Early Ideas

Concept Summary

object Java objects model objects from a problem domain.

class Objects are created from classes. The class describes the kind of object; the objects represent individual instantiations of the class.

method We can communicate with objects by invoking methods on them. Objects usually do something if we invoke a method.

parameter Methods can have parameters to provide additional information for a task.

signature The header of a method is called its signature. It provides information needed to invoke that method.

type Parameters have types. The type defines what kinds of values a parameter can take.

multiple instances Many similar objects can be created from a single class.

state Objects have state. The state is represented by storing values in fields.

method-calling Objects can communicate by calling each other's methods.

source code The source code of a class determines the structure and the behavior (the fields and methods) of each of the objects in that class.

result Methods may return information about an object via a return value.

field Fields store data for an object to use. Fields are also known as instance variables.

comment Comments are inserted into the source code of a class to provide explanations to human readers. They have no effect on the functionality of the class or code.

constructor Constructors allow each object to be set up properly when it is first created.

scope The scope of a variable defines the section of source code from where the variable can be accessed.

lifetime The lifetime of a variable describes how long the variable continues to exist before it is destroyed.

assignment Assignment statements store the value represented by the right-hand side of the statement in the variable named on the left.

method Methods consist of two parts: a header and a body.

accessor method Accessor methods return information about the state of an object.

mutator method Mutator methods change the state of an object.

println The method “System.out.println” prints its parameter to the text terminal.

conditional A conditional statement takes one of two possible actions based upon the result of a test.

boolean expression Boolean expressions have only two possible values: true and false. They are commonly found controlling the choice between the two paths through a conditional statement.

local variable A local variable is a variable declared and used within a single method. Its scope and lifetime are limited to that of the method.

Terms

problem solving: The process of formulating a problem, finding a solution, and expressing the solution.

program: A sequence of instructions that specifies how to perform tasks on a computer.

programming: The application of problem solving to creating executable computer programs.

computer science: The scientific and practical approach to computation and its applications.

algorithm: A procedure or formula for solving a problem, with or without a computer.

bug: An error in a program.

debugging: The process of finding and removing errors.

high-level language: A programming language that is designed to be easy for humans to read and write like Java.

low-level language: A programming language that is designed to be easy for a computer to run. Also called “machine language” or “assembly language”.

portable: The ability of a program to run on more than one kind of computer.

interpret: To run a program in a high-level language by translating it one line at a time and immediately executing the corresponding instructions.

compile: To translate a program in a high-level language into a low-level language, all at once, in preparation for later execution.

source code: A program in a high-level language, before being compiled.

object code: The output of the compiler, after translating the program.

executable: Another name for object code that is ready to run on specific hardware.

byte code: A special kind of object code used for Java programs. Byte code is similar to a low-level language, but it is portable like a high-level language.

statement: Part of a program that specifies one step of an algorithm.

print statement: A statement that causes output to be displayed on the screen.

method: A named sequence of statements.

class: For now, a collection of related methods. (We will see later that there is more to it.)

comment: A part of a program that contains information about the program but has no effect when the program runs.

string: A sequence of characters; the primary data type for text.

variable: A named storage location for values. All variables have a type, which is declared when the variable is created.
value: A number, string, or other data that can be stored in a variable. Every value belongs to a type (for example, int or String).

declaration: A statement that creates a new variable and specifies its type.

type: Mathematically speaking, a set of values. The type of a variable determines which values it can have. In programming, it means a the “kind” like String or Integer.

syntax: The structure of a program; the arrangement of the words and symbols it contains.

keyword: A reserved word used by the compiler to analyze programs. You cannot use keywords (like public, class, and void) as variable names.

assignment: A statement that gives a value to a variable.
initialize: To assign a variable for the first time.

state diagram: A graphical representation of the state of a program at a point in time.

operator: A symbol that represents a computation like addition, multiplication, or string concatenation.

operand: One of the values on which an operator operates. Most operators in Java require two operands.

expression: A combination of variables, operators, and values that represents a single value. Expressions also have types, as determined by their operators and operands.

floating-point: A data type that represents numbers with an integer part and a fractional part. In Java, the default floating-point type is double.

rounding error: The difference between the number we want to represent and the nearest floating-point number.

concatenate: To join two values, often strings, end-to-end. order of operations: The rules that determine in what order operations are evaluated.

composition: The ability to combine simple expressions and statements into compound expressions and statements.

compile-time error: An error in the source code that makes it impossible to compile. Also called a “syntax error”.

parse: To analyze the structure of a program; what the compiler does first.

run-time error: An error in a program that makes it impossible to run to completion. Also called an “exception”.

logic error: An error in a program that makes it do something other than what the programmer intended.