Programming

Computer programs are developed by computer programmers or software engineers. Computer programming encompasses a broad set of activities that include planning, coding, testing, and documenting. Most programmers participate in all of these phases of program development, but focus on the coding process.

Methodologies can be classified as predictive or agile. A predictive methodology requires extensive planning and documentation up front. Predictive methodologies are preferred for large software development projects. an agile methodology focuses on flexible development and specifications that evolve as a project progresses.

Programming paradigm refers to a way of conceptualizing and structuring the tasks a computer performs. Today’s most popular programming paradigms are: Event-driven, Procedural, Object-oriented, Declarative.

Programmers can use a text editor, program editor, or VDE to code computer programs. A text editor is any word processor that can be used for basic text editing tasks, such as writing e-mail, creating documents, or coding computer programs. A program editor is a type of text editor specially designed for entering code for computer programs. A VDE (visual development environment) provides programmers with tools to build substantial sections of a program by pointing and clicking rather than typing lines of code.

When a program doesn’t work correctly, it is usually the result of a syntax, logic, or runtime error. A syntax error occurs when an instruction does not follow the syntax rules, or grammar, of the programming language. A runtime error occurs when a program runs. A logic error is a type of runtime error in the logic or design of a program, such as using the wrong formula. Programmers can locate errors in a program by reading through lines of code or they can use a tool called a debugger.

Thousands of different programming languages have been created and more being created every year. The mainstream consists of languages that are procedural in nature. Examples of procedural languages are Algol, BASIC , C, COBOL , FORTRAN , and Pascal.

A variant of procedural languages is the object-oriented language. for example, C++, Java, and Smalltalk. To understand when and why using object-oriented languages let’s compare it with a common alternative. In procedural code, the program is written as a long series of operations to execute. So, instead of writing a single large program, my object-oriented code is split apart into several self-contained objects. Almost like several mini programs where each object contains its own data and logic to describe how it behaves and interacts with other objects. One of the main advantages of using an object-oriented approach is code re-usability.

Although the bulk of today’s software is written using procedural languages, there are some important languages constructed using quite different paradigms. LISP, for example, is a powerful language used in artificial intelligence applications.

You might encounter logic programming languages, like Prolog. Or functional programming languages, like Haskell. However, those both tend to live in very specialized environments.

There are also “logic programming” languages, of which Prolog is best known. Here a chain of logical steps is constructed such that the program can traverse it to find the solution of a problem.

A variety of languages have also been designed for tasks such as data management, interfacing Web pages, and system administration (scripting languages, awk, Perl, PHP, and Python).

Artificial intelligence (AI) is the ability of a machine or a computer program to to think and learn. The concept of AI is based on the idea of building machines capable of thinking, acting, and learning like humans. AI is not the same as machine learning. The entire machine learning concept is based on the assumption that we should give machines access to information and let them learn from it themselves. Artificial intelligence, in its turn, is a bunch of technologies that include machine learning and some other technologies like natural language processing, inference algorithms, neural networks, etc. Examples of artificial intelligence in use today include smart home devices like Google’s NEST, self-driving cars like those produced by Tesla, gadgets able to recognize, Spotify, Pandora, and Apple Music, and online games like Alien: Isolation.