Chad Philip Johnson
CSCI20, Worthington
November 26th, 2012

Pencil & Paper 13

1. What does UML stand for?
The acronym UML stands for Unified Modeling Language.

2. Briefly (one sentence or so), what is UML and what does it include?
According to Wikipedia, Unified Modeling Language is a standardized general-purpose modeling language that employs a set of graphic notation techniques to create visual models of object-oriented software-intensive systems.

3. What is the purpose of use case diagrams?
According to the article "Practical UML: A Hands-On Introduction for Developers", use case diagrams "describe what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how." Use case diagrams are visual representations of the various ways a piece of software will be used. They communicate the functions of a program through simple descriptions using "actors" (labeled stick figures indicating who or what initiates the events involved in a task), "use cases" (summaries of scenarios for a single task or goal), and "communication association" (the types of connections made between actors and use cases). An alternate description of use case diagrams from Wikipedia states that they "describe the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases."

4. List the three areas in which use case diagrams are most useful.
According to the article "Practical UML: A Hands-On Introduction for Developers", use case diagrams are most useful in the following areas: 1) determining features and requirements, 2) communicating with clients, and 3) generating test cases. These benefits allow use cases to be effective in establishing new requirements as a piece of software takes shape, to provide simple notation so that developers are able to communicate with clients, and to represent a collection of scenarios which may suggest a suite of test cases for those scenarios (where a scenario is "an example of what happens when someone interacts with the system").

5. What is the purpose of a class diagram?
According to the article "Practical UML: A Hands-On Introduction for Developers", class diagrams "give an overview of a system by showing its classes and the relationships among them." Class diagrams display which classes interact with one another but do not detail the types of interactions that occur. An alternate description from Wikipedia states that a class diagram "describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes."

6. List the three parts of the notation used to describe a single class in UML (i.e., the three parts of the class "rectangle").
Class diagrams are represented by a rectangle which is divided into three parts. The first part contains the class name (such as Order), the second its attributes (local fields/variables), and the third its operations (static and instance methods accessible within and/or through the class or an object). Abstract classes also use the same format, but the type of class is emphasized by using italics in the title.

7. List and briefly describe the three types of relationships that can be described by a class diagram.
In a UML document, classes are connected to other classes in ways that communicate the qualities of relationships that exist between them. According to the article "Practical UML: A Hands-On Introduction for Developers", class diagrams have three kinds of relationships: 1) association, which communicates a relationship between instances of two classes and is represented by a link; 2) aggregation, which indicates that one class belongs to a collection through a graphical link containing a diamond symbol; and 3) generalization, which is an inheritance link indicating that one class is a superclass of another which is detailed through a link with a triangle symbol.

8. What is the meaning/purpose of multiplicity in a class diagram?
For every association between classes a number of additional details about the class relationship can be communicated through its link; multiplicity is one of these details ("role name" and "navigability" are others). According to the article "Practical UML: A Hands-On Introduction for Developers", the multiplicity of an association is "the number of possible instances of the class associated with a single instance of the other end. Multiplicities are single numbers or ranges of numbers." An alternate description from Wikipedia states that multiplicity is an optional notation at the end of each association that indicates the number of instances of that entity (that is, the number of objects that participate in the association).

9. What is the difference between a class diagram and an object diagram?
Object diagrams represent individual instances of a class whereas class diagrams represent the class itself. An alternate description of object diagrams from Wikipedia states that they "show a complete or partial view of the structure of an example modeled system at a specific time." Another important difference between the class diagram and object diagram occurs in the naming of the object: object titles have their names underlined and are followed by both a colon and the name of the class from which the object was instantiated.

10. What is the purpose of an interaction diagram?
According to the article "Practical UML: A Hands-On Introduction for Developers", unlike class and object diagrams, which are static model views, interaction diagrams are dynamic and describe how different objects collaborate. Interaction diagrams have unique properties such as the "lifeline" which represents the amount of time that an object exists after a message call. An alternate description of interaction diagrams from Wikipedia states that they "provide an overview in which the nodes represent communication diagrams."