Fill-in-blank. The items in the table below may be used never, once, or multiple times.

classes	state	methods	module	invariance	relationships
identity	objects	construct	action	passive	attributes

(a) UML as a graphical language can be used to _____ software.
(b) For defining UML classes, one should first define _____
(c) To develop UML class diagrams, one should define _____before defining _____.
(d) State of an object is defined as _____ of a class.

Consider the basic relationship types that may be used between two classes. Rank order these relationships from being the strongest (most specific) to weakest (least specific). Clearly write your answers.

Relationship Type	strongest (3), neither strongest nor weakest (2), weakest (1)
Specialization	
Association	
Dependency	

Consider an object and complete the table below. In answering this question, consider each row as completing the sentence "**An object can be …**". Clearly mark your answers with **X**.

	Yes	No
Active		
Passive		
passive and Active		
neither passive nor active		

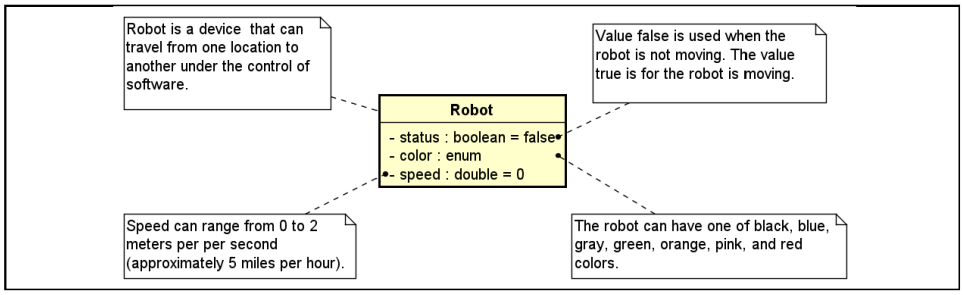
Which of the basic principles of the Object Model should be defined before any other?

(a) Answer _____
(b) Provide a brief explanation.

Consider two modules in a software design.

(a) Each module has maximal relationships within it, but they have minimal relationships to each other. Name and briefly explain two benefits of this design.
(b) Identify the Software Complexity attribute most directly relates to the design in part (a) and explain briefly your answer.

Consider the attributes of the UML class below.



(a) Which one of the attributes contributes the most to software complexity? Explain your answer.
(b) Suppose there is an operation for setting the speed attribute of the `Robot` class. The operation is specified as **`+ setSpeed(double arg) : void`**. Specify an identity should an instance of this class have for the **`setSpeed`** operation to be called by another object. Use Java programming language.

Consider battery-charged flashlights, each having a switch. The switch has two buttons. One is for turning on and another for turning off a light bulb. Such devices can operate as long as their batteries have charge.

(a) Is this kind of device active? YES _____, NO _____.
(b) Explain briefly your answer.

[8 points] Consider the **`RepoAccount`** class that supports reading files. This class was described as a basic repository account that allows adding and removing files where each file contains information about a person's vaccination record for infectious diseases. Suppose the `RepoAccount` class specification is asked to follow the Design-by-Contract method.

(a) What kind of invariance can be used for the `RepoAccount` class?
(b) Define an invariance for the `RepoAccount` class using your answer in part (a).

Consider the hierarchy and IS-A concepts.

(a) Explain a difference between the hierarchy and IS-A.
(b) Provide a simple example of your own choosing with a brief explanation for the IS-A concept.

Consider classes named **`Cal-A`** and **`Cal-B`** for two different calculators. Calculators can divide two numbers ranging between **1** and **99**. Assume one number is defined as variable $n1$ and another is defined as variable $n2$. **`Cal-A`** can operate on integer numbers. **`Cal-B`** can operate on real numbers. Assume the calculators can be used to divide the weight of two packages.

(a) Specify the classes for these calculators using the UML visual notation. Each should store the numbers that can be divided. Include attribute(s) and method(s) needed for the UML classes. Provide brief descriptions of the attribute(s) and method(s) defined for the **`Cal-A`** class. Do not include a description of the **`Cal-A`** class. Do not include descriptions for **`Cal-B`** class, its attributed(s), and its method(s).
(b) Identify the units, if any, for the attribute(s) of the **`Cal-B`** class.

Consider a calculator **`Cal-C`** that can divide $n3$ and $n4$ variables with values ranging from -99.00 to 99.00. The values for $n3$ and $n4$ are real numbers.

(a) Specify the class for this calculator using the UML visual notation. Include the attributes and methods needed for the UML class. Provide descriptions for the method(s) defined for the **`Cal-C`** class. Do not include descriptions for the class and its attribute(s).
(b) What kind of suitable relationships can **`Cal-B`** and **`Cal-C`** have with each other? **`Cal-B`** is your answer to Question 10 (a). Clearly mark your answers with **X**.

	Yes	No
No relationship		
Aggregation		
Class and sub-class		
Class and superclass		
Superclass and sub-class		

Specify a UML class diagram that has the `Cal-B` and `Cal-C` classes. Define a suitable relationship between these classes and include it in the class diagram.