

The multiple-choice section tests whether you can say precisely what a construct does, and the free-response section is graded against a rubric that names constructs by their proper terms. Learn these sentences; then, when tracing code, say the term aloud as you meet it.

Objects and classes

Class – a blueprint that defines the state (instance variables) and behaviour (methods) shared by objects of that type

Object – an instance of a class, created with `new`, holding its own copy of the instance variables

Constructor – a special method with the class's name and no return type, run once when an object is created, to initialise its state

Instance variable – a variable declared in a class but outside any method, one copy per object

Static variable – a variable declared static, shared by every object of the class rather than copied per object

Encapsulation – keeping instance variables private and giving access through public methods, so an object controls its own state

Accessor and mutator – an accessor returns a value without changing state; a mutator changes state and typically returns void

Inheritance and polymorphism

Inheritance – a subclass acquiring the accessible fields and methods of its superclass, declared with `extends`

Method overriding – a subclass providing its own implementation of a method with the same signature as one in the superclass

Method overloading – two methods in the same class sharing a name but differing in parameter list

Polymorphism – the ability of a superclass reference to hold a subclass object, so the overridden method actually run is chosen at RUN TIME by the object's type

Abstract class – a class that cannot be instantiated and may declare abstract methods that its subclasses must implement

super super method – a reference used to call the superclass's constructor or an overridden superclass method

Data, control and arrays

- Primitive type** – a type such as int, double or boolean holding a value directly, not a reference
- Reference type** – a type whose variable holds the address of an object; two references can point to the same object
- Integer division** – division of two ints, which discards the fractional part rather than rounding
– $7/2$ is 3
- Short-circuit evaluation** – `&&` and `||` stop evaluating as soon as the result is determined, so the right operand may never run
- Array** – a fixed-length, zero-indexed sequence of elements of one type, whose length is a field, not a method
- ArrayList** – a resizable list of object references whose size is a METHOD, and which stores objects rather than primitives
- Enhanced for loop** `for` – a loop that visits each element in turn; assigning to its variable does not change the underlying array

Correctness and analysis

- Off-by-one error** – a loop bound that runs one iteration too few or too many; with zero-indexing, valid indices run to length $- 1$
- NullPointerException** – a run-time error from calling a method on a reference that is null
- ArrayIndexOutOfBoundsException** – a run-time error from using an index below 0 or at or above the array's length
- Recursion** – a method calling itself on a smaller input, requiring a base case that returns without recursing
- Linear search** – checking elements in order until the target is found or the list ends; it does not require sorted data
- Binary search** – repeatedly halving a SORTED range by comparing with its middle element