

Answers to Chapter 6 Questions

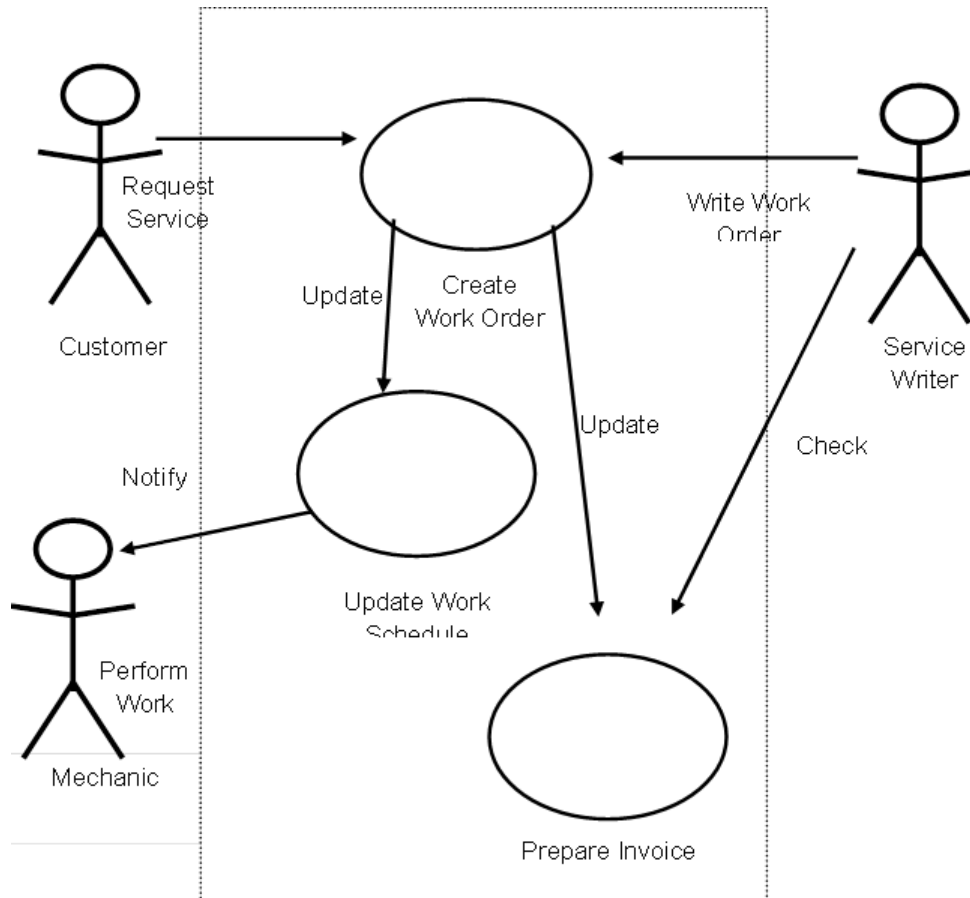
1. What is object-oriented analysis, and what are some advantages of this method?
Object-oriented analysis is a popular approach that sees a system from the viewpoint of the objects themselves as they function and interact with the system during the implementation phase of the SDLC. Systems analysts and programmers transform the objects into program code modules. A modular approach saves money and time, because the modules can be optimized, tested, and reused as often as necessary. **(Page 230)**
2. Define an object, and provide three examples.
An object represents a real person, place, event, or transaction. For example, when a patient makes an appointment to see a doctor, the patient is an object, the doctor is an object, and the appointment itself is an object. **(Page 230)**
3. Define an attribute, and provide three examples.
Attributes are characteristics that describe the object. For example, if you own a vehicle, it has attributes such as make, model, and color. **(Page 230)**
4. Define a method, and provide three examples.
A method defines specific tasks that an object can perform. Just as objects are similar to nouns and attributes are similar to adjectives, methods resemble verbs that describe what and how an object does something. For example, a student object can register for a course, pay a tuition bill, and submit an assignment. **(Page 230)**
5. Define encapsulation, and explain how it is used in object-oriented analysis.
Encapsulation means that all data and methods are self-contained. By limiting access to internal processes, an object prevents its internal code from being altered by another object or process. Encapsulation allows objects to be used as modular components anywhere in the system, because objects send and receive messages, but do not alter the internal methods of other objects. **(Page 236)**
6. Define polymorphism, and provide three examples.
Polymorphism is the concept that a message can have different meanings to different objects. For example the message GOOD NIGHT signals the PARENT object to read a bedtime story, but the same message to the DOG object tells the dog to go to sleep. The GOOD NIGHT message to the CHILD object signals it to get ready for bed. **(Page 235)**
7. Define a class, subclass, and superclass, and provide three examples of each.
A class is a group of similar objects. Subclasses are more specific categories within a class. A superclass is a more general category into which a class falls. For example, a NOVEL class belongs to a superclass called BOOK, because all novels are books. The NOVEL class can have subclasses called HARDCOVER and PAPERBACK. Students be able to suggest many more examples of this concept. **(Pages 236 - 237)**
8. Define an actor, and provide three examples.
An actor is an external entity that initiates a use case by requesting the system to perform a function or process. For example, in a medical office system, a PATIENT (actor) can

MAKE AN APPOINTMENT (use case). In a library, a library patron member can borrow a book, and in an auto shop, a mechanic actor can perform tasks. (Page 239)

9. Define a use case and a use case diagram, and prepare a sample of each.

A use case represents the steps in a specific business function or process. A use case diagram is a visual summary of related use cases within a system or subsystem. One example is an auto service department. The operation involves customers, service writers who prepare work orders and invoices, and mechanics who perform the work.

Example of a use case diagram for an auto service department: (Page 239; 241)



10. Define the term *black box*, and explain why it is an important concept in object-oriented analysis. Can you think of other black boxes that you use in everyday life?

An object can be viewed as a black box, because a message to the object triggers changes within the object without specifying how the changes must be carried out. One advantage to using a black box approach is that you do not need to think about how the object performs a task, as long as it does so properly. (Page 235)