

The University of Melbourne
Department of Computer Science and Software Engineering
433-254 Software Design
Semester 2, 2003

Answers for Tutorial 3
Week 4

1. What are instance and class variables?

Sample Answer:

Instance variables are created for each instance object of a class. Each instance variable is unique to its corresponding object. On the other hand, class variables are shared among all instance objects of a class. In fact, class variables and class methods can exist and can be accessed via their corresponding class without requiring any instance objects.

All variables are instance variables by default unless they are explicitly defined as class variable (using keyword *static*).

2. Explain each of the following:
 - a. Creation of an object
 - b. Instance methods
 - c. Class constructors (with no and one/more arguments)

Sample Answer:

- a) A new object is created whenever a class is instantiated (using keyword *new*). As the result of an object creation (class instantiation), enough memory is allocated to the newly created object to store its instance variables as well as other relevant information then a reference to the beginning of that memory location is returned.
- b) Instance methods can only be called upon instance objects of a class (and not the class itself). By contrast, class methods can be called upon the class that they are defined in as well as instance objects of type that class.

All methods are instance methods by default unless are explicitly defined as class methods (also using keyword *static*).

- c) Constructor is a special kind of method (with the same name as its class) for initializing objects whenever they are created. Object initialization means assigning values to instance variables of an object. A class may have more than one constructor, but each constructor should have a unique signature (different list of input parameters). Constructors don't return any value.
3. Explain why we are able to invoke `Math.sqrt(25.0)` without creating objects of the class `Math`.

Sample Answer:

Well, because it's a class method!