, leading to more scalable and maintainable applications

COMPLEX DATA TYPES

COMPLEX DATA TYPE (REFERENCE DATA TYPE)

- Unlike primitive data types, store references to memory locations where the actual data is stored
- More flexible and versatile than primitives, allowing you to represent more complex structures and data relationships
- Built on top of primitive data types
- Include attributes and methods that operate on the data they encapsulate
- Also referred to as reference types because they store references (memory addresses) to the actual data rather than the data itself

REFERENCES

Liang, Y. D. (2022). *Introduction to java programming and Data Structures: Comprehensive Version*. Pearson Education Limited.

Simonite, T. (2009, March 24). Celebrating Ada Lovelace: The "world's first programmer" - short sharp science - new scientist. https://web.archive.org/web/20090327073325/https://www.newscientist.com/blogs/shortsharpscience/2009/03/ada-lovelace-day.html

Menabrea, L., & King, A. A. (1842, October). *Sketch of invented by Charles Babbage*. Sketch of The Analytical Engine. https://www.fourmilab.ch/babbage/sketch.html