

	<b>MIDTERM EXAMINATION</b> SPRING 2007 CS504 – SOFTWARE ENGINEERING-I	Marks: 40 Time: 90min
--	---	--------------------------

**Q1:** Object models (as described for example using class diagrams) often incorporate the following relations:

- generalisation-specialisation (inheritance)
- whole-part (aggregation)
- association
- object *is instance of* class

Using whichever of these relations you feel is the most appropriate, draw simple diagrams to model the ideas in each of the following sentences (you should use the appropriate UML notation and include labels and numbers where appropriate).

- i. JBuilder is a Java program    ii. A polygon is built from a number of points.    iii. A football team consists of 11 players and 5 substitutes.    iv. Mouse and keyboard are different input devices.    v. A class can have multiple attributes.

**Q:** Software is a product and can be manufactured using the same technologies used for other engineering artifacts. **(T,F)**

**Q:** Ambiguity means \_\_\_\_\_.

- a. ambiguity means that two different readers of the same document interpret the requirement differently
- b. ambiguity means that two different writers of the same document interpret the requirement differently
- c. ambiguity means that different readers of the different documents interpret the requirement differently
- d. both “b” and “c”

# [WWW.Shiningstudy.com](http://WWW.Shiningstudy.com)

**Q:** Determine which of the following relationships are inheritance, association or aggregation. Be aware that there may be 3-way or N-way associations so do not assume that any N-way relationship is inheritance. Also draw the object model in each case.

- a. Car and engine
- b. A polygon is composed of an ordered set of points
- c. Computer and its accessories like keyboard, mouse and monitor

**Q:** Software engineering is often characterized as “programming-in-the-large”. List ten reasons why large projects present such a different challenge to “programming-in-the-small”.

(You should list the ten you feel are the most important – do not give more than ten as any additional ones will not be considered).

**Q:** The current software crisis was caused by the Y2K problem whose seeds were first sown by careless programmers in the early 1970's. (**T, F**)