**Class:** A programmer-defined type. A class definition creates a new class object.

**Class object:** An object that contains information about a programmer-defined type. The class object can be used to create instances of the type.

**Instance:** An object that belongs to a class.

**Instantiate:** To create a new object.

**Attribute:** One of the named values associated with an object.

**Embedded object:** An object that is stored as an attribute of another object.

**Shallow copy**: To copy the contents of an object, including any references to embedded objects; implemented by the copy function in the copy module.

**Deep copy:** To copy the contents of an object as well as any embedded objects, and any objects embedded in them, and so on; implemented by the deepcopy function in the copy module.

**Object diagram:** A diagram that shows objects, their attributes, and the values of the attribute

**Prototype and patch:** A development plan that involves writing a rough draft of a program, testing, and correcting errors as they are found.

**Designed development:** A development plan that involves high-level insight into the problem and more planning than incremental development or prototype development.

**Pure function:** A function that does not modify any of the objects it receives as arguments. Most pure functions are fruitful.

**Modifier:** A function that changes one or more of the objects it receives as arguments. Most modifiers are void; that is, they return none.

**Functional programming style:** A style of program design in which the majority of functions are pure. Invariant: A condition that should always be true during the execution of a program.

**Object-oriented language:** A language that provides features, such as programmerdefined types and methods that facilitate object-oriented programming.

**Object-oriented programming:** A style of programming in which data and the operations that manipulate it are organized into classes and methods.

**Method:** A function that is defined inside a class definition and is invoked on instances of that class.

**Subject:** The object a method is invoked on.

**Positional argument:** An argument that does not include a parameter name, so it is not a keyword argument.

**Operator overloading**: Changing the behavior of an operator like + so it works with a programmer-defined type.

**Type-based dispatch:** A programming pattern that checks the type of an operand and invokes different functions for different types. polymorphic: Pertaining to a function that can work with more than one type