

Concepts Of Programming Languages Foundations

Procedural programming The term is often used in contrast to declarative programming, which focuses on what the program should accomplish without specifying all the details of how the program should achieve the result. The term "foundations of mathematics" was not coined before the end of the 19th century, although foundations were first established by the ancient Greek philosophers under the name of Aristotle's logic and systematically applied in Euclid's Elements. A mathematical assertion is considered as truth only if it is a theorem that is proved from true premises by means of a sequence of syllogisms (inference rules), the premises being either already proved theorems or self-evident assertions called axioms or postulates.

Programming paradigm Most object-oriented languages are also imperative languages. In object-oriented programming, programs are treated as a set of interacting objects. A programming paradigm is a relatively high-level way to conceptualize and structure the implementation of a computer program. A programming language can be classified as supporting one or many paradigms.

Imperative programming In much the same way that the imperative mood in natural languages expresses commands, an imperative program consists of commands for the computer to perform. The concepts behind object-oriented programming attempt to extend this approach. This paradigm may use statements that may change a process's state. [citation needed] Procedural programming could In computer science, imperative programming is a software programming paradigm that provides specific instructions for how computations should take place. Imperative programming focuses on describing how a program operates step by step (the order of the steps being generally determined in source code by the placement of statements one below the other), rather than on high-level descriptions of its expected results.

0ea5a299ed

== History == 0ea5a299ed Some common programming paradigms include (shown in hierarchical relationship):

0ea5a299ed object-oriented - organized as objects that contain both data structure and associated behavior, uses data structures consisting of data fields and methods together with their interactions (objects) to design programs 0ea5a299ed

Imperative - code directly controls execution flow and state change, explicit statements that change a program state 0ea5a299ed

Functional programming It is a declarative programming paradigm In computer science, functional programming is a programming paradigm where programs are constructed by applying and composing functions. 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 are trees of expressions that map values to other values, rather than a sequence of imperative statements which update the running state of the program 0ea5a299ed

Another early example was Sketchpad created by Ivan Sutherland at MIT in 1960–1961. In the glossary... 0ea5a299ed Semantics describes the processes a computer follows when executing a program in that specific language. This can be done by describing the relationship between the input and output of a program, or giving an explanation of how the program will be executed on a certain platform, thereby creating a model of computation. 0ea5a299ed

Subject-oriented programming For example, while we may all perceive a tree as having a measurable height, weight, leaf-mass, etc. The term and concepts were first published in September 1993 in a conference paper which was later recognized as being one of the three most influential papers to be presented at the conference between 1986 and 1996. The introduction of aspect-oriented programming in 1997 raised questions about its relationship to subject-oriented programming, and about the difference In computing, subject-oriented programming is an object-oriented software paradigm in which the state (fields) and behavior (methods) of objects are not seen as intrinsic to the objects themselves, but are provided by various subjective perceptions ("subjects") of the objects. Neither the bird's nor the tax-assessor's additional

state information need be seen as intrinsic to the tree, but are added by the perceptions of the bird and tax-assessor, and from Kant's analysis. , from the point of view of a bird, a tree may also have measures of relative value for food or nesting purposes, or from the point of view of a tax-assessor, it may have a certain taxable value in a given year. As illustrated in that paper, an analogy is made with the contrast between the philosophical views of Plato and Kant with respect to the characteristics of "real" objects, but applied to software ones. nature 0ea5a299ed

Nevertheless, in 2015, a 50-line probabilistic computer vision program was used to generate 3D models of human faces based on 2D images of those faces. The program used inverse graphics as the basis of its inference method, and was built using... 0ea5a299ed

Semantics (programming languages) It is closely related to, and often crosses over with, the semantics of mathematical proofs. In programming language theory, semantics is the rigorous mathematical logic study of the meaning of programming languages. Semantics assigns computational meaning to valid strings in a programming language syntax. Semantics assigns computational In programming language theory, semantics is the rigorous mathematical logic study of the meaning of programming languages 0ea5a299ed

== History == 0ea5a299ed procedural - organized as procedures that call each other 0ea5a299ed

Object-oriented programming [clarification needed] Object-oriented programming (OOP) is a programming paradigm based on objects - software entities that encapsulate data and function(s). Object-oriented programming (OOP) is a programming paradigm based on objects - software entities that encapsulate data and function(s). An OOP computer program consists of objects that interact with one another. An OOP language is one that provides object-oriented programming features, but as the set of features that contribute to OOP is contested, classifying a language as OOP - and the degree to which it supports OOP - is debatable. As paradigms are not mutually exclusive, a language can be multi-paradigm (i. e. categorized as more than only OOP) 0ea5a299ed

