History of programming

Functional programming

Imperative programming

Functional programming

- In computer science, **functional programming** is a programming paradigm where programs are constructed by applying and composing functions. It is a declarative programming paradigm in which function definitions [..] map values to other values, rather than a sequence of imperative statements which update the running state of the program.
- Functional programming is characterized by immutability, declarative code, and the use of pure functions, which avoid side effects and mutable data.
- Functional programming has its roots in academia

https://en.wikipedia.org/wiki/Functional_programming

Imperative programming

- In computer science, **imperative programming** is a programming paradigm that uses statements that change a program's state. [...] an imperative program consists of commands for the computer to perform.
- **Procedural programming** is a type of imperative programming in which the program is built from one or more procedures (also termed subroutines or functions). The terms are often used as synonyms, but the use of procedures has a dramatic effect on how imperative programs appear and how they are constructed.
- Heavy procedural programming, in which states changes are localized to procedures or restricted to explicit arguments and returns from procedures, is a form of structured programs. Since the 1960s, structured programming and modular programming in general have been promoted as techniques to improve the maintenability and overall quality of imperative programs.

https://en.wikipedia.org/wiki/Imperative_programming

1957 Born of necessity

- In the early 1950s, computer programming was the exclusive domain of a small group of specialists who wrote code in machine language, a complex and cumbersome set of instructions. Programming was for experts only — outsiders need not apply. Then came **Fortran**.
- **Fortran** instigated the process of abstracting software from the hardware on which it ran. Previous machine language programs had to be written for a specific computer, while a Fortran program could run on any system with a Fortran compiler.

https://www.ibm.com/history/fortran

First Programming Languages

- **Fortran is a procedural programming language**
- The **C programming language**, developed in 1972 at Bell Labs, primarily as a system programming language to write the Unix operating system.
- The C language was further refined and standardized over the years.
- The C language is an **imperative, procedural language**
- C has influenced many other modern programming languages, such as C++, Java and Python.

Procedural
vs
Object-Oriented
Programming

Object-Oriented Programming

- **Simula** was the first programming language that introduced essential object-oriented idea, such as classes and inheritance.
- Inspired by Simula, Alan Kay developed **Smalltalk** which found use in business.
- Smalltalk influenced many others programming languages, such as C++.

> I thought of objects being like biological cells and/or individual computers on a network, only able to communicate with messages (so messaging came at the very beginning – it took a while to see how to do messaging in a programming language efficiently enough to be useful).
>
> Alan Kay, [1]

https://en.wikipedia.org/wiki/Object-oriented_programming

Object-Oriented Programming

> I thought of objects being like biological cells and/or individual computers on a network, only able to communicate with messages (so messaging came at the very beginning – it took a while to see how to do messaging in a programming language efficiently enough to be useful).
>
> Alan Kay, [1]

- **Simula** was the first programming language that introduced essential object-oriented idea, such as classes and inheritance.
- Inspired by Simula, Alan Kay developed **Smalltalk** which found use in business.
- Smalltalk influenced many others programming languages, such as C++.

https://en.wikipedia.org/wiki/Object-oriented_programming

Object-Oriented Programming

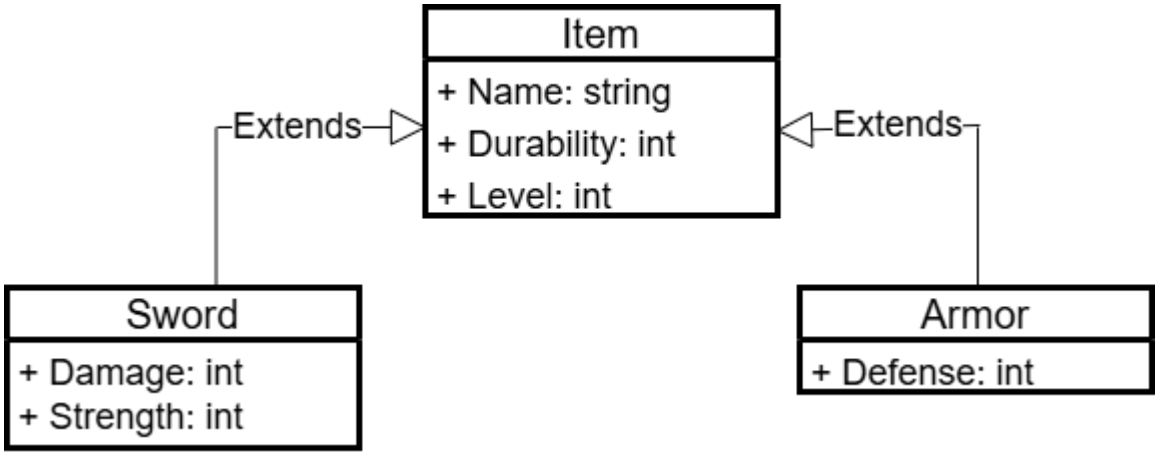
- Many of the most widely used programming languages support object-oriented programming to a greater or lesser degree, typically as part of multiple programming in combination with others such as imperative programming and declarative programming.
- In software development, an **object** is an entity that has a state and represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain. An object encapsulates actions to perform on its state.

Object-Oriented Programming

- An object is a type of data structure that has two main parts: fields and methods. Fields may also be known as members, attributes, or properties, and hold information in the form of state variables. Methods are actions, subroutines, or procedures, defining the object's behavior in code.

https://en.wikipedia.org/wiki/Object-oriented_programming

An example..


Item
+ Name: string
+ Durability: int
+ Level: int
Extends
Extends
Sword
+ Damage: int
+ Strength: int
Armor
+ Defense: int