Glossary of computer programming

This is a glossary of computer programming.

A

abstract class

A class that cannot be directly constructed, one that can be constructed only through construction of some of its subclasses.

abstract type

A type in a nominative type system that cannot be instantiated.

actual argument

A value, or reference to a value, passed to a function.

app

An application that executes on a small, handheld device.

application

A program or integrated suite of programs that has a defined function.

argument

A value, or reference to a value, passed to a function; an actual argument.

argument

A parameter in a function definition; a formal argument.

array

An ordered sequence of same-typed values whose elements are fast to access by their numerical index in the array.

B

Boolean

A data type for yes or no, true or false values.

C

class

A set of objects having the same behavior (but typically differing in state), or a template defining such a set.

compiler

A computer program which transforms source code into object code.

constant

An identifier that is bound to an invariant value.

constructor

A class method (in object-oriented programming) that creates and initializes each instance of an object.

D

data type

A classification or category of various types of data, that states the possible values that can be taken, how they are stored, and what range of operations are allowed on them.

destructor

In object-oriented programming, the command sequence that is launched when the execution of an object is finished.

dump

A formatted listing of the contents of program storage, especially when produced automatically by a failing program

E

enumeration

A data type whose values are a set of mutually exclusive named constants.

exception

An interruption in normal processing, especially as caused by an error condition.

F

floating point

A method of representing real numbers as a pair of integers (the mantissa and characteristic)

flow chart

A schematic representation of the logic that defines the flow of control through a program

formal argument

A parameter in a function definition.

function

A routine that receives zero or more arguments and may return a result.

functional programming

A programming paradigm that treats computation as the evaluation of mathematical functions, avoids state and mutable data, and makes it easy to construct functions as if they were data objects.

G

goto / **go to**

A statement (in source code) that transfers control unconditionally to another part of a program

H

heap

An area of memory reserved for dynamically allocated data objects, contrasted to the stack.

I

identifier

A formal name used in source code to refer to a variable, function, procedure, package, etc.

integer

A data type for integer values.

interpreter

A program which executes another program written in a programming language other than machine code.

L

linker / link editor

A computer program that takes one or more objects generated by compilers and assembles them into a single executable program.

logic programming

A style or paradigm of computer programming exemplified by the language Prolog.

M

machine code

System of instructions and data directly understandable by a computer's central processing unit.

method

In object-oriented languages, a subroutine or function belonging to a class or object.

module

A program that is linked with others to form a functioning application; one method of implementing a subroutine

O

object

An instance of a class.

object code

The output of a compiler or assembler, not necessarily executable directly without linking to other modules.

object-oriented

Using entities called objects that can process data and exchange messages with other objects.

P

paradigm

A fundamental style of computer programming to which the design of a programming language typically has to cater, such as imperative programming, declarative programming, or, on a finer level, functional programming, logic programming or object-oriented programming.

parameter

A name in a function or subroutine definition that is replaced by, or bound to, the corresponding actual argument when the function or subroutine is called.

procedure

A subroutine or function coded to perform a specific task.

program

A software application, or a collection of software applications, designed to perform a specific task.

R

real

a number containing a decimal point, e.g. the number pi is a real number with a value of approximately 3.14159268

run time

The time during which a program is executing, as oppose to the compile time.

S

source code

Human-readable instructions in a programming language, to be transformed into machine instructions by a compiler, interpreter, assembler or other such system.

stack

The portion of the computer memory used to keep track of called procedures or call instructions.

string

A data type for a sequence of characters such as letters of English alphabet.

subclass

In object-oriented programming, an object class derived from another class (its superclass) from which it inherits a base set of properties and methods.

subprogram

A program contained within a larger program.

subroutine

A section of code that implements a task. While it may be used at more than one point in a program, it need not be.

superclass

A class that passes attributes and methods down the hierarchy to subclasses.

T

type

A tag attached to variables and values used in determining what values may be assigned to what variables.

V

variable

A named memory location in which a program can store intermediate results and from which it can read them.