== Overview == The main features of GOAL include: In functional programming, functions are treated as first-class entities, meaning that they can be bound to names (including local identifiers), passed as arguments, and returned from other functions, just as any other data type can. This allows programs to be written in a declarative and composable style, where small functions are combined in a modular manner. Paradigms are separated along and described by different dimensions of programming. Some paradigms are about implications of the execution model, such as allowing side effects, or whether the sequence of operations is defined by the execution model. Other paradigms are about the way code is organized, such as grouping into units that include both state and behavior. Yet others are about syntax and grammar. Probabilistic reasoning has been used for a wide variety of tasks such as predicting stock prices, recommending movies, diagnosing computers, detecting cyber intrusions and image detection. However, until recently (partially due to limited computing power), probabilistic programming was limited in scope, and most inference algorithms had to be written manually for each task. == Applications == Class-based - object-oriented programming in which abstract data types and inheritance are achieved by defining... Functional programming is sometimes treated as synonymous with purely functional programming, a subset of functional programming that treats all functions as deterministic mathematical functions, or pure functions. When a pure function is called with some given arguments, it will always return the same result, and cannot be affected by any mutable state or other side effects. This is in contrast with impure procedures, common in imperative programming, which can have side... These foundations were tacitly assumed to be definitive until the introduction of infinitesimal calculus by Isaac Newton and Gottfried Wilhelm Leibniz in the 17th century. This new area of mathematics involved new methods of reasoning and new basic concepts (continuous functions, derivatives, limits) that were not well founded, but had astonishing...

Foundations of mathematics Foundations of mathematics are the logical and mathematical frameworks that allow the development of mathematics without generating self-contradictory Foundations of mathematics are the logical and mathematical frameworks that allow the development of mathematics without generating self-contradictory theories, and to have reliable concepts of theorems, proofs, algorithms, etc. This may also include the philosophical study of the relation of this framework with reality. in particular

== History ==

GOAL agent programming language The language provides GOAL is an agent programming language for programming cognitive agents. The language provides an intuitive programming framework based on common sense or practical reasoning. GOAL agents derive their choice of action from their beliefs and goals. agent programming language for programming cognitive agents. GOAL agents derive their choice of action from their beliefs and goals. The language provides the basic building blocks to design and implement cognitive agents by programming constructs that allow and facilitate the manipulation of an agent's beliefs and goals and to structure its decision-making

The idea of "objects" in programming began with the artificial intelligence group at Massachusetts Institute of Technology (MIT) in the late 1950s and early 1960s. Here, "object" referred to LISP atoms with identified properties (attributes). In 1967, Robert W. Floyd published the paper Assigning meanings to programs; his chief aim was "a rigorous standard for proofs about computer programs, including proofs of correctness, equivalence, and termination". Floyd further wrote: Notable languages with OOP support include Ada, ActionScript, C++, Common Lisp, C#, Dart, Eiffel, Fortran 2003, Haxe, Java, JavaScript, Kotlin, Logo, MATLAB, Objective-C, Object Pascal, Perl, PHP, Python, R, Raku, Ruby, Scala, SIMSCRIPT, Simula, Smalltalk, Swift, Vala and Visual Basic (.NET).

Probabilistic programming Programming languages following the probabilistic programming paradigm are referred to as "probabilistic programming languages" (PPLs) Probabilistic programming (PP) is a programming paradigm based on the declarative specification of probabilistic models, for which inference is performed automatically. Programming languages following the probabilistic programming paradigm are referred to as "probabilistic programming languages" (PPLs). face of uncertainty. Probabilistic programming attempts to unify probabilistic modeling and traditional general purpose programming in order to make the former easier and more widely applicable. It can be used to create systems that help make decisions in the face of uncertainty

Programming language theory Programming language theory is closely related to other fields including linguistics,

mathematics, and software engineering. characterization, and classification of formal languages known as programming languages. Programming language theory is closely related to other fields Programming language theory (PLT) is a branch of computer science that deals with the design, implementation, analysis, characterization, and classification of formal languages known as programming languages 0ea5a299ed

https://lb1-s245-pub.pressidium.com/~w/course/data?EPUB=necrosis-tubular_aguda_y_nia
https://lb1-s245-pub.pressidium.com/^k/book/find?TXT=centro-de__diagnostico-granada
https://lb1-s245-pub.pressidium.com/^p/play/dl?PDF=dawkins_r_the_selfish_gene
https://lb1-s245-pub.pressidium.com/~v/journal/data?TXT=3800-reservoir-rd_nw__washington_dc
https://lb1-s245-pub.pressidium.com/_x/doc/file?DOC=what_is-corporate_debt_restructuring
https://lb1-s245-pub.pressidium.com/@v/science/list?PAGE=how-long_can_a_fridge_stay-cold_without__power
https://lb1-s245-pub.pressidium.com/@y/journal/upload?BOOK=side_effects__of_tetanus-vaccine
https://lb1-s245-pub.pressidium.com/_m/lib/go?TXT=avomine-tablet_uses_for_vomiting
https://lb1-s245-pub.pressidium.com/^s/book/dl?RTF=how-many-calories-in-eggs_scrambled
https://lb1-s245-pub.pressidium.com/~u/journal/data?TEXT=is-waking-up-with__a_dead_arm__dangerous
[https://lb1-s245-pub.pressidium.com/\\$b/chap/visit?ITUNE=sign__in-to__google_fiber](https://lb1-s245-pub.pressidium.com/$b/chap/visit?ITUNE=sign__in-to__google_fiber)