

**SPECIALIST READING**

**A** Find the answers to these questions in the following text.

- 1 What advantages of using object-oriented programming are mentioned in the text?
- 2 What are the three key features of OOP?
- 3 What multimedia data types are referred to in the text?
- 4 List the different types of triangle mentioned in the text.
- 5 What feature avoids the problem of deciding how each separate type of data is integrated and synchronized into a working whole?
- 6 What specific type of rectangle is named in the text?
- 7 What common properties of a rectangle are mentioned in the text?
- 8 What features are made quicker by code reusability?

**OBJECT-ORIENTED PROGRAMMING**

One of the principal motivations for using OOP is to handle multimedia applications in which such diverse data types as sound and video can be packaged together into executable modules.

5 Another is writing program code that's more intuitive and reusable; in other words, code that shortens program-development time.

Perhaps the key feature of OOP is encapsulation - bundling data and program instructions into modules called 'objects'. Here's an example of how objects work. An icon on a display screen might be called 'Triangles'. When the user selects the Triangles icon - which is an object composed of the properties of triangles (see fig. below) and other data and instructions - a menu might appear on the screen offering several choices.

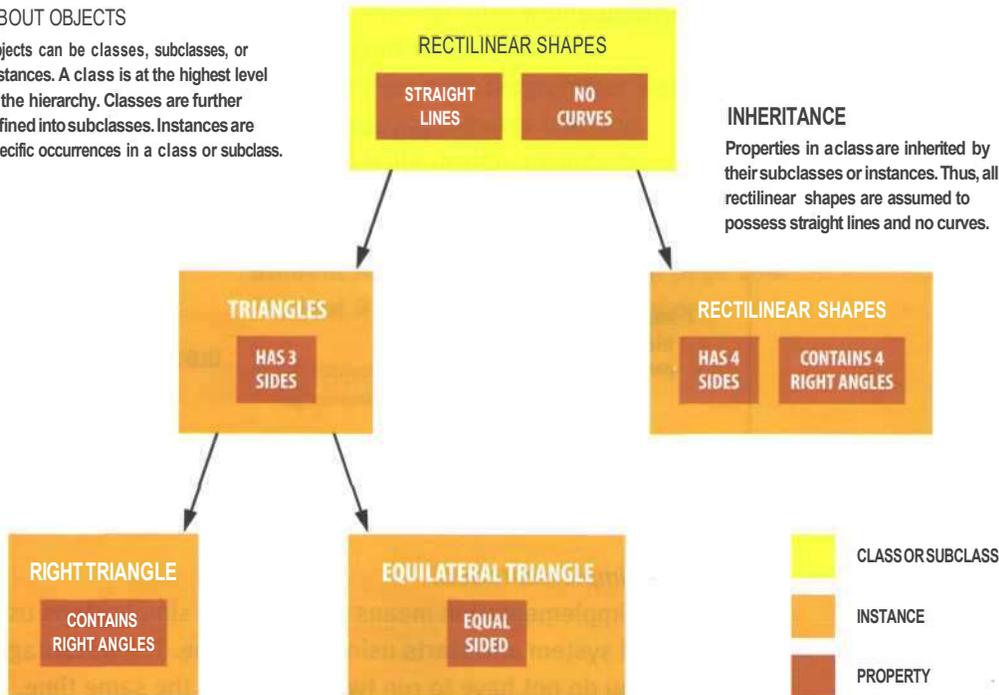
10 The choices may be (1) create a new triangle and (2) fetch a triangle already in storage. The menu, too, is an object, as are the choices on it. Each time a user selects an object, instructions inside the object are executed with whatever properties or data the object holds, to get to the next step. For instance, when the user wants to create a

**ABOUT OBJECTS**

Objects can be classes, subclasses, or instances. A class is at the highest level of the hierarchy. Classes are further refined into subclasses. Instances are specific occurrences in a class or subclass.

**INHERITANCE**

Properties in a class are inherited by their subclasses or instances. Thus, all rectilinear shapes are assumed to possess straight lines and no curves.



25 triangle, the application might execute a set of instructions that displays several types of triangles - right, equilateral, isosceles, and so on.

30 Many industry observers feel that the encapsulation feature of OOP is the natural tool for complex applications in which speech and moving images are integrated with text and graphics. With moving images and voice built into the objects themselves, program developers avoid the sticky problem of deciding how each separate type of data is to be integrated and synchronized into a working whole.

35 A second key feature of OOP is inheritance. This allows OOP developers to define one class of objects, say 'Rectangles', and a specific instance of this class, say 'Squares' (a rectangle with equal sides). Thus, all properties of rectangles - 'Has 4 sides' and 'Contains 4 right angles' are the two shown here - are automatically inherited by Squares. Inheritance is a useful property in rapidly processing business data. For instance, consider a business that has a class called 'Employees at the Dearborn Plant' and a specific instance of this class, 'Welders'. If employees at the Dearborn plant are eligible for a specific benefits package, welders automatically qualify for the package. If a welder named John Smith is later relocated from Dearborn to Birmingham, Alabama, where a different benefits package is available, revision is simple. An icon representing John Smith - such as John Smith's face - can be selected on the screen and dragged with a mouse to an icon representing the Birmingham plant. He then automatically 'inherits' the Birmingham benefit package.

40 A third principle behind OOP is polymorphism. This means that different objects can receive the same instructions but deal with them in different ways. For instance, consider again the triangles example. If the user right clicks the mouse on 'Right triangle', a voice clip might explain the properties of right triangles. However, if the mouse is right clicked on 'Equilateral triangle' the voice instead explains properties of equilateral triangles.

45 The combination of encapsulation, inheritance and polymorphism leads to code reusability. 'Reusable code' means that new programs can easily be copied and pasted together from old programs. All one has to do is access a library of objects and stitch them into a working whole. This eliminates the need to write code from scratch and then debug it. Code reusability makes both program development and program maintenance faster.

**B Re-read the text to find the answers to these questions.**

**1 Match the terms in Table A with the statements in Table B.**

Table A	
a	OOP
b	Encapsulation
c	Object
d	Menu
e	Square
f	Polymorphism
g	Library

Table B	
i	An OOP property that allows data and program instructions to be bundled into an object
ii	A list of choices
iii	An OOP property that enables different objects to deal with the same instruction in different ways
iv	A reusable collection of objects
v	A module containing data and program instructions
vi	Object-Oriented Programming
vii	A rectangle with equal sides

**2 Complete the following text using words from the reading text:**

Encapsulation, ..... and polymorphism are key features of ..... programming. Encapsulation allows data and program instructions to be bundled together in ..... called objects. Inheritance means that specific ..... of a class of objects ..... the properties of the class of objects. Polymorphism means that instructions are treated differently by different..... . The combination of these ..... features of OOP means that program code is reusable. This speeds up ..... and ..... of programs.

[Adapted from 'Understanding Computers Today and Tomorrow', 1998 edition, Charles S. Parker, The Dryden Press]