**Glossary**

**attribute**

Some state or value that belongs to a particular object. For example, tess has a color.

**canvas**

A surface within a window where drawing takes place.

**control flow**

See *flow of execution* in the next chapter.

**for loop**

A statement in Python for convenient repetition of statements in the *body* of the loop.

**instance**

An object that belongs to a class. *tess* and *alex* are different instances of the class *Turtle*

**invoke**

An object has methods. We use the verb invoke to mean *activate the method*. Invoking a method is done by putting parentheses after the method name, with some possible arguments. So wn.exitonclick() is an invocation of the exitonclick method.

**iteration**

A basic building block for algorithms (programs). It allows steps to be repeated. Sometimes called *looping*.

**loop body**

Any number of statements nested inside a loop. The nesting is indicated by the fact that the statements are indented under the for loop statement.

**loop variable**

A variable used as part of a for loop. It is assigned a different value on each iteration of the loop, and is used as part of the terminating condition of the loop,

**method**

A function that is attached to an object. Invoking or activating the method causes the object to respond in some way, e.g. forward is the method when we say tess.forward(100).

**module**

A file containing Python definitions and statements intended for use in other Python programs. The contents of a module are made available to the other program by using the *import* statement.

**object**

A “thing” to which a variable can refer. This could be a screen window, or one of the turtles you have created.

**range**

A built-in function in Python for generating sequences of integers. It is especially useful when we need to write a for loop that executes a fixed number of times.

**sequential**

The default behavior of a program. Step by step processing of algorithm.

**state**

The collection of attribute values that a specific data object maintains.

**terminating condition**

A condition that occurs which causes a loop to stop repeating its body. In the for loops we saw in this chapter, the terminating condition has been when there are no more elements to assign to the loop variable.

**turtle**

A data object used to create pictures (known as turtle graphics).

**deterministic**

A process that is repeatable and predictable.

**documentation**

A place where you can go to get detailed information about aspects of your programming language.

**module**

A file containing Python definitions and statements intended for use in other Python programs. The contents of a module are made available to the other program by using the *import* statement.

**pseudo-random number**

A number that is not genuinely random but is instead created algorithmically.

**random number**

A number that is generated in such a way as to exhibit statistical randomness.

**random number generator**

A function that will provide you with random numbers, usually between 0 and 1.

**standard library**

A collection of modules that are part of the normal installation of Python.

**Glossary**

attribute

One of the named data items that makes up an instance.

class

A user-defined compound type. A class can also be thought of as a template for the objects that are instances of it.

constructor

Every class has a “factory”, called by the same name as the class, for making new instances. If the class has an *initializer method*, this method is used to get the attributes (i.e. the state) of the new object properly set up.

initializer method

A special method in Python (called \_\_init\_\_) that is invoked automatically to set a newly created object’s attributes to their initial (factory-default) state.

instance

An object whose type is of some class. Instance and object are used interchangeably.

instantiate

To create an instance of a class, and to run its initializer.

method

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

object

A compound data type that is often used to model a thing or concept in the real world. It bundles together the data and the operations that are relevant for that kind of data. Instance and object are used interchangeably.

object-oriented programming

A powerful style of programming in which data and the operations that manipulate it are organized into classes and methods.

object-oriented language

A language that provides features, such as user-defined classes and inheritance, that facilitate object-oriented programming.

**Glossary**

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.

deep equality

Equality of values, or two references that point to objects that have the same value.

shallow copy

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

shallow equality

Equality of references, or two references that point to the same object